WebFOCUS

Developing Reporting Applications With Graphical Tools

Version 7 Release 6.9 and Higher
Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher’s intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2009, by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.
How It Works ............................................................................................................. 73
Using ClearCase Integration as the Source Control Product ........................................ 81
Using CVS Integration as the Source Control Product ................................................ 84
Using Subversion (SVN) Integration as the Source Control Product ......................... 91
Setting Up Developer Studio With ClearCase, CVS, Subversion (SVN), or ChangeMan DS ......................................................................................................................... 96
Source Control Options .......................................................................................... 105
Securing a Project ......................................................................................................... 108
Removing a Project ........................................................................................................ 109
4. Creating a Reporting Procedure .............................................................................. 111
Selecting a Creation Tool ............................................................................................... 112
Incorporating a Procedure Into an Application ........................................................... 118
Copying a Procedure ...................................................................................................... 120
Creating a Procedure Component .................................................................................. 120
Working With a Component ............................................................................................ 127
Using the SQL Report Wizard ......................................................................................... 130
Assigning a Logical Name With the Allocation Wizard .................................................... 138
Clearing Allocations ....................................................................................................... 155
Calling a Procedure From the Current One ....................................................................... 156
Managing Flow of Control ............................................................................................... 164
Working With a Full Procedure ....................................................................................... 166
Running a Procedure ..................................................................................................... 168
Canceling a Running Procedure ....................................................................................... 168
5. Enhancing a User Interface ...................................................................................... 169
Displaying a Report in a Helper Application ..................................................................... 170
Controlling Multiple Reports .......................................................................................... 176
Adding JavaScript for Drill-Down Reporting ..................................................................... 191
Facilitating Report Manipulation ...................................................................................... 198
Using the WebFOCUS Viewer ......................................................................................... 199
Opening and Closing the WebFOCUS Viewer .................................................................. 201
Printing With On-Demand Paging ............................................................................... 203
WebFOCUS Viewer Navigation ....................................................................................... 203
Using a Cascading Style Sheet to Standardize Display .............................................................. 204
Displaying a Previously Run Report ......................................................................................... 211
Passing a User ID From HTML for a Custom Menu ............................................................... 212
Customizing a Menu .............................................................................................................. 213

6. Partitioning and Deploying Project Files ......................................................................... 239
   Deployment Basics .............................................................................................................. 240
   Summary of Steps .............................................................................................................. 240
   Step 1: Identify the Target Servers ..................................................................................... 241
   Step 2: Create a Deployment Scenario .............................................................................. 241
   Step 3: Partition the Project Files ....................................................................................... 249
     Working in a Default Partition Environment ..................................................................... 253
     Setting Running Paths .................................................................................................... 255
   Step 4: Deploy the Project Files .......................................................................................... 256
   Configuring the Target Servers (Optional) .......................................................................... 261

7. Editing Application Components as Text in Developer Studio ..................................... 269
   Text Editor ........................................................................................................................ 270
     Accessing Text Editors ................................................................................................. 271
   The Other Component ...................................................................................................... 272
   The Comment Component ............................................................................................... 273
     Adding and Removing Comments in the Text Editor ...................................................... 274
   Creating a Text File or a Procedure Component as Text .................................................. 276
   Opening Application Components as Text ........................................................................ 279
   Finding and Replacing Text .............................................................................................. 279
   Changing Text Color and Case Size ................................................................................. 280
   Adding Headings and Footings .......................................................................................... 282
   Using Bookmarks to Move Within a File ......................................................................... 283
   Running a Procedure From the Editor .............................................................................. 285
   Opening a Graphical Tool From the Text Editor ............................................................... 286

8. Creating an Update Application With Update Assist ..................................................... 289
   Update Assist (Step 1 of 6): Selecting Segments to Update ............................................. 290
   Update Assist (Step 2 of 6): Selecting Fields to Update ................................................... 291
     Change in When User-Supplied Data Is Validated ........................................................ 293
## Contents

Update Assist (Step 3 of 6): Selecting Navigation Options ................................................................. 297  
Update Assist (Step 4 of 6): Selecting a Color Scheme ........................................................................ 299  
Update Assist (Step 5 of 6): Selecting Output File Options .................................................................. 300  
Update Assist (Step 6 of 6): Confirming Selections ............................................................................. 301  
About Your Update Assist Application ..................................................................................................... 302  
Editing Your Update Assist Application .................................................................................................... 305  
  Changing the Search Field for Tree and Combo Box Navigation ............................................................ 305  
  Customizing the Tree Control .................................................................................................................. 307  
  Calendar Control for Date-Formatted Fields ......................................................................................... 309  
  Date-Stamping Fields ............................................................................................................................... 309  
  Auto-numbering Fields in Update Assist Applications ......................................................................... 310  
  Continuing Displaying Currently Displayed Values After a New Action .............................................. 311  
Calling an Update Assist Procedure From a WebFOCUS Report ............................................................ 312  
  Calling an Update Assist Project From a WebFOCUS Report Example ................................................ 313  
Usage Notes ............................................................................................................................................... 322  

**Reader Comments** ................................................................................................................................. 331
Preface

This documentation describes how to use WebFOCUS graphical tools to create and deploy self-service report applications on the Internet or corporate intranets. It is intended for application developers.

Note: The WebFOCUS toolset generates the rich FOCUS fourth generation language. While this language is very extensive, the WebFOCUS toolset only supports a subset of the language and only specific syntax constructs. While the user can manually modify the content of these WebFOCUS procedures/files, there is no guarantee that the user will be able to open the modified procedure in the tool.

How This Manual Is Organized

This manual includes the following chapters:

<table>
<thead>
<tr>
<th>Chapter/Appendix</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Where Am I?</td>
<td>Explains where to find the information you are looking for within the WebFOCUS documentation set.</td>
</tr>
<tr>
<td>2 Understanding User Interface Basics</td>
<td>Introduces user interface design in WebFOCUS and describes the components of a typical user interface.</td>
</tr>
<tr>
<td>3 Creating a Reporting Application</td>
<td>Describes how to create a reporting application in the local development environment.</td>
</tr>
<tr>
<td>4 Creating a Reporting Procedure</td>
<td>Describes how to create a procedure, which is the core element of a reporting application.</td>
</tr>
<tr>
<td>5 Enhancing a User Interface</td>
<td>Describes coding capabilities that extend the functionality and usability of an interface, including the display of reports in popular formats and the display of multiple reports on a single launch page.</td>
</tr>
<tr>
<td>6 Partitioning and Deploying Project Files</td>
<td>Describes the steps you will perform for successful deployment of your project files to the Web.</td>
</tr>
<tr>
<td>Chapter/Appendix</td>
<td>Contents</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
</tr>
<tr>
<td>7</td>
<td>Editing Application Components as Text in Developer Studio</td>
</tr>
<tr>
<td>8</td>
<td>Creating an Update Application With Update Assist</td>
</tr>
</tbody>
</table>

### Documentation Conventions

The following table lists and describes the conventions that apply in this manual.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>THIS TYPEFACE</strong> or <strong>this typeface</strong></td>
<td>Denotes syntax that you must enter exactly as shown.</td>
</tr>
<tr>
<td><strong>this typeface</strong></td>
<td>Represents a placeholder (or variable), a cross-reference, or an important term.</td>
</tr>
<tr>
<td><strong>underscore</strong></td>
<td>Indicates a default setting.</td>
</tr>
<tr>
<td><strong>this typeface</strong></td>
<td>Highlights a file name or command. It may also indicate a button, menu item, or dialog box option you can click or select.</td>
</tr>
<tr>
<td>Key + Key</td>
<td>Indicates keys that you must press simultaneously.</td>
</tr>
<tr>
<td>{}</td>
<td>Indicates two or three choices; type one of them, not the braces.</td>
</tr>
<tr>
<td>[]</td>
<td>Indicates a group of optional parameters. None is required, but you may select one of them. Type only the parameter in the brackets, not the brackets.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis points (...).</td>
</tr>
</tbody>
</table>
**Convention**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicates that there are (or could be) intervening or additional commands.</td>
</tr>
</tbody>
</table>

**Related Publications**

To view a current listing of our publications and to place an order, visit our Technical Documentation Library, [http://documentation.informationbuilders.com](http://documentation.informationbuilders.com). You can also contact the Publications Order Department at (800) 969-4636.

**Customer Support**

Do you have any questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques, [http://forums.informationbuilders.com/eve/forums](http://forums.informationbuilders.com/eve/forums).

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, [http://www.informationbuilders.com](http://www.informationbuilders.com). It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.informationbuilders.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Service (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (xxxx.xx) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

**Information You Should Have**

To help our consultants answer your questions effectively, be prepared to provide the following information when you call:
Information You Should Have

- Your six-digit site code (xxxx.xx).

- Your WebFOCUS configuration:
  - The front-end you are using, including vendor and release.
  - The communications protocol (for example, TCP/IP or HLLAPI), including vendor and release.
  - The software release.
  - Your server version and release. You can find this information using the Version option in the Web Console.

- The stored procedure (preferably with line numbers) or SQL statements being used in server access.

- The Master File and Access File.

- The exact nature of the problem:
  - Are the results or the format incorrect? Are the text or calculations missing or misplaced?
  - The error message and return code, if applicable.
  - Is this related to any other problem?

- Has the procedure or query ever worked in its present form? Has it been changed recently? How often does the problem occur?

- What release of the operating system are you using? Has it, your security system, communications protocol, or front-end software changed?

- Is this problem reproducible? If so, how?

- Have you tried to reproduce your problem in the simplest form possible? For example, if you are having problems joining two data sources, have you tried executing a query containing just the code to access the data source?

- Do you have a trace file?

- How is the problem affecting your business? Is it halting development or production? Do you just have questions about functionality or documentation?
User Feedback

In an effort to produce effective documentation, the Documentation Services staff welcomes your opinions regarding this manual. Please use the Reader Comments form at the end of this manual to communicate suggestions for improving this publication or to alert us to corrections. You can also use the Documentation Feedback form on our Web site, http://documentation.informationbuilders.com/feedback.asp.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our World Wide Web site (http://www.informationbuilders.com) or call (800) 969-INFO to speak to an Education Representative.
Where Am I?

The WebFOCUS core set of documentation contains information on using data sources, developing applications, creating reports, and reference materials. This topic helps explain where to find the information you are looking for.

Topics:
- WebFOCUS Core Documentation Roadmap
WebFOCUS Core Documentation Roadmap

The following illustrates the relationship of the manuals in your WebFOCUS Core documentation set.

If you are developing a WebFOCUS application...

- *Developer Studio Application Development Getting Started* provides an overview of the process of developing an application in WebFOCUS Developer Studio and deploying it to the Web. This also includes a description of the tools that are available for developing your applications, as well as tutorials to familiarize you with the development process.

- *Developing Reporting Applications* discusses application logic and the structure of an application. This includes detailed instructions on how to create the procedures and user interface that will comprise your application.

- *Developing Reporting Applications With Graphical Tools* discusses application logic and the structure of an application. It provides the techniques for developing an application in WebFOCUS Developer Studio and deploying it to the Web.
If you are creating a report request...

- *Creating Reports With WebFOCUS Language* describes how to create, style, as well as, analyze reports and graphs with the WebFOCUS language.

- *Creating Reports With Graphical Tools* describes how to create, style, and analyze reports and graphs with the tools provided in WebFOCUS Developer Studio. It also includes information on ReportCaster.

If you are an administrator or developer configuring the data sources for a WebFOCUS application...

- *Describing Data With WebFOCUS Language* contains information on how to use the WebFOCUS language to create the metadata for the data sources that your WebFOCUS applications will access.

- *Describing Data With Graphical Tools* contains information on how to use WebFOCUS Developer Studio to create the metadata for the data sources that your WebFOCUS applications will access.

If you are looking for information on a reference topic...

- *Using Functions* describes how to use Information Builders-supplied functions to perform complex calculations and manipulate data in your procedures.

If you do not find the information you are looking for...

- *Functions and Subroutines*, previously found in *Developing Reporting Applications*, is now a stand-alone manual titled *Using Functions*. This manual describes all reporting and Maintain functions, and explains how to use them to perform calculations and manipulate data.

- The following topics previously found in *Developing Reporting Applications* are now located in *Unique FOCUS Topics*:

  - *Working With Cross-Century Dates*
  - *Establishing String Substitutions*
  - *Designing User Interfaces for Windows Reporting Applications*

*Unique FOCUS Topics* contains topics previously contained in *Developing Reporting Applications* and other Information Builders manuals that are not part of the current Web methodologies of WebFOCUS, but are still supported for those who wish to use them. You can find *Unique FOCUS Topics* on the WebFOCUS Documentation CD.
Additional Documentation

In addition to the documentation listed in this topic, documentation is available for WebFOCUS installation and configuration as well as other WebFOCUS products. This documentation is available at http://documentation.informationbuilders.com.
A user interface for a Web application is a collection of HTML files that display pages in a browser.

You can design a user interface by using the Developer Studio’s HTML Composer to create HTML forms and procedures in one integrated process. For details, see the Designing a User Interface for a Web Application With the HTML Composer manual.

In addition to the above technique for creating an interface, you can code a user interface or modify a supplied template. For details, see Coding a User Interface in the Developing Reporting Applications manual and Enhancing a User Interface on page 169.

Topics:
- User Interface Basics
- About a Logon Page
- About a Launch Page
- About an HTML Display Page
User Interface Basics

A user interface is the means by which you enable communication between a user and your application.

A WebFOCUS user interface allows a user to supply information, choose options, and run reports. A simple one typically consists of:

- **A logon page**, which prompts for security credentials to ensure that a user is authorized to access an application. A logon page is recommended but not required.

- **A launch page**, which enables a user to run one or more reports.

- **An HTML display page**, on which a report appears.

- **A main (home) page**, an application’s entry point, providing access to all application features. There are many different kinds of main pages, each depending on the purpose and complexity of the application.

User interface design is one of the most creative phases of application development. Developer Studio provides many tools that support it.

About a Logon Page

In this section:

Setting a Browser

How to:

Modify WFSIGNON.HTML

One way to ensure connection to a secure WebFOCUS Reporting Server is to include a logon page. The sample logon page WFSIGNON.HTML is supplied with WebFOCUS software in the following default location:

**Windows:** `install_drive:\ibi\WebFOCUS76\ibi_html`

**UNIX:** `/ibi/WebFOCUS76/ibi_html`

**z/OS:** `/ibi/WebFOCUS76/ibi_html`

You can copy the file and modify the code for your application, or create a custom page of your own.
The sample logon page prompts the user for an ID and password. The values are assigned to the variables IBIC_user and IBIC_pass and sent to the Reporting Server for authentication. If authentication is successful, the values are encrypted and stored in a cookie. Using a cookie, you set security credentials once. Subsequent requests to the Reporting Server automatically include the proper authorization.

For details on security, see the WebFOCUS Security and Administration manual.

You can use values collected on a logon page to customize components of an interface for an individual user. For an example, see Enhancing a User Interface on page 169.

The following is the supplied logon page displayed in a browser.

![Logon Page Example](http://localhost/ibi_html/wfsignon.html)

**Procedure: How to Modify WFSIGNON.HTML**

1. Modify WFSIGNON.HTML by using the Developer Studio text editor (TED).

   **Note:** For details on this editor, see Editing Application Components as Text in Developer Studio on page 269.
2. Modify the lines with letters on the left, as explained in the notes following the code.

```html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head>
<title>WebFOCUS signon screen</title>
</head>
<body bgcolor="#FFFFFF" alink="#0000FF">
<font size=4>

a. <form action="/ib_apps/WFServlet" method="POST">
   <a href="http://www.informationbuilders.com"><img border="0"
    src="ib_logo_home.gif" width="114" height="48"></a>
   <p>&nbsp;</p>
   <p>
   UserID :
   <input type=text size=15 maxlength=256 name="IBIC_user" value="">
   <pre>
   </pre>
   PassWord :
   <input type=password size=15 maxlength=256 name="IBIC_pass"
   value="">
   <pre>
   </pre>
   <input type=submit name="submit" value="Submit"></tr>
   <input type=reset name="reset" value="Reset"></tr>

b. <input type=hidden name="IBIWF_action" value="WF_SIGNON"></tr>
</form>
</font>
</body>
</html>
```

a. Replace the hyperlink to the Web site shown with a hyperlink that applies to your
application. Do the same for the image.

b. After this line

```html
<input type=hidden name="IBIWF_action" value="WF_SIGNON"></tr>
```
add the code that identifies the first application page that displays after successful
logon. The syntax is

```html
<input type="hidden" name="WF_SIGNON_MESSAGE"value="address/first_pg.htm">
```

where:

`address` is an absolute or relative address. See the examples. Use the address
that applies to WebFOCUS.
first_pg is the name of the page that displays after logon. For example, if your application includes a main page, it is the name of that page.

3. Apply formatting to the page as desired, and save it.

4. Use the Deploy Wizard to deploy the logon page with the rest of your application files. For details, see Partitioning and Deploying Project Files on page 239.

Example: Specifying an Absolute Address and Specifying a Relative Address

The following are examples of specifying an absolute address and specifying a relative address.

**Specifying an Absolute Address**

```html
<input type="hidden" name="WF_SIGNON_MESSAGE" value="http://123.4.5.6/app_frame/home.htm">
<input type="hidden" name="WF_SIGNON_MESSAGE" value="http://jonesweb/app_frame/home.htm">
```

**Specifying a Relative Address**

The following assumes the existence of a Web server alias (ibi_html) for the path in which HTML files reside:

```html
<input type="hidden" name="WF_SIGNON_MESSAGE" value="/ibi_html/sales_app/welcome.htm">
```

Setting a Browser

**How to:**

Set Microsoft Internet Explorer to Accept All Cookies

A secure WebFOCUS application that includes a logon page requires a browser that accepts cookies. A cookie is the mechanism that WebFOCUS uses to pass information to and from the Web server. If your browser is not configured to accept all cookies, you may experience unexpected behavior.

All examples in the user interface chapters are run with Microsoft Internet Explorer on Windows 71. Netscape Navigator is not supported.

**Procedure:** How to Set Microsoft Internet Explorer to Accept All Cookies

1. Launch Microsoft Internet Explorer.

2. From the Tools menu, select Internet Options.
3. Click the Security tab.
4. Click Custom Level.
5. Under the Cookies option, select Enable for both items.
6. Click OK.

About a Launch Page

In this section:
- Adding a Control to a Form
- Types of Launch Pages
- Ways to Create a Launch Page

Suppose that managers on their company intranet periodically need a sales report for one of four regions. You:

- Create the procedure, including a variable for the region so that the report can run with different values.
- Create the launch page, from which a manager runs the report. The page prompts for the region of interest.

The following image shows a sample launch page.

![Sample Launch Page](image-url)
If a manager selects Midwest and clicks Run, the report appears.

Coffee Sales for Midwest Region

<table>
<thead>
<tr>
<th>Unit Sales</th>
<th>Dollar Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>332777</td>
<td>4178513</td>
</tr>
</tbody>
</table>

At minimum, a launch page is a simple form that contains a Run and Clear button, allowing the user to run a report. More often, it is a form, like the one shown, containing controls that prompt for user input.

Controls include text boxes, check boxes, radio buttons, drop-down menus, push buttons, and hyperlinks. A launch page usually displays default values or valid values to help the user provide input. The Reporting Server uses the input to process the procedure and return the desired output.

When a launch page is complete, deploy it to the Web server using the Deploy Wizard. Supporting files such as procedures are deployed to the Reporting Server. Once the files are stored on the appropriate servers, users can access the launch page by typing the URL in the browser. See *Partitioning and Deploying Project Files* on page 239.

### Adding a Control to a Form

**Reference:** WebFOCUS Controls

A control prompts a user for a value required by a procedure. You can add a single control, or a combination of controls, to a form on a launch page.

You must add a Submit button. Submitting a form calls WebFOCUS and passes the supplied values to the procedure.

**Reference:** WebFOCUS Controls

The following table describes types of controls:

<table>
<thead>
<tr>
<th>Control</th>
<th>Enables a user to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Line Text Box</td>
<td>Enter a single value for a variable, such as user name or password on a custom logon page.</td>
</tr>
<tr>
<td>Radio Button</td>
<td>Choose one value from a group.</td>
</tr>
</tbody>
</table>
About a Launch Page

<table>
<thead>
<tr>
<th>Control</th>
<th>Enables a user to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Box</td>
<td>Choose multiple values.</td>
</tr>
<tr>
<td>Drop-Down List</td>
<td>Choose a value from a drop-down list.</td>
</tr>
<tr>
<td></td>
<td>You can add text to describe the values.</td>
</tr>
<tr>
<td>Multi-Select Drop-Down List</td>
<td>Choose multiple values from a drop-down list. Allows a user to report on several field values at one time.</td>
</tr>
<tr>
<td>Dynamic Multi-Select Drop-Down List</td>
<td>Choose multiple values from a drop-down list populated by field values retrieved live from a data source.</td>
</tr>
<tr>
<td>Text Area</td>
<td>Enter an ad hoc report request.</td>
</tr>
<tr>
<td>User-Customized Menu</td>
<td>For information on this control, see Enhancing a User Interface on page 169.</td>
</tr>
<tr>
<td>Submit Button</td>
<td>Run a report.</td>
</tr>
<tr>
<td></td>
<td>You must include a Submit button on a form. You may optionally include a button that clears the form of current values and restores the defaults.</td>
</tr>
<tr>
<td>Push Button</td>
<td>Trigger an action.</td>
</tr>
<tr>
<td>Hyperlink</td>
<td>Run a procedure using fixed values.</td>
</tr>
</tbody>
</table>

**Types of Launch Pages**

A launch page can be a single page or a frameset. A single page is a one-page display that is replaced by report output when a user submits a request. A frameset is a multiple-page display, and any page (frame) can be replaced by report output.

**Ways to Create a Launch Page**

A launch page calls the WebFOCUS Client to pass the name of a report to run, with any required values. The WebFOCUS Client may be the CGI (Common Graphical Interface), ISAPI (the proprietary API for the Microsoft Internet Information Server), or the WebFOCUS Servlet.

You can create a launch page in the following ways:

- Use the HTML Composer. The HTML Composer enables you to create a launch and display page in an integrated process within Developer Studio. See the Designing a User Interface for a Web Application With the HTML Composer manual.
- Code the call to the WebFOCUS Client yourself. For details, see *Coding a User Interface* in the *Developing Reporting Applications* manual.

- Customize templates supplied with your software. They include the required code for the CGI, but need modification for your application. For details, see *Coding a User Interface* in the *Developing Reporting Applications* manual.

### About an HTML Display Page

Report output from a Reporting Server appears on an HTML page in a browser. You have many design options for report display. For example, you can display a report on a page by itself or in a frame on a page, or you can design a custom page with multiple reports.

Your application will probably include different types of reports, which may influence the display pages you design. Some types of reports include:

- **Dynamic.** A report created and formatted at the time a user requests it. This type of report displays the most current information from live data retrieval and is called by the WebFOCUS Client. Two kinds of dynamic reports are drill-down and ad hoc.

- **Drill-Down.** A dynamic, multilevel report consisting of a parent (summary) component and a child (detailed) component. For an example of a drill-down report, see *Enhancing a User Interface* on page 169.

- **Ad Hoc.** Dynamic report, coded by a user directly on a launch page. For instructions on enabling ad hoc reporting, see *Coding a User Interface* in the *Developing Reporting Applications* manual.

- **Static.** HTML file created and formatted before a user requests it. Ready for access without live data retrieval. Called by HTML code (A HREF).

  For instructions on creating and calling a static report, see *Coding a User Interface* in the *Developing Reporting Applications* manual.

You can create a report using the Report Painter or TED.
You can make reports available throughout your enterprise by developing reporting applications. Reports created in a reporting application can be grouped together and run using a Web-based form. You can develop basic or complex applications locally and then deploy them to the Web, where they are run as self-service applications.

You can maintain a reporting application by updating or adding Master Files, reporting procedures, HTML files, or other components.

This section describes how to create a self-service reporting application in the local development environment. For information on creating a Managed Reporting application, see your Managed Reporting documentation.

---

**Topics:**
- What Is a Reporting Application?
- Project-based and Remote Development Environments
- Creating a Project
- Organizing a Project
- Using Workspace Files
- Adding a Master File to a Project
- Uploading Data Files
- Viewing and Modifying Project Properties
- Managing a Project With Source Control
- Securing a Project
- Removing a Project
What Is a Reporting Application?

A reporting application is a self-contained program with an interface that helps the user accomplish reporting tasks. It typically consists of one or more Master Files, procedures, and HTML files. The Master Files interpret the data sources that are accessed, while procedures define how that data is retrieved and displayed. HTML forms enable a user to interact with the application and manage the display of data in a browser.

For example, to run a self-service reporting application, a user opens a launch page in a browser and enters the values for the desired output through controls such as drop-down lists or radio buttons. For instance, the user may request a sales report for the Northeast region rather than the Southeast region. When the report is run, Developer Studio passes the user-supplied values to the WebFOCUS Reporting Server (via the Web server), where the procedure is processed.

The procedure requests data from the data source, using the Master File to understand how the data is organized. The Reporting Server compiles the answer set from the data source and returns the output to an HTML page that is displayed in the user's browser.

Project-based and Remote Development Environments

Developer Studio provides a local (stand-alone) development environment and a remote development environment. From the Explorer's Projects folder, you can develop projects locally on your machine or against a remote WebFOCUS environment. From the Explorer's WebFOCUS Environments folder, you develop directly against environments that are configured remotely.

In Developer Studio, you start building a reporting application as a project consisting of different kinds of files. You can create the project as a stand-alone application in the development environment or as a Web-based self-service application that you can deploy.

If you install a WebFOCUS Reporting Server during the Developer Studio installation procedure, you can:

- Locally develop and deploy self-service applications from the Projects area.
  
  Stand-alone Project-based development and deployment requires installation of a WebFOCUS Reporting Server on the same machine as Developer Studio. A WebFOCUS Client is also required for Project-based development. The files that you create for a local project reside in a subdirectory under APPROOT (defined in the configuration files edaserve.cfg and cgivars.wfs). The Application Root directories (APPROOT directories) attribute must point to the same directory for Project-based development since files will be created with the WebFOCUS Client, which resides on the Web server.

- Connect to one or more remote servers and modify existing self-service applications on those servers. For example, you can add a reporting procedure to an existing application.
Configure access to one or more WebFOCUS environments so that you can manage resources on the WebFOCUS Client and Reporting Server, and in the Managed Reporting Repository (if installed). From the environment tree you can create and edit procedures, metadata, HTML files, and more.

If you do not install a WebFOCUS Reporting Server during the Developer Studio installation procedure, your environment allows the last two capabilities.

For details, see the Developer Studio Application Development Getting Started manual.

Creating a Project

**How to:**
Create a Project Directory

**Reference:**
File Types

In Developer Studio, you start building a reporting application as a project consisting of different kinds of files. The project can be a stand-alone application in the development environment or a Web-based application implemented through the deployment feature.

Before you begin local development, you must create a project directory that contains the associated files using the Project Wizard. The Project Wizard enables you to name the project, designate a directory for it, and optionally add other directory paths to data sources from which the project can retrieve information, or paths to other resources.

When you have completed the Project Wizard, the new project is added as a node in the Explorer under Projects on localhost. By default in the Developer Studio Edition, a project node is created with folders labeled HTML Files, Maintain Files, Master Files, Procedures and Other. (Maintain is not available in the Power Reporter.) These folders are called virtual folders because they apply a logical structure to a project but do not actually exist as physical directories.

**Tip:** You can organize a project in many ways. For details, see Organizing a Project on page 35.

To create a self-service reporting application, you typically add files to each of these folders with the exception of the Maintain Files folder, which is optional. To add data maintenance functions, use the Maintain Files folder to launch the Maintain development environment. For more information, see the Maintain Getting Started manual.

A sample project named SESSION is available as a storage space for your use.

**Tip:** You can drag and drop any component of a project (for example, an HTML file, Master File, or Procedure) between projects or Explorer environments.
Creating a Project

Reference: File Types

The major project components are organized in folders:

- **HTML Files folder.** Stores HTML files (.htm). This file type represents a Web form (launch page) and manages the display of reports in a browser. For more information on launch pages, see *Understanding User Interface Basics* on page 17. You can add other HTML resources to this folder, including .html, .js, .class, and image files, and view them from the folder.

- **Master Files folder.** Stores Master Files (.mas) and associated Access Files (.acx). These file types describe the data you report against. For more information on Master Files and Access Files, see the *Describing Data With Graphical Tools* manual and the *Describing Data With WebFOCUS Language* manual.

- **Procedures folder.** Stores procedure files (.fex). This file type is the core element of a project. It specifies the data to be retrieved from a data source and the format in which the data is presented. For more information, see *Creating a Reporting Procedure* on page 111.

- **Other folder.** Stores any visible file type that is not assigned to an existing virtual folder. For example, if SQL files are available in the application and are not assigned to a different virtual folder through the Filters option, then SQL files are listed in the Other folder. File types visible in the Other folder can be reassigned so they appear in a different virtual folder. For more information about reassigning file types, see *Organizing a Project* on page 35.

A project can use all of the following types of files.

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure</td>
<td>Includes files that contain the executable functions of a project, such as procedures (.fex).</td>
</tr>
<tr>
<td>File description</td>
<td>Includes Master Files (.mas) and Access Files (.acx).</td>
</tr>
<tr>
<td>Data source</td>
<td>Includes files that contain data values.</td>
</tr>
<tr>
<td>HTML resource</td>
<td>Includes files displayed to the end user through a Web browser, such as Web pages and HTML files (.htm, .html), graphic images (.gif, .jpg), and Java executable objects (.class).</td>
</tr>
<tr>
<td>Temporary data file</td>
<td>Includes temporary hold files (.ftm) that Developer Studio creates during processing, and a number of temporary work files used internally.</td>
</tr>
</tbody>
</table>
**Procedure: How to Create a Project Directory**

1. Do one of the following:
   - Right-click the Projects on localhost node and choose New Project.
   - Highlight the Projects on localhost node and click (New) on the Main toolbar.
   The Create a Project - Step 1 of 2 dialog box opens.

2. In the first input field, type a name for the new project. The name can be up to 18 characters in length. It can include spaces.

   Developer Studio creates a new subdirectory on the WebFOCUS Reporting Server under APPROOT. APPROOT is a variable defined in the edaserve.cfg file. The default setting for APPROOT is `install_drive:\ibi\apps`.

   For example, if you create a project named Sales, Developer Studio creates a subdirectory named Sales under APPROOT. The newly created subdirectory will contain all project files, including the .GFA file. The .GFA file is an XML control file that contains information about the project, such as associated deployment scenarios, folder names and file types, and accessible servers.

   **Important:** Do not modify the .GFA file. Modification may corrupt the project.

   Subdirectory names do not support spaces. If you include a space in the project name (for example, Human Resources), Developer Studio replaces the space with an underscore in the subdirectory name (for example, Human_Resources).
3. The Project Wizard displays the project name as the default directory in the second input field.

In the second input field, you can optionally enter the name of a subdirectory associated with another project. In that case, the new project and the existing project will share project files.

**Note:** Project directories are created in lowercase.

4. Before you entered information on the dialog box, the default location and name of the new project was shown as IBFS:/localhost/DEV. This value represents the location of the project in the Information Builders File System (IBFS), where localhost is the name of the server on which the project will be created, and DEV is an internal string indicating that you are working in the project (development) area. The value of the IBFS location translates to the value of APPROOT.

Once you enter information on the dialog box, the location and name changes to a value like IBFS:/localhost/DEV/sales/Sales.GFA, where Sales is the project directory, and Sales.GFA is the project control file created by the Project Wizard.

The following image shows a completed dialog box.

5. Click Next to continue.

6. A message is displayed if the directory does not exist. Click Yes to create the new directory. Click No to change the name.
The Create a Project - Step 2 of 2 dialog box opens.

7. From this dialog box, you can add other directories to the project path. Project files in those directories are visible to the project and can be made accessible for development and deployment. The order of directories in the list box represents the search sequence that the project uses.

To add directories to the project, click Add. From the Browse for Folder dialog box, select a folder to add to the project path. The dialog box displays the folders available on the WebFOCUS Reporting Server, based on the value of APPROOT.
By default, baseapp is automatically added to the search path for project resources. You can use the baseapp folder to place common (shared) files in a central location.

Click OK. The directory (folder) is added. To add more directories, repeat this step.

If you highlight a directory in the list box on the Create a Project dialog box, click Add, and select a directory in the Browse for Folder dialog box, the new directory will be added after the highlighted directory.

- To change the order of the directories, select the directory and click Move Up or Move Down.
- To remove a directory from the list, select the directory and click Remove.

When you are satisfied with the order of the directories and have no other changes, click Finish, and the new project is added to the Explorer.

**Tip:** You can change directory paths for a project after you exit the Project Wizard. See Viewing and Modifying Project Properties on page 56.
**Organizing a Project**

**How to:**

- Add a Virtual Folder to a Project and Customize File Types
- Customize Directories for Virtual Folders
- Rename a Component
- Remove a Component

Developer Studio has many features that help you organize a project:

- You can add a virtual folder to an existing project, or a virtual subfolder to an existing folder, and customize the file types that display in it. For details, see *How to Add a Virtual Folder to a Project and Customize File Types* on page 36.

- You can customize the file types that display in any folder using the Project Properties dialog box. For details, see *How to Customize Directories for Virtual Folders* on page 38. To change the display, see *How to Customize Display of File Types* on page 59.

- You can rename a component to identify its function in the project, or remove a component that is no longer used. You can rename and remove components such as folders, subfolders, HTML files, Master Files, and procedures. For more details, see *How to Rename a Component* on page 40. To remove a component, see *How to Remove a Component* on page 40.
Procedure: How to Add a Virtual Folder to a Project and Customize File Types

1. Select and right-click the project. Select New from the context menu, and click Virtual Folder.

   The New Virtual Folder dialog box opens.

2. In the Name field, enter a name for the new virtual folder.

   Customize the list of file types that will be displayed in the folder. File types (filters) already defined in the project are shown in the list box. Click the check box next to a file type in this list to add it to the new folder.
To add a file type not shown in the list, click the Add new file type filter(s) button. The New Filter dialog box opens, listing the file types registered with the operating system.

![New Filter dialog box](image)

3. Scroll through the list and select one or more file types. Click OK when you are done. The selected file type is added to the list.

To remove a file type from the list, select it and click the Remove file type filter(s) button. Confirm the deletion as prompted.

Click OK on the New Virtual Folder dialog box to close it and return to the Explorer.

This sample window shows a new folder named Text Files, which includes .log files used by the project named Sales.
Tip: You can also customize file types and directories from a drop-down list. To access the drop-down list, click the down arrow next to the Filters button on the Explorer toolbar.

Procedure: How to Customize Directories for Virtual Folders

1. Highlight the Procedures folder in a project that has virtual folders and click the Filters button on the Explorer toolbar.
The Edit Filters dialog box opens.

2. The Location list box shows the directory paths associated with the project. By default, the boxes next to directory paths are checked. Click the box to deselect it if you want to remove a directory from the search path.

3. Click OK to apply the changes and close the Edit Filters dialog box.
Tip: You can also customize file types and directories from a drop-down list. To access the drop-down list, click the down arrow next to the Filters button on the Explorer toolbar.

Procedure: **How to Rename a Component**

1. Select and right-click the component. Select *Rename* from the context menu.
2. Type the new name at the cursor location.

Procedure: **How to Remove a Component**

1. Select and right-click the component. Select *Delete* from the context menu. If you are deleting a virtual folder, select *Delete Virtual Folder*.
2. Click Yes to confirm the permanent deletion.
Using Workspace Files

**How to:**
- Create a Workspace File
- Save Workspace Files
- Close a Workspace File
- Load a Workspace File
- Rename the Workspace File
- Change the WebFOCUS Environment for the Workspace File
- Change the Projects Assigned to the Workspace File
- Remove a Project From the Workspace File
- Insert a Project into the Workspace File

**Reference:**
- Properties Dialog Box of a Workspace File

You can control and limit projects (.GFA files) that are visible using Workspace files in the Projects area. Workspace files are created through Developer Studio and are stored on your PC. The Projects area is populated based on projects listed in the Workspace file. This is applicable when performing local or remote project-based development and enables you to organize your projects. In case of remote group development, you may view only your own projects. This enhances performance upon launching Developer Studio.

When using Workspace files, you may:
- Create a Workspace file using the New Workspace Wizard.
- Select an existing Workspace file and load it.
- Change the Workspace file name and provide a description for the Workspace file.
- Save a copy of the current workspace.
- Select an available project and remove it from the Workspace file.
- Add projects to the Workspace file.
- Create more than one Workspace file per environment.
- View properties for the workspace which enables you to change the workspace environment and change the projects assigned to the Workspace file.
The following image is an example of the Workspace file options available from the context menu.

![Workspace File Options](image_url)

**Note:** You may also select these options from the File menu.

**Procedure:** **How to Create a Workspace File**

1. From the Projects area in Developer Studio, select *Projects on localhost*.
2. Right-click and select *New Workspace* from the context menu.
The New Workspace Wizard- Page 1 of 2 opens.

3. Type a name for the Workspace file in the Workspace Name field or use the default Workspace name.

   **Note:** The Workspace file has a .GFW extension.

4. Change the default save location by selecting **Browse**.

   **Note:** This step is optional. The default location for Workspace files is `drive:\Documents and Settings\user_ID\My Documents\Workspaces\`.

5. Click **Next**.
The New Workspace Wizard-Page 2 or 2 appears where you may add or remove projects to your workspace.

6. Click Add to insert projects into the workspace.
7. Select from the list of available projects.
8. Double-click, or click OK, to add the project.

The project name is added to the New Workspace Wizard.
9. Click Finish.
   The Projects area refreshes, showing the selected Workspace file.
   **Note:** New projects are added and created in the Workspace file.

**Procedure: How to Save Workspace Files**

1. From the Projects area in Developer Studio, select the Workspace project name.
2. Right-click and select *Save Workspace As* from the context menu.
3. Type a file name for the Workspace file.
   **Note:** The Workspace file has a .GFW extension.
4. Click Save.
   The Projects area shows the selected Workspace file by default.

**Procedure: How to Close a Workspace File**

1. From the Projects area in Developer Studio, select the Workspace project name.
2. Right-click and select *Close Workspace* from the context menu.
   The Projects area refreshes to its original view, where all projects are visible.

**Procedure: How to Load a Workspace File**

1. From the Projects area in Developer Studio, select *Projects on localhost*.
2. Right-click and select *Load Workspace* from the context menu.
3. Navigate to the application folder location where the Workspace file is saved.
   **Note:** The default location for Workspace files is `drive:\Documents and Settings\user_ID\My Documents\Workspaces\`.
4. Select the workspace (.GFW) file.
5. Click Open.
   The Projects area refreshes, showing the selected Workspace file.

**Procedure: How to Rename the Workspace File**

1. From the Projects area in Developer Studio, select the Workspace project name.
2. Right-click and select *Rename Workspace* from the context menu.
3. Type in a name for the Workspace file.
4. Click the Enter key (on your keyboard) to retain the Workspace file name.

**Procedure: How to Change the WebFOCUS Environment for the Workspace File**

Changing the environment is useful if the original environment description name was changed, or if you want to use a different environment that has the same projects.

1. From the Projects area in Developer Studio, select the Workspace project name.
2. Right-click and select Properties from the context menu.
   
   The Properties dialog box opens.

3. Select the Workspace tab to change the target environment.

4. Click Target Environment to select the appropriate environment from the list.
5. Click OK.
   
   The target environment appears as the current environment for the Workspace file.

**Note:** If the environment name changes and it does not contain projects that exist in the Workspace file, the projects appear as unresolved in the Projects area.
**Procedure:**  **How to Change the Projects Assigned to the Workspace File**

1. From the Projects area in Developer Studio, select the Workspace project name.
2. Right-click and select *Properties* from the context menu.
   The Properties dialog box opens.
3. Select the *Workspace* tab to change the projects assigned to the Workspace file.
4. Click *OK* to close the Properties dialog box.

**Note:** You may also insert and remove projects from the Workspace file directly from the Projects area.

**Reference:**  **Properties Dialog Box of a Workspace File**

The Properties dialog box has the following fields/options:

**General tab**
Shows name, location, and other Workspace file information.

**Workspace tab**
Shows the environment and projects assigned to the workspace. You can change projects assigned to the workspace and change the WebFOCUS environment.

**Comment tab**
Enables you to add comments for the Workspace file.

**Procedure:**  **How to Remove a Project From the Workspace File**

1. From the Workspace file, select the project name that you are removing.
2. Right-click and select *Remove from Workspace* from the context menu.
   The project is removed from the Workspace file.

**Note:** You may also remove a project from the Workspace file by using the Workspace tab of the Properties dialog box.

**Procedure:**  **How to Insert a Project into the Workspace File**

1. From the Workspace file, select the Workspace file where you are inserting the project.
2. Right-click and select *Insert Project into Workspace* from the context menu.
3. Select a project folder from the available project lists.

**Note:** You may add multiple projects.
4. Click OK.

The Workspace File refreshes, showing the added project name.

**Note:** You may also insert a project into the Workspace file by using the Workspace tab of the Properties dialog box.

### Adding a Master File to a Project

**How to:**

- Add a Master File to a Project
- Remove a Master File From a Project
- Permanently Delete a Master File

Before you can create a procedure that reports against a data source, your application must understand how the data is organized. To obtain this information, your application reads a synonym, which generally consists of two files:

- **A Master File** that describes the data so WebFOCUS can report on it. This file contains field names and formats for the columns in the data source. The synonym also contains an alias for the data source.

- **An Access File** that contains additional information needed by WebFOCUS to access data. This information, which is required by many data sources, includes the data source’s real name and location.

Often, a synonym already exists on the server, and reporting can begin at once. However, if the synonym you need does not exist on the server, you can create it directly from Developer Studio using the Create Synonym tool.

There are several ways to provide metadata to a project:

- **Add visible files to a project.** When you designate one or more project paths in the Project Wizard, all of the Master Files in those paths are visible to your current project and can be made accessible for development and deployment.

  When you add a visible Master File to a project, Developer Studio creates an active link to the file; it does not copy the Master File to the project directory. See *How to Add a Master File to a Project* on page 50 for details.

- **Create synonyms.** Using the Create Synonym Tool, you can create synonyms for remote data sources configured with the WebFOCUS Reporting Server or for tables that reside on a subserver. For more information, see the *Describing Data With Graphical Tools* manual.
Enhancing a synonym. Intended primarily for use by a database administrator, the Synonym Editor can be used to view and edit attributes of synonym components. An application developer can use this tool to view the structure of a data source to make minor, authorized modifications to the Master File. The Synonym Editor also validates changes to a synonym and displays appropriate error messages. For more information, see the Describing Data With Graphical Tools manual.

Included in Developer Studio are several sample Master Files (.MAS) and corresponding data sources (.FOC). Optionally use these sample files to practice creating projects.

Procedure: How to Add a Master File to a Project

1. Open the Master Files folder in a project.

2. Click the Displays all files in the project path button on the Explorer toolbar.

   The right window pane displays a list of all Master Files in the paths defined for the selected project. Master Files that are not added to the project have a light gray icon next to them and the file type description is Master File Visible within Project Path.

3. Right-click a Master File from the available Master Files list in the right window pane and select Add to Project.

   or

   Select a Master File and click the Inserts the selected items into the project button on the Explorer toolbar.

   The Master File icon changes color to signify that it has been added to the list of Master Files in your project and the file type description changes to Master File.

Tip: You can add two or more Master Files at one time. Hold down the Ctrl key to select multiple Master Files.

Procedure: How to Remove a Master File From a Project

1. Open the Master Files folder in a project.

2. Click the Displays all files in the project path button on the Explorer toolbar.

   The right window pane displays a list of all Master Files in the paths defined for the selected project. Master Files that are added to the project path have a color icon next to them and the file type description is Master File.
3. In the right window pane, right-click a Master File and choose Remove from Project.

or

Select a Master File and click the Removes the selected items from the project button on the Explorer toolbar.

Click Yes to confirm removal of the Master File.

Tip: You can remove two or more Master Files at one time. Hold down the Ctrl key to select multiple Master Files.

Procedure: How to Permanently Delete a Master File

1. Select a Master File and click the Delete button on the Explorer toolbar.

2. Click Yes to confirm permanent deletion of the Master File.

Uploading Data Files

How to:
Upload a Data File

Reference:
Upload Data File Considerations

In Developer Studio, you can upload (import) external data files for use in WebFOCUS reporting tools. This functionality enables you to easily create a WebFOCUS file description and data file for use in your reporting application. The Upload Data File option is available from the Master Files folder while developing in the Projects or Data Servers areas.
Procedure: How to Upload a Data File

1. right-click the Master Files from the New menu, and select Upload Data File as shown in the following image.

The first page of the Upload Data File dialog box opens displaying three sections that require you to make a selection:

- Select a file
- File Format
- Field Names

2. Click the Browse button to the right of the Select a file section. A Choose file dialog box opens.

3. Navigate to where the file is located and select Open.
4. In the File Format section, select one of the following supported formats for the file you want to import:

- Excel Spreadsheet (XLS)
- Comma-separated values (CSV)
- Tab-separated values
- Pipe-separated values

5. In the Field Names section, select one of the following supported methods for how you want the field names created:

- Auto-generate field names
- First row contains field names

6. Click Next at the bottom of the Upload Data File dialog box.
7. For each field in the file you imported, you can select the field column heading and then edit the following attributes for that field:

- Field Name
- Alias
- (Data) Type
- Format

If you change any of the attributes for a field, click the Apply button to apply the changes and refresh the data.

8. When all fields have been reviewed, enter a valid name in the File Name input box (spaces are not allowed) and use the Application Directory menu to select where you want the file created.
9. Click Next to upload the file.

Two files are created:

- Master File
- Comma delimited data file (.DAT)

**Note:** If a file with the same file name already exists, a dialog box is displayed prompting you to allow file replacement.

**Reference: Upload Data File Considerations**

- When uploading data from a supported text file, the file must have the same type of data in each field and the same fields in every row.

- When uploading data from an Excel spreadsheet:
  - The data must be arranged in an appropriate tabular format and the spreadsheet must have the same type of data in each column and the same fields in every row.
  - The data must be stored in the first worksheet of the workbook.
  - Cells with formulas and special characters are not supported. For example, if percentages are used the cells should be formatted using the percentage data type and should not have the percentage special character "%" in the cell.
  - Excel files must be saved in a binary format. To ensure this, open an Excel file, select **File**, select **Save As**, use the Save as type drop-down list to select either **Microsoft Office Excel Workbook (*.xls)** or **CSV (Comma Delimited)/(*.csv)**, then click **Save**.
Viewing and Modifying Project Properties

How to:

View Project Properties
Apply Read-only Security
Modify Directory Paths
Customize Display of File Types
Add a Comment
Select a Deployment Scenario

Reference:

Project Properties Dialog Box
New Filter Dialog Box

You can refer to How to View Project Properties on page 57:

- **Apply read-only security.** When selected, this property prohibits modification to or deletion of a project. A project with read-only security displays a lock next to it in the Explorer.

- **Modify directory paths.** The Related Directories list contains directories in the project path. A project searches the directory paths in the order they appear on the list. You can add or remove a path, or change the search order.

- **Customize the display of items (file types) in a folder.** You can modify the file types that display in a folder in the Explorer.

- **Add a comment.** You can add a descriptive comment about a project.

- **Select a deployment scenario.** A deployment scenario defines the partitioning of the files and the selection of servers. For example, you might have two deployment scenarios for a project, one that maps the files to a production server and one that maps the files to a test server. You can associate a specific deployment scenario with a project.

- **Select a starting object.** You can flag an object to run first when a project is started.

You can also view project properties such as file type, location, size, and date of last modification.
**Procedure:** How to View Project Properties

1. Right-click a project and choose Properties.

   The Project Properties dialog box opens, displaying the General, Paths, Edit Filters, Comment, and Deployment tabs.

   The General tab is selected by default. It displays the project type, location, size, and date of last modification.

2. You can modify certain project properties using this dialog box.

**Procedure:** How to Apply Read-only Security

1. Right-click a project and choose Properties.

   The Project Properties dialog box opens.

2. On the General tab, click the Read-only check box.

3. Complete the dialog box as described in this topic. Click OK to apply changes and close the dialog box, or click Apply to temporarily apply changes and keep the dialog box open. Click Close to apply the temporary changes.

**Procedure:** How to Modify Directory Paths

1. Right-click a project and select Properties.

   The Project Properties dialog box opens.
2. Click the *Paths* tab.

![Sales Properties dialog box](image)

Make the following modifications to the directory paths:

- To add a new directory path, click the *Add new directory* button. Select from the available folders and click *OK*. The new directory is added to the Related Directories list.

- To remove a directory path, highlight it and click the *Remove directory* button. Click *Yes* to confirm the deletion.

- To reposition a directory path, highlight it and click the *Move Directory Up* or *Move Directory Down* button. These buttons are inactive for a single directory.

3. Complete the dialog box as described in this topic. Click *OK* to apply changes and close the dialog box, or click *Apply* to temporarily apply changes and keep the dialog box open. Click *Close* to apply the temporary changes.
**Procedure:** How to Customize Display of File Types

1. Right-click a project and choose Properties.
   The Project Properties dialog box opens.

2. Click the Edit Filters tab.
The Filters list box shows the file types associated with the project. You can make the following modifications to the list:

- To add an item, click the Add new file type filters button. The New Filter dialog box opens, listing the file types registered with the operating system.

Select a registered file type and click OK.

- To remove an item, highlight it and click the Remove file type filters button. Click Yes to confirm the deletion.

3. Click OK when you are done.

**Procedure: How to Add a Comment**

1. Right-click a project and select Properties.

   The Project Properties dialog box opens.
2. Click the Comment tab.

![Sales Properties](image)

3. Place the cursor in the Comment field, and type your comment.

4. Complete the dialog box as described in this topic. Click OK to apply changes and close the dialog box, or click Apply to temporarily apply changes and keep the dialog box open. Click Close to apply the temporary changes.

**Procedure: How to Select a Deployment Scenario**

1. Right-click a project and select Properties.

   The Project Properties dialog box opens displaying the General tab.

2. Select the Deployment tab.
3. On the Deployment tab, click the **Active Scenario** drop-down list, and select a scenario. The list contains all the scenarios created using the New Scenario option of the Deploy feature, and the default scenario (Local Deploy).

If you select a scenario other than the default scenario (Local Deploy), the **Target Application** field is enabled. This displays the application name to be created in the target deploy environment, or you can enter a new application name that will be used on the Web server and WebFOCUS Reporting Server deployment paths.

4. Click the **Starting Object** drop-down list, and make your selection. You can select any object from the list, but typically the starting object for a deployed application is an HTML file. The list contains all the valid starting objects created for this project.

5. Click **OK** to apply changes and close the dialog box, or click **Apply** to temporarily apply changes and keep the dialog box open. Click **Close** to apply the temporary changes.

In the Explorer, the icon for a starting object shows a down arrow, indicating that it will run first.

**Tip:** You can also designate a starting object by right-clicking it and choosing Set as Scenario **Starting Object**, or by highlighting it and selecting Set as Scenario **Starting Object** from the File menu.
**Reference:** Project Properties Dialog Box

**General Tab**

![Sales Properties Dialog Box](image)

**Type**

Is the type of project. Developer Studio supplies this value.

**Location**

Is the path in which the project and its associated files reside. Developer Studio supplies this value.

**Size**

Is the size of the project, in bytes. Developer Studio supplies this value.
**Modified**

Is the date and time the project was last changed. Developer Studio supplies this value.

**Read-only**

Prohibits modification to or deletion of a project.

**Run Remote**

Runs the project on a remote server.

**OK**

Confirms changes and returns to the Explorer.

**Cancel**

Aborts changes and returns to the Explorer.

**Apply**

Temporarily applies changes and keeps the dialog box open. Click Close to apply them permanently.
**Paths Tab**

![Sales Properties Image]

**Related Application Paths**

Are the directories in the project path. Developer Studio searches the directories in the order in which they appear on the list. Use the Add new directory, Remove directory, Move Directory Up, and Move Directory Down buttons to modify directory paths.

**Add New Directory**

Adds a directory to the project search path.

**Remove Directory**

Deletes a directory from the project search path.

**Move Directory Up**

Moves a directory up in the project search path.
**Move Directory Down**

Moves a directory down in the project search path.

**OK**

Confirms changes and returns to the Explorer.

**Cancel**

Aborts changes and returns to the Explorer.

**Apply**

Temporarily applies changes and keeps the dialog box open. Click **Close** to apply them permanently.
**Edit Filters Tab**

**Filters**

Are the file types associated with a project. Use the Add new file type filter(s) and Remove file type filter(s) buttons to modify the Filters list.

**Add New File Type Filter(s)**

Adds a filter.

**Remove File Type Filter(s)**

Removes a filter.

**OK**

Confirms changes and returns to the Explorer.
Cancel
Aborts changes and returns to the Explorer.

Apply
Temporarily applies changes and keeps the dialog box open. Click Close to apply them permanently.

Comment Tab

Comment
Type a descriptive comment for the project in the Comment field.

OK
Confirms changes and returns to the Explorer.
**Cancel**

Aborts changes and returns to the Explorer.

**Apply**

Temporarily applies changes and keeps the dialog box open. Click Close to apply them permanently.

**Deployment Tab**

![Sales Properties Dialog Box]

**Active Scenario**

Indicates how to partition the files. Select from the available scenarios in the drop-down list.
Target Application Name

If you select a scenario other than the default scenario (Local Deploy), this field is enabled. Otherwise, this field is disabled. Optionally enter an application name that will be used on the Web server and WebFOCUS Reporting Server deployment paths, or accept the default value (the current project name).

If you are deploying files to your development server, you must provide a target application name other than your development directory.

If you are deploying files to a WebFOCUS Reporting Server on z/OS, the target application name can be up to 8 characters. For deployment servers on all other supported platforms, the name can be up to 18 characters. Spaces are not allowed. (If a space is entered, it is converted to an underscore.)

Starting Object

Indicates the first component that runs. For a deployed application, the starting object is typically an HTML file. Select from the available objects in the drop-down list.

Smart Deploy

Choose this option to deploy only the files that have changed since the last time you deployed the application. This option can significantly increase the speed of deployment.

Ignore unresolved items

This option ensures that deployment is completed in a case where the project contains an unresolved file. An unresolved file can be any of the following: a file that once belonged to the project but was deleted outside of Developer Studio or from another project in which it was shared; a file located in a directory that changed and can no longer be found.

Compile Maintain Procedures

Choose this option to compile Maintain procedures during deployment. Compilation improves application performance. For more information, see the Developing WebFOCUS Maintain Applications manual.

Compile for verbose Maintain trace (Debug)

This option is reserved for debugging purposes. It can affect performance. For more information, see the Developing WebFOCUS Maintain Applications manual.

Include Server Paths

Choose this option to allow the deployment process to include server paths in the Call and Exec statements within Maintain procedures. Do not choose this option when deploying Maintain applications to platforms that do not support APP PATH commands (for example, VM).

For details about deployment, see Partitioning and Deploying Project Files on page 239.
Reference: New Filter Dialog Box

Registered File Types

Select a file type from those registered with the operating system, and click OK to add it to the Filters list on the Edit Filters tab.

**OK**

Confirms changes and returns to the Project Properties dialog box.

**Cancel**

Aborts changes and returns to the Project Properties dialog box.
Managing a Project With Source Control

Developer Studio supports third-party source code management products such as Microsoft Visual SourceSafe, ClearCase, CVS, Subversion, PVCS Version Manager from Merant, and others, that use a common industry standard API. Source code management (called source control in Developer Studio) provides version control for your individual and multi-user application resources, allowing multiple users to work on the same files. A source control product manages access to source code and keeps track of all code changes.

Developer Studio provides access to basic source control functions through menu options in the Developer Studio Explorer (The Developer Studio user interface for accessing files). If your site has installed a supported source control product, you can track the history of project code without leaving your development environment.

Source control is available from the Projects area in the Developer Studio Explorer, when performing local/stand-alone development or remote project development against a central WebFOCUS repository. This feature is also available when working from the Managed Reporting area of Developer Studio.

WebFOCUS Developer Studio Source Control feature uses the Source Code Control API (SCC API) that is utilized by the most popular Version Control Systems (VCS) on the market. For example, Microsoft Visual SourceSafe, Rational ClearCase, PVCS, and many more. The SCC API is an industry standard specification for Version Control Systems (VCS) connectivity defined by Microsoft.
Reference: Requirements for Source Control

- The client software for the Version Control System (VCS) that is used must be installed on the computer where Developer Studio is installed. The Version Control System (VCS) database can reside anywhere on the network and on any platform.

- The required client software can be installed after the Developer Studio installation.

- Multiple Version Control System (VCS) client software can be installed and the developer can select the provider they want to use. From the Developer Studio main menu, select Window and then Options. When the Developer Studio Options dialog box opens, click the Source Control tab. Select the required system based on the detected Source Control Systems.

How It Works

How to:

Determine If Source Control Is Enabled
Access Source Control Options
Use Source Control from Managed Reporting
Check Files In and Out of Source Control
View Source Control Options
Restore Write Access to a Single File
Restore Write Access to a Project

Reference:

Current Limitations for Source Control
Source Control Icons

When you start Developer Studio, it checks your system to determine if a supported source control product is installed.

If Developer Studio detects a supported product, it enables optional use of that product for the management of Developer Studio application resources.

Procedure: How to Determine If Source Control Is Enabled

1. In the Developer Studio Explorer, select Options from the Window menu.
2. From the Developer Studio Options dialog box, select the Source Control tab.
The source control system or systems detected by Developer Studio during startup are listed.

**Note:** If no systems are detected, this tab indicates that no Source Control providers are available.

3. If more than one system is listed, check the one you want to use, and click **OK**. Otherwise, Developer Studio will use the system that is listed and checked by default.

**Note:** If you do not want to use the Source Control feature, uncheck the selected system.

4. Click **Cancel** to close the dialog box and return to Developer Studio Explorer.

**Procedure:** How to Access Source Control Options

You can implement source control on a project, domain subfolder level or file level. Developer Studio writes information regarding source control in the project’s .GFA file when you select the option **Add to Source Control**. For Managed Reporting, a cookie is created at the domain level.

The following information pertains to using Source Control from the Projects area. For using Source Control from the Managed Reporting environment, see *How to Use Source Control from Managed Reporting* on page 75.

Depending on the way you access a source control option, that option is applied to a Developer Studio project (GFA), or to a single file or multiple files if selected. A project refers to an entire folder with all its contents (files).

If an object is selected (for example, a project or file), a source control option is applied to that object.

Access source control options in one of the following ways:

- Click the **Source Control** button on the toolbar.
- Select **Source Control** from the menu bar.
- Select and right-click **Projects on localhost** (or the WebFOCUS environment name that you are using), and select **Source Control** from the context menu.
- Select and right-click a project in the Projects area, and then select **Source Control** from the context menu.
- Select and right-click a virtual folder under a project, and then select **Source Control** from the context menu.
- Select and right-click a file in a virtual folder, and then select **Source Control** from the context menu.
**Procedure: How to Use Source Control from Managed Reporting**

From the Managed Reporting area of Developer Studio, you can add domain files to the Version Control System (VCS) database by creating a new project for the domain. You can also share files that are already available in the Version Control System (VCS) database.

**Note:** The examples and images in this procedure use Microsoft Visual SourceSafe.

The connection process and creation of the project on the Version Control System (VCS) database vary by product.

1. Select specific files from a Managed Reporting domain, or select a domain to add all files for that domain to the Version Control System (VCS) database.

2. Click the **Add to Source Control** icon in the toolbar. Individual files or groups can be added using the right-click context menu or the Source Control icon in the toolbar.

   You are prompted to connect to the Version Control System (VCS) database.

3. Log on to the Version Control System (VCS) database as follows:
   
   a. In the Username field, type a valid user name for the Version Control System (VCS) database.
   
   b. In the Password field, type a valid password for the Version Control System (VCS) database.
   
   c. In the Database field, type the full path of the Version Control System (VCS) database. You can also click **Browse** to navigate to the location of the Version Control System (VCS) database.
   
   d. Click **OK**.
Once authentication is successful, the Add to SourceSafe Project window opens, as shown in the following image.

4. Select the project directory and the project name. For Project name, it is recommended that you use the name provided by Developer Studio to avoid mapping issues between the Managed Reporting domain name and the Version Control System (VCS) project. It also helps developers match and identify the projects.

You can:

- Create the project at the root of the database. To do this, select the root of the tree "$/", provide a project name, and click OK.

- Optionally, organize projects to have the same structure as in the Object Explorer area of Developer Studio. For example, when the root of the tree is selected, you can type "WebFOCUS Environments", click Create, select the WebFOCUS Environments folder, and type a new name in the project field for the Environment name before clicking Create. This creates a subfolder under WebFOCUS Environments with the name you typed. Select the Environment name, type the project name for the domain, and click OK.

5. If this is a new project, a message appears asking whether you want to create the project. Click Yes.
A window opens that shows the files that will be added to the Version Control System (VCS) project you created.

6. Click OK.
The files and their Source Control status are visible in Developer Studio.

![Image of Developer Studio]

**Procedure:** How to Check Files In and Out of Source Control

To check out file(s), select the required file(s), right-click the selection, and select **Check Out**. A window with the files requested opens.

![Image of Check out file(s) dialog]

To check in file(s), select the required file(s), right-click the selection, and select **Check In**.
Procedure: How to View Source Control Options

You can view file properties or perform other tasks available from the Source Control menu.

1. Right-click a file that is added to Source Control.
2. From the context menu, select Source Control and click the required option (for example, Source Control Properties).

Reference: Current Limitations for Source Control

Files are not secured for updates when accessed outside of Developer Studio. To ensure files are not overwritten, all developers must use Developer Studio as their development tool.
Reference: Source Control Icons

The following table describes how Developer Studio uses the icons to indicate the status of a file in regard to source control.

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑ Controlled</td>
<td>The file has been added to source control.</td>
</tr>
<tr>
<td>- Controlled</td>
<td></td>
</tr>
<tr>
<td>✓ Checked Out</td>
<td>The file has been checked out of the source control database for editing.</td>
</tr>
<tr>
<td>- Checked Out</td>
<td></td>
</tr>
<tr>
<td>☑ Controlled &amp; Shared</td>
<td>The current Developer Studio project is sharing it with another Developer Studio project that has been added to source control.</td>
</tr>
<tr>
<td>- Controlled &amp; Shared</td>
<td></td>
</tr>
<tr>
<td>☑ Checked Out &amp; Shared</td>
<td>The file has been checked out of the source control database for editing.</td>
</tr>
<tr>
<td>- Checked Out &amp; Shared</td>
<td></td>
</tr>
<tr>
<td>☑ Exclusive Checked Out &amp; Shared</td>
<td>The file is being shared with another Developer Studio project. It has been checked out by the other project.</td>
</tr>
<tr>
<td>- Exclusive Checked Out &amp; Shared</td>
<td></td>
</tr>
<tr>
<td>☑ Not Controlled</td>
<td>The file has not been added to source control, or has been added and then removed.</td>
</tr>
<tr>
<td>- Not Controlled</td>
<td></td>
</tr>
</tbody>
</table>

Procedure: How to Restore Write Access to a Single File

1. In the Developer Studio Explorer, select and right-click the file. Select Properties on the context menu.
2. On the General tab on the File Properties dialog box, click the Read-only check box to remove the check mark.
3. Click OK to close the dialog box.

In the Developer Studio Explorer, notice that the attribute R (read only) has been removed from the file.
**Procedure: How to Restore Write Access to a Project**

1. Select and right-click the project.
2. Select Properties on the context menu.
3. On the General tab in the Project Properties dialog box, click the Read-only check box to remove the check mark.
4. Click OK to close the dialog box.

In the Developer Studio Explorer, notice that the attribute R (read only) has been removed from the project.

**Using ClearCase Integration as the Source Control Product**

In this section:

- ClearCase Setup Requirements
- How to:
  - Set ClearCase User Options
- Reference:
  - ClearCase Notes and Recommendations

This section describes the relation between ClearCase and WebFOCUS applications accessed and created by Developer Studio and provides recommendations regarding the configuration of ClearCase and Developer Studio.

**ClearCase Setup Requirements**

ClearCase requires setting up a development view to access files that are under ClearCase management. The following views are available:

- **Snapshot Views** denote a physical location on a hard drive.
  
  **Note:** Some of the user interface requests are available through the Windows Explorer ClearCase context menu.

- **Dynamic Views** are client logical views of VOB’s. VOB is a repository for storing versioned projects and metadata.
  
  **Note:** Through Windows Explorer, limited operations are allowed to be performed on the set of exposed files. For instance, a checked in file under ClearCase control cannot be removed via Windows Explorer commands or other Windows API call. The same thing is true for changing the Red-Only flag.
Snapshot Views work while disconnected from the network. Views are updated when you perform manual requests. However, this can cause views to become out of date. The file loaded during checkout may not necessarily be the latest version in the VOB.

Dynamic Views are always up to date and require that you are connected to the network. If you are not connected, network issues will arise while attempting to use Developer Studio, ClearCase Explorer, or even Windows Explorer, as the Dynamic view will not be loaded.

To use the Developer Studio Source Control feature with ClearCase, you must associate the WebFOCUS applications (residing in a local or remote WebFOCUS Environment) to a ClearCase view.

**Note:** Other Source Control systems support the concept of Current Working Directory, which in fact is an ad-hoc association of a local directory to a Source Controlled project or directory. In other words, to an SCC project/directory could be associated any physical, reachable location, where all SCC operations will take place. For ClearCase, only the designated view is the one to be recognized by ClearCase as the location (physical or virtual) where the SCC operations could take place.

From Developer Studio, applying a Source Code Control operation on a file found on a remote WebFOCUS Environment involves an HTTP copying from the WebFOCUS application to the view directory, and/or the other way around. Depending on the view type, these operations could also involve modifying files attributes and/or modification timestamp.

**Tip:** Since the view is just a file transfer place between the ClearCase system and the WebFOCUS application, some problems arise with the Snapshot views as files added to the view may be out of date, or may already be checked in by another developer. Given all these limitations, Dynamic Views are recommended.

**Procedure: How to Set ClearCase User Options**

It is recommended to set the following ClearCase user options for easier operation.

1. From the ClearCase Explorer Tools menu, select *ClearCase Options* from the Options submenu.

2. In the Check In/Add to Source Control section of the Operations tab, select the *Use original checkout information* and *Preserve file modification time* options.

3. In the Check Out section of the Operations tab, select the *Preserve file modification time* option.

   **Note:** Other options can be customized as necessary.
The following image is the ClearCase Explorer with these default options set.

Reference: ClearCase Notes and Recommendations

The following notes and recommendations should be taken into consideration when using ClearCase as the Source Control.

- Parallel development (multiple checkouts of the same file) is supported by ClearCase and automatic merge technology is available from ClearCase.

  Note: ClearCase can handle most merging aspects, there are however cases that conflicts may arise that will require intervention by the developer. To avoid issues that may occur as a result of merging, single stream development is recommended.

- Developers working with WebFOCUS files and applications must be aware that this behaves like a shared file system. Any modification of a WebFOCUS file could be seen by all other developers using that specific file. Working with the file is taking place at the WebFOCUS location where that file resides. The Source Code Control system just imposes restrictions on accessing that specific file.
Using CVS Integration as the Source Control Product

How to:
Start the CVS Configuration Tool
Set the CVS User Options

This section describes the relation between CVS and WebFOCUS applications accessed and created by Developer Studio and provides recommendations regarding the configuration of CVS and Developer Studio.

The CVS integration requires that a third party product (plug-in) is used to provide the SCC Compliance that is used by Developer Studio to connect to the Version Control System’s (VCS) repository.

Note: Although you may use any CVS plug-in, this section is specific to a product provided by PushOK.

You will have to setup and configure the PushOK CVS Plug-in and specify the PushOK CVS Proxy properties.

Procedure: How to Start the CVS Configuration Tool

1. Install the PushOK CVSSCC NT plug-in.
2. From the start menu, select CVSSCC NT from the PushOK Software menu.
3. Select Configure Plugin from the CVSSCC NT submenu.

The PushOK CVS Proxy properties dialog box appears.

Procedure: How to Set the CVS User Options

When you start the CVS Configuration tool, the PushOK CVS Proxy properties dialog box appears. Use the configuration tabs to set the recommended CVS user options.

1. The Files Types tab enables you to specify how different file types are treated.
To enforce a certain behavior for a certain file type, enter the extension in the appropriate column to add or remove text files, binary files, or unicode files.

2. The CVS executables tab enables you to specify custom applications to be used with the plug-in.

- It is recommended to leave the use specified option unchecked so that PushOK CVS acts as the CVS client. This allows for more functionality than the regular freeware CVS client.

- Type in or select a CVS GUI viewer application in the GUI executable field. This is necessary in the case that you would want to use a GUI application instead of the command line to query the CVS repository.

   **Note:** PushOK CVS does not come with a built-in CVS GUI interface.
Select a **Diff/Merge** tool.

**Note:** PushOK CVS comes with a built-in Diff/merge tool however it sometimes behaves unexpectedly.

Select a file viewer tool from the **Viewing** field.

**Note:** PushOK CVS provides a built-in file viewer but you may also specify a custom tool.

The following image is an example of the CVS executables tab on the PushOK CVS Proxy properties dialog box.
3. The CVS options tab enables you to specify how files are treated by CVS and how the status of these files are reported.

**Note:** It is recommended to set the following CVS options, as they appear in the image below, in order to avoid conflicts.
4. The Known lists tab enables you to administer the CVSROOT strings to be used by the PushOK CVSSCC client, and to add/remove CVS modules (directories under the CVS repository) and tags.

   a. To Add a New CVSROOT String:

      - Select Known Roots from the Select list type drop-down list.
      - Type in the CVSROOT string in the input field under the List section.
      - Click Add.

      The string is added to the List section box.

      **Note:** Repeat this step for adding multiple strings.

      - Select the string that you want to use as your default and click Set As Default.

      The string will be marked with the (Default) suffix and will be used in future operations with your integrated development environment (in this case Developer Studio).

   b. To Add New Projects:

      Although most of the projects (modules) will be done from the integrated development environment, PushOK CVS allows you to add new projects.

      - Select Known Modules from the Select list type drop-down list.
      - Type in the name of the module in the input field under the List section. This is the relative path to the repository directory.
      - Click Add.

      The module is added to the List section box.

      **Note:** There is no default module option.
The following image is an example of the Known lists tab on the PushOK CVS Proxy properties dialog box.

![Image of Known lists tab on PushOK CVS Proxy properties dialog box]

5. The **Server options** tab enables you to specify how the connection to the server is being handled.

- Select a string from the **Select CVSROOT** drop-down list.
  
  **Note:** It is recommended to select the default string to avoid confusion. If the selected string is not the default, type in the desired string and click **Default**.

- Select the **This server is CVSNT ver >= 2.058** option if the CVS server you intend to connect with is of a version higher or equal to 2.058. If not, leave this option unchecked.

- The **SSH Options** are only enabled when using :ext: protocol.

- The **Encryption** option is only enabled for protocols which support encryption. For example, sspi.

- Select the **Check filenames on server** option if a UNIX machine hosts the CVS server. This will resolve possible conflicts of file name case sensitivity.
Managing a Project With Source Control

The following image is an example of the Server options tab on the PushOK CVS Proxy properties dialog box.
Using Subversion (SVN) Integration as the Source Control Product

**How to:**

Start the Subversion (SVN) Configuration Tool
Set the Subversion (SVN) User Options

This section describes the relation between Subversion (SVN) and WebFOCUS applications accessed and created by Developer Studio and provides recommendations regarding the configuration of Subversion and Developer Studio.

Subversion does not have native support for the SCC API, however there are software vendors that provide SCC API plug-ins to allow development tools such as Developer Studio to access the Subversion repositories and use functions that are available from Subversion.

The Subversion (SVN) integration requires that a third party product (plug-in) is used to provide the SCC Compliance that is used by Developer Studio to connect to the Version Control System’s (VCS) repository.

**Note:** Although you may use any Subversion plug-in as your Source Control, this section is specific to PushOK SVN configuration.

You will have to setup and configure the PushOK SVN Plug-in and specify the PushOK SVN Proxy properties.

**Procedure:** How to Start the Subversion (SVN) Configuration Tool

1. Install the PushOK SVN plug-in.
2. From the start menu, select SVNSCC from the PushOK Software menu.
3. Select Configure Plugin from the SVNSCC submenu.

The PushOK Subversion (SVN) Proxy properties dialog box appears.
Procedure: How to Set the Subversion (SVN) User Options

When you start the Subversion (SVN) Configuration tool, the PushOK SVN Proxy properties dialog box appears. Use the configuration tabs to set the recommended Subversion user options.

1. The SVN executables tab enables you to set tools used by the SVN SCC API to expose standard functionality such as running differences, repository viewers, O/S integration tools, and so on.

   - Type in or select a SVN GUI viewer application in the GUI executable field. Since the PushOK SVN SCC client does not provide its own standalone viewer, you can specify a viewer by passing it to the current connection string. This is necessary in the case that you would want to use a GUI application instead of the command line to query the SVN repository.

     Note: If before installing the plug-in you have TortoiseSVN installed on your machine, PushOK SVN SCC plug-in installer will detect it and populate this field for you.

   - Select a Diff/Merge tool.

     PushOK SVN offers the possibility to use its internal comparison tool by selecting the SVN conflict editor (built-in) option in the Diff/Merge box. You may also specify an external tool from this field.

   - Select the Conflicts resolving tool. When checking in a file and the conflicts cannot be resolved, a resolving visual tool is shown as set here. This can be provided by PushOK SVN or it can be provided externally.

   - Select a file viewer tool from the Viewing field.
The following image is an example of the SVN executables tab on the PushOK SVN Proxy properties dialog box.
2. The SVN **options** tab enables you to specify how files are treated by Subversion and how the status of these files are reported.

**Note:** It is recommended to set the following SVN options, as they appear in the image below, in order to avoid conflicts.
3. The Known lists tab enables you to administer the SVN connection strings to be used by the PushOK SVNSCC client, and to add/remove SVN modules (directories under the SVN repository) and tags.

a. To Add a New SVN connections string:

- Select Known Roots from the Select list type drop-down list.
- Type in the SVN string value in the input field under the List section.
- Click Add.

The string is added to the List section box.

The following image is an example of the Known lists tab on the PushOK SVN Proxy properties dialog box.
Setting Up Developer Studio With ClearCase, CVS, Subversion (SVN), or ChangeMan DS

**How to:**

Set the Developer Studio Source Control Options
Set the SCC_Name Variable in the WebFOCUS Client

**Reference:**

Validating the Source Control Variable in WebFOCUS

After ClearCase, CVS, Subversion (SVN), ChangeMan DS, or any other required client software is installed and configured, there are several steps involved in setting up Developer Studio to work with your Source Control provider. You must set the Developer Studio Source Control options and validate and configure the Source Control variable in the WebFOCUS Client Configuration file.

**Procedure: How to Set the Developer Studio Source Control Options**

- For ClearCase, install and configure the ClearCase Client software on the PC where Developer Studio is installed and create the required Dynamic View(s) attached to the VOB you are interested in.

  **Note:** The drive assigned to a view does not need to be the same for all developers, and more than one view can be created.

- For CVS, Subversion (SVN), and ChangeMan DS, install and configure the client software or the required plug-in, as described in the CVS and Subversion sections. For details, see Using CVS Integration as the Source Control Product on page 84 and Using Subversion (SVN) Integration as the Source Control Product on page 91.

1. When Developer Studio starts, it checks to see if any supported Version Control Systems are available.

   If supported systems are detected they will be visible in the Developer Studio Options dialog box, under the Source Control tab, and the product will activate Source Control commands in the main menu, context menu, toolbar and the Developer Studio Explorer.
The following image is an example of the Developer Studio Options dialog box with ClearCase selected.

**Note:** If more than one version control systems is detected, ensure that ClearCase is selected.
Managing a Project With Source Control

The following image is an example of the Developer Studio Options dialog box with PushOK CVSSCC NT selected.
The following image is an example of the Developer Studio Options dialog box with Subversion (PushOK SVNSCC) selected.
The following image is an example of the Developer Studio Options dialog box with ChangeMan DS selected.

2. Select the *Requires Custom Development Directory* option to enable the Settings button.

3. Click the *Settings* button.

   The SCC Development Folders dialog box appears.

   **Note:** There may be a delay before the next dialog is shown.

4. From the SCC Development Folder dialog box, type or select a directory in the *Default Development Directory* field.

   **Note when using ClearCase:** This directory should be an existing directory inside a Dynamic view. To avoid errors, it is recommended that you browse to select the development directory. The directory specified in the Development Directory will be used as the root directory for all ClearCase local operations.
Note when using CVS: This is the directory where all CVS files transfers will take place before or after the CVS actions files can be copied to or from their real location in this directory.

Note when using Subversion (SVN): This is the directory where Subversion (SVN) will store the files specified in the Source Control Directory. By default, the value is the directory specified in the Default Development Directory.

Note when using ChangeMan DS: This is usually a directory on the local PC where the development work takes place and it is used to store files that are checked in/out of the version control systems repository.

The following image is an example of the SCC Development Directory dialog box with the Default Development Directory selected.

5. Click OK to save and close the SCC Development Folder dialog box.
After setting the Developer Studio Source Control Options, a validation message appears, as in the example image shown below.

Reference: Validating the Source Control Variable in WebFOCUS

Version Control products that require a Development Directory (for example: ClearCase, CVS, Subversion, ChangeMan) need to also have a Source Control variable set in the WebFOCUS Client of the WebFOCUS Environment that is used for development. This variable, SCC_NAME, is used by the WebFOCUS Developer Studio Source Control feature to identify the development environment.

When specifying values for the variable SCC_NAME:

- The variable is usually set to the name of the machine (it should be different than the SCC_NAME of all other systems that contribute files to the Source Control repository). An example of using the machine’s host name would be SCC_NAME=dev-server-1.

- The value needs to be unique for every WebFOCUS environment accessing the same Source Control repository, and the name should not contain spaces or any of the following special characters: ./ \ " ` ? : * < > |

Note: Avoid changing the value once files have been added to the Source Control repository. If you need to change the value, either all files will have to be added to the repository again, or the repository will need to be changed to be consistent with the new value. This is because the SCC_NAME value is used to establish the file identity in the repository. Changing the value will require that all files are added to the repository again; or that the repository be changed to be consistent with the new value.
**Procedure:** **How to Set the SCC_Name Variable in the WebFOCUS Client**

You may set the Source Control variable from the WebFOCUS Administration Console, or by manually adding it to the WebFOCUS Client configuration file.

1. From the WebFOCUS Administration Console, click *General* under the Configuration section.
   a. Identify the development environment for the Source Control in the SCC_NAME field.
      
      **Note:** By default, there is no value set for this field.
   b. Provide a name for the SCC_Name variable.
   c. Click Save to save the Client Settings.

The following image is an example of the WebFOCUS Administration Console showing the SCC_NAME Field.

![WebFOCUS Administration Console SCC_NAME Field](image-url)
2. To manually add the variable, you will have to edit the WebFOCUS Client configuration file (CGIVARS.WFS).

**Tip:** You may use this step when using older versions of Developer Studio.

a. Locate and open the CGIVARS.WFS file.

   For example, the configuration file may be located in \
   ibi\WebFOCUS71\client71\wfc\etc.

b. Add an entry under the <!WFConsole begin General> section called SCC_NAME and identify the development environment for the Source Control as the machine name.

   For example, the SCC_Name variable may appear as:

   ```
   <!WFConsole begin General>
   APPROOT=C:/ibi/apps
   SCC_NAME= machinename
   ...
   ```

c. Save and close the CGIVARS.WFS file.

d. You must restart the web and application server after editing this file.

   **Note:** If the unique identity of the environment could not be established based on the SCC_NAME then the HTML Alias could be used as an alternative. The HTML Alias is also useful when more than one WebFOCUS environments are setup on the same server. This enables you to specify which WebFOCUS Environment(s) to be used. For all other cases, HTML Alias does not need to be altered.
Source Control Options

In this section:
Get Latest Version
Check Out
Check In
Undo Check Out
Add to Source Control
Remove From Source Control
Show History
Show Differences
Source Control Properties
Share From Source Control
Refresh Status
Source Control
Unbind From Source Control
Open From Source Control

This topic describes the source control options available from Developer Studio.

Note: These options depend on the Version Control System (VCS) utilized.

Get Latest Version
Select this option to view a file without changing it. This option retrieves the latest version of the selected file and provides you with a read only copy.
If you attempt to make changes, they are not saved.

Check Out
To make changes to a file in source control, you must first check it out of the database.
The Check Out option retrieves the latest version of the selected file and allows you to edit it. Source control removes the attribute R (read only) from a checked out file.
A file that has been checked out displays the description Checked Out in the Developer Studio Explorer.
Check In
Select this option to return a checked out file to the source control database. Source control stores any changes made to the file in the database.

Undo Check Out
Select this option to reverse a check out and cancel any changes made to a file in your working area.

Add to Source Control
Before you can add Developer Studio files to source control, you must have a source control project in which to place them. Therefore, to implement source control, the first option that you will usually choose is Add to Source Control.

When you add a Developer Studio project (GFA) to Microsoft Visual SourceSafe for example, SourceSafe creates a corresponding project (folder) in which to store the files that get sourced, using the same name as the Developer Studio project. A project refers to an entire folder with all its contents (files).

A file added to source control displays the description Controlled in the Developer Studio Explorer.

In this example, the HTML files have been added to source control.

Remove From Source Control
Select this option to remove a file from the source control database.

A file that has been removed from the database, or one that has not been added to the database, displays the description Not Controlled in the Developer Studio Explorer.
Show History
Select this option to view information about a file, such as version history, date of creation or modification, and other details.

Show Differences
Select this option to display the differences between a local copy of a file and the copy stored in the database.

Source Control Properties
Select this option to view or edit certain file properties. For example, you can view the checkout status (including the user, date of checkout, and version), comments, and other file properties.

Share From Source Control
Select this option when you are working in a Developer Studio project and wish to use files from a different Developer Studio project that has been added to the source control database.

A file that is shared with another project displays the description Controlled & Shared or Checked Out & Shared in the Developer Studio Explorer.

A shared file that is checked out by one project displays the description Exclusive Checked Out & Shared in the other project.

Refresh Status
Select this option to refresh the display of file status in the Developer Studio Explorer.

Source Control
This option opens the source control product. It is available in one of the following ways:

- Click the Source Control button on the toolbar and select Source Control from the drop-down menu.
- Select Source Control on the menu bar and select Source Control from the drop-down menu.

Unbind From Source Control
Select this option to disassociate a Developer Studio project from source control. Developer Studio removes the information regarding source control from the project’s .GFA file (it added the information when you selected Add to Source Control).

This feature is available from the Source Control option on the menu bar or the Source Control menu when you select and right-click Projects on localhost.
Open From Source Control

Select this option to create a local Developer Studio project (GFA) based on a project in the source control database. It enables a developer to get a project that has already been created and added to source control by another developer.

It is available from the Source Control option on the menu bar or the Source Control menu when you select and right-click Projects on localhost.

Reference: Notes for Using MicroSoft Visual SourceSafe as the Source Control Product

- When you add a Developer Studio project to Microsoft Visual SourceSafe, that project is assigned to a corresponding SourceSafe project. If one Developer Studio project shares files from a second Developer Studio project, you must also add the second project to SourceSafe. For example, if a Developer Studio project named Sales uses files from baseapp, you must add the baseapp project to SourceSafe. If you do not, the shared files cannot be managed by source control.

- When you add files to Visual SourceSafe, SourceSafe assigns the attribute read only (R) to them. When you check out a file, SourceSafe removes the read only attribute to allow edits to the file.

  When you remove files from SourceSafe, SourceSafe retains the read only attribute originally assigned. You must manually change the attribute to restore write access. See How to Restore Write Access to a Single File on page 80 and How to Restore Write Access to a Project on page 81 for instructions.

- To use source control effectively, you must be familiar with the specific source control product that you have and its database structure. Your source control administrator can give you information on the setup at your site.

Securing a Project

How to:

Secure a Project

Some Master Files or procedures in a project may have security restrictions applied to them. To see the data or portion of the data in a Master File or procedure you have access to, you must enable your password.

Enabling your password also allows you to encrypt data to secure information in a procedure or Master File, or decrypt data for use with the Developer Studio tools. For information on encryption and decryption, see the Describing Data With Graphical Tools manual. For additional information on security, see the WebFOCUS Security and Administration manual.
Procedure: **How to Secure a Project**

1. Highlight a project and choose Password from the Command menu.
2. Enter the password to secure procedures or Master Files in the current project, and click OK.

Removing a Project

**How to:**
Remove a Project

When necessary, you can remove a project from the Explorer. The .GFA control file is deleted from the default directory. You also have the option of deleting the associated project files from the directory.

Procedure: **How to Remove a Project**

1. Select the project.
2. Choose one of the following methods:
   - Right-click the project and select Delete from the context menu. You are asked if you want to delete the .GFA control file and, optionally, the associated files. Respond to the prompts as desired.
   - or
   - Press the Delete key. You are asked if you want to delete the .GFA file and, optionally, the associated files. Respond to the prompts as desired.
Removing a Project
Creating a Reporting Procedure

A procedure is the core element of a reporting application and is always created within the context of an application. It specifies the data to be retrieved from a data source and the format in which data is presented to the user. A procedure can consist of a single component or multiple components, in any meaningful combination.

The most common procedure with one component is a report or graph. A procedure that includes multiple components may contain Joins, virtual fields (Defines), environment settings, as well as a report or graph. When a procedure is run, the components are executed in the order in which they are specified.

More complex procedures may contain calls to other procedures, nested procedures, data extracted to an intermediate (HOLD) file for additional reporting, and WebFOCUS code such as Dialogue Manager commands to handle flow of control.

Topics:
- Selecting a Creation Tool
- Incorporating a Procedure Into an Application
- Copying a Procedure
- Creating a Procedure Component
- Working With a Component
- Using the SQL Report Wizard
- Assigning a Logical Name With the Allocation Wizard
- Calling a Procedure From the Current One
- Using the Engine Tool
- Managing Flow of Control
- Working With a Full Procedure
- Running a Procedure
- Canceling a Running Procedure
Selecting a Creation Tool

How to:
Select a Creation Tool
Create a Report Procedure With the Report Painter
Create a Reporting Procedure Using the Procedure Window
Create a Procedure With the Text Editor

Reference:
WebFOCUS Logon Dialog Box
Add Procedure Dialog Box

You can create a procedure with one of the following tools:

- **Procedure Viewer.** Provides a graphical environment for the creation of a generic procedure. This option is the default.
- **Report Painter.** Provides a graphical environment for the creation of a report procedure.
- **Document Composer.** Provides a graphical environment for the creation of a compound report.
- **SQL Report Wizard.** Assists you with SQL passthru, which allows you to execute SQL code that retrieves data from an RDBMS. You can use the resulting extract file in the Report Painter or the Graph Assistant.
- **Graph Assistant.** Allows you to transform almost any type of data into an effective graph that you can customize.
- **Text Editor.** Provides a standard text editor, allowing you to create a procedure by entering commands.
**Procedure: How to Select a Creation Tool**

1. Open a project.

2. Create a new procedure by doing one of the following:
   - Right-click the Procedures folder, select New, and click Procedure.
   - or
   - Highlight the Procedures folder, select New from the File menu, and click Procedure.

The Add Procedure dialog box opens.

![Add Procedure Dialog Box](image)

The display box lists all the procedures currently visible in the project path, or all procedures that have been added to the project path (QTY_ALL.fex in the sample dialog box). Click the Displays all files in the project path button to toggle between the two displays.

3. In the File name field, type a name for the new procedure. Do not include any spaces in the name or add an extension; Developer Studio automatically appends the extension .fex.

4. In the Files of type drop-down list, accept the default, Procedure Files (*.fex).
5. In the Create with drop-down list, select the creation tool:

- **Procedure Viewer** provides a graphical environment for the creation of a generic procedure. For information, see *How to Create a Reporting Procedure Using the Procedure Window* on page 115.

- **Report Painter** provides a graphical environment for the creation of a report procedure. For information, see *How to Create a Report Procedure With the Report Painter* on page 115.

- **Document Composer** enables you to design reports, and to coordinate and distribute layouts made up of multiple reports and graphs in a single file.
  
  For details, see *Creating Reports With the Document Composer* in the *Creating Reports With Graphical Tools*.

- **SQL Report Wizard** assists you with SQL passthru, which allows you to execute SQL code that retrieves data from an RDBMS. You can use the resulting extract file in the Report Painter or the Graph Assistant. For information, see *Using the SQL Report Wizard* on page 130.

- **Text Editor** provides a standard text editor, allowing you to create a procedure by entering commands. For information, see *How to Create a Procedure With the Text Editor* on page 116.

- **Graph Assistant** accesses the Graph Assistant, where you can transform almost any type of data into an effective graph that you can customize.
  
  For information, see the *Creating Charts With Graph Tools* manual.

**Reference:** **WebFOCUS Logon Dialog Box**

When you access either the Projects folder or the WebFOCUS Environments folder and you need to access a secure server, a logon box displays, prompting you for an ID and password.

There is a check box option for remembering the ID or password. By default, it is not checked. If you check this box, your credentials will be stored and encrypted in the wfscom.wfs file, the local configuration file that is used to store information processed by the Developer Studio communication layer. For details about the wfscom.wfs file, see *Managing the WebFOCUS Environment* in the *Developing Reporting Applications* manual. If you decide to change the stored credentials later, you can do so from the WebFOCUS Environments Properties dialog box.
Procedure:  How to Create a Report Procedure With the Report Painter

1. In the Add Procedure dialog box, enter a name for the new procedure in the File name field, then select Report Painter from the Create with drop-down list, and click Open. For details on the Add Procedure dialog box, see Add Procedure Dialog Box on page 117.

The Open dialog box prompts you for the name of the Master File to use.

The display box lists all the Master Files currently visible in the project path, or all the Master Files that have been added to the project path. Click the Displays all files in the project path button to toggle between the two displays.

2. Select a Master File from the display box to add it to the File name field.

3. In the Files of type drop-down list, accept the default, Master Files (*.mas).

4. Choose Open Report Painter and click Open to start the selected tool.

You can now build the report procedure. See the Creating Reports With Graphical Tools manual or the Creating Reports With WebFOCUS Language manual for information.

Procedure:  How to Create a Reporting Procedure Using the Procedure Window

1. In the Add Procedure dialog box, enter a name for the new procedure, then select Procedure Viewer, and click Open.
The Procedure window opens.

![Procedure window with a Comment component]

This window contains a default Comment component with two diamond objects on either side called component connectors. The Comment component does not contain any executable commands. At this point, you have created a procedure shell. You can now make this shell executable by adding components, such as a report or graph.

Components are stored in a file with the extension .FEX.

2. Click and hold a diamond object and select a component, such as Report or Graph. For details on how to proceed, see Creating a Procedure Component on page 120.

**Procedure: How to Create a Procedure With the Text Editor**

1. In the Add Procedure dialog box, enter a name for the new procedure, select Text Editor, and click Open. The text editor opens.

![Text editor window]

2. Enter the syntax for your procedure.

   See the Creating Reports With WebFOCUS Language manual for details on WebFOCUS syntax.

3. Click the Save button on the text editor toolbar when you are done. See Editing Application Components as Text in Developer Studio on page 269 for details on the text editor.
Reference: Add Procedure Dialog Box

The Add Procedure dialog box contains the following fields/options:

**Look in**
Is the name of the project.

**Procedure List Box**
Lists the procedures associated with the selected project.

**File name**
Is the name of the procedure.

**Files of type**
Is the type of file.

**Create with**
Is the creation tool to be used. The options are:

- Procedure Viewer
- Report Painter
- Document Composer
- SQL Report Wizard
Incorporating a Procedure Into an Application

How to:
Add a Procedure to a Project
Remove a Procedure From a Project

When you create a procedure in the current project, it is automatically incorporated into that project. However, you are not limited to the use of procedures created in the project. You can make procedures from another project available in the current project.

To incorporate a procedure from another project into the current project, complete the following steps:

1. **Add the procedure's directory to the project path.** When you add a procedure's directory to the project path, all project files become available. For details on modifying available directories for a project, see *How to Modify Directory Paths* on page 57.

2. **Add a procedure to the project.** Once a procedure's directory is visible to a project, add the procedure to the project so it will be available for reporting and deployment. For details, see *How to Add a Procedure to a Project* on page 118.

**Note:** You can also remove a procedure from a project. For details, see *How to Remove a Procedure From a Project* on page 119.

**Procedure:** How to Add a Procedure to a Project

1. Open the Procedures folder in the project.

2. If not already selected, click the *Displays all files in the project path* button on the Explorer toolbar.

   The right window pane displays all procedures in the paths defined for the selected project. Procedures that are not added to the project path have a light gray icon next to them, and the file type description is Focus Executable Visible within Project Path.
3. Do one of the following:

- Right-click a procedure from the available procedures on the right, and select Add to Project from the shortcut menu.

  or

- Select a procedure and click the Inserts the selected items into the project button on the Explorer toolbar.

  The procedure icon changes color to signify that it has been added to the project, and the file type changes to Focus Executable.

Tip: Clicking the Displays all files in the project path button on the Explorer toolbar toggles between the files that are visible in the project path and the files that have been added to the project.

**Procedure: How to Remove a Procedure From a Project**

1. Open the Procedures folder in the project.

2. If not already selected, click the Displays all files in the project path button on the Explorer toolbar.

   The right window pane displays all procedures in the paths defined for the selected project. Procedures that have been added to the project path have a color icon next to them, and the file type description is Focus Executable.

3. Do one of the following:

   - In the right window pane, right-click a procedure and choose Remove from Project from the pop-up menu.

     or

   - Select a procedure and click the Delete button on the Explorer toolbar.

4. Click Yes to confirm permanent removal of the procedure.
Copying a Procedure

How to: Copy a Procedure Within a Project

You can drag and drop any component of a project, including a procedure, between projects or WebFOCUS environments, or within a single project. When a procedure is copied within a project, it is identified as Copy of procedure_name. A second copy is identified as Copy (2) of procedure_name, and so on.

Procedure: How to Copy a Procedure Within a Project

1. Open the Procedures folder for a project.
2. Right-click the procedure to be copied, and select Copy from the pop-up menu.
3. Right-click the Procedures folder, and select Paste from the pop-up menu.

Creating a Procedure Component

How to: Create a Component in an Existing Procedure

Reference: Component Types

After creating a procedure shell, you can create components that make the shell executable. The Component Connector toolbar contains the available types of components. The Comment component is created by default and is not included on the toolbar.

Several of the components on the toolbar are usually placed in a specific order based on the logical flow of a procedure. These components are Set, Allocation, Use, Join, Define, and Report or Graph, in that order. The remaining components—Execute, Include, and Other—enhance the user interface and efficiency, and can be placed anywhere in the order of execution. However, Dimension must be placed before the Report component.
You can have one or several components in a procedure. The following example includes a Comment, four instances of a Report, a Use, Define, and Other components. Together these components make up the procedure named APP6.

### Component Types

**Reference:** Component Types

You can include the following components in a procedure.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Creates a tabular representation of the data retrieved from a data source. A Report is placed after a Join and a Define. For details, see the Creating Reports With Graphical Tools manual.</td>
</tr>
<tr>
<td>SQL Report</td>
<td>The SQL Report Wizard assists you with SQL passthru, which allows you to execute SQL code that retrieves data from an RDBMS. You can use the resulting extract file in the Report Painter or the Graph Assistant. For details, see Using the SQL Report Wizard on page 130.</td>
</tr>
<tr>
<td>Define</td>
<td>Creates a temporary (virtual) field that is evaluated before a report is run. A Define is placed before a Report. For details, see the Creating Reports With Graphical Tools manual.</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Graph         | Creates a graphical representation of the data retrieved from a data source.  
A Graph is placed after a Join and a Define. For details, see the Creating Charts With Graph Tools manual.|
| Advanced Graph| A powerful graph tool that provides a user-friendly, easy-to-navigate interface with advanced functionality for creating and editing basic and complex graphs.  
For details, see the Creating Charts With Graph Tools manual.|
| Set           |Customizes the Developer Studio and WebFOCUS environments, including the way reports and graphs display, report and graph content, data retrieval characteristics that affect performance, system responses to end user requests, metadata setup, and date manipulation.  
For details on the Set tool and parameters, see Customizing Your Environment in the Developing Reporting Applications manual.|
| Join          |Defines a relationship between two or more data sources so that a report can use data from all of them at once.  
A Join is placed before a Define, and if there is no Define, it is placed before a Report. For details, see the Creating Reports With Graphical Tools manual.|
| Use           |Provides flexibility in using a FOCUS data source. You can specify read-only access, treat multiple data sources as one, and request the use of data sources not stored in your current directory (whether elsewhere on your hard disk or on a network file server).  
For details on accessing FOCUS data sources, including techniques for maintaining Use commands, see Accessing a FOCUS Data Source in the Developing Reporting Applications manual. |
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocation</td>
<td>Assigns a temporary name to a file created and used by WebFOCUS. To preserve an extract file (created by the HOLD command) on the server’s disk, you must issue a FILEDEF command to a specific location. Also, the Dialogue Manager -READ and -WRITE commands refer to files by names that must first be established with a FILEDEF command. For details on the Allocation Wizard, see Assigning a Logical Name With the Allocation Wizard on page 138.</td>
</tr>
<tr>
<td>Execute</td>
<td>Enables one procedure to call another. You can select a procedure to be called by the one you are editing. An Execute component can be placed anywhere in a procedure. For details on the Execute component, see Calling a Procedure From the Current One on page 156.</td>
</tr>
<tr>
<td>Include</td>
<td>Allows you to use a portion of a procedure if you store it separately. A separate, included procedure is treated as part of the procedure currently being edited. It enables the reuse of code, making projects easier to write and maintain. After an included procedure runs, control returns to the calling procedure. An Include component can be placed anywhere in a procedure. For details on the Include component, see Calling a Procedure From the Current One on page 156.</td>
</tr>
<tr>
<td>HtmlForm</td>
<td>Creates a text file you can use to add HTML code to the procedure currently being edited. HTML forms are used to launch applications in the Web environment. <strong>Note:</strong> The Deploy Wizard guides you through the process of creating a configuration that manages the deployment of your project files, including HTML forms, to the Web. For more information on the Deploy Wizard, see the Developer Studio Application Development Getting Started manual.</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Other           | Enables you to enter commands that connect other components. This feature is useful for writing code that does not have a graphical tool (such as Dialogue Manager). However, you can also use Other for any kind of manual coding, for example, to code a virtual field or a report. When used this way, the Other component automatically converts to the corresponding component type.  
An Other component can be placed anywhere in a procedure.  
For information on the Other component, see the *Developer Studio Application Development Getting Started* manual. For details on the WebFOCUS language, see the *Creating Reports With WebFOCUS Language* manual. |
| Olap Dimensions | Creates an OLAP hierarchy and dimensions based on data for multi-dimensional analysis without changing the Master File. The new logical view is saved as part of the procedure.  
A Dimension component applies to FOCUS data sources and relational tables. It must be placed before the Report component.  
For information on defining dimensions, see the *Describing Data With Graphical Tools* manual and the *Describing Data With WebFOCUS Language* manual.  
Unlike the Dimension Builder (addressed in the *Describing Data With Graphical Tools* manual), the Dimension Tool, accessed through the Dimension component, does not show the graphical representation of a Master File; rather, it lists the fields. The process of creating a hierarchy is the same; drag and drop fields from the left pane to the right pane. |
| SQL Editor      | The SQL Editor enables you to code SQL Passthru and highlights any SQL commands within the code. This editor also allows you to choose your connection engine and connection name from drop-down lists on the toolbar.  
For more information on the SQL Editor, see the *Developer Studio Application Development Getting Started* manual. |
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match</td>
<td>The Match Wizard enables you to merge two or more data sources, and specify which records to merge and which to sort out. The wizard creates a new data source, a HOLD file, into which it merges fields from the selected records. You can report from the new data source and use it as you would any other HOLD file. For more information on the Match Wizard, see the <em>Developer Studio Application Development Getting Started</em> manual.</td>
</tr>
<tr>
<td>Engine</td>
<td>The Engine tool exposes the FOCUS ENGINE SET commands and enables you to enter ENGINE commands or connection attributes, and override parameters. <strong>Note:</strong> The Engine tool is only used to create ENGINE statements. You are responsible for having the knowledge of the ENGINE SET command or the Connection Attribute needed. For details see the <em>Using the Engine Tool</em> on page 164.</td>
</tr>
<tr>
<td>Dialogue Mngr</td>
<td>The Dialogue Manager enables you to control the flow of your application with the use of variables. For more information on the Dialogue Manager, see <em>Managing Flow of Control</em> on page 165.</td>
</tr>
<tr>
<td>Modify</td>
<td>Changes or modifies data. There is no Modify tool but you can use the Other tool to create a Modify request by typing code, at which point the Other tool changes to a Modify component. For details on Modify, see <em>Unique FOCUS Topics</em> on the WebFOCUS Documentation CD.</td>
</tr>
<tr>
<td>Comment</td>
<td>Includes a comment for the project. This comment is not necessary and you can delete it if you wish. A procedure shell, when created, contains a default Comment component. Once you create an executable component for a procedure, the file name defaults as a comment in this component. For example, if you named the shell TEST, the default Comment component will contain the comment .:* File TEST.FEX. For information on the Comment component, see the <em>Developer Studio Application Development Getting Started</em> manual.</td>
</tr>
</tbody>
</table>
Procedure: **How to Create a Component in an Existing Procedure**

1. Right-click a procedure in the Procedures folder and choose *Open*. The Procedure window opens.

2. Click and hold a component connector. The Component Connector menu opens as shown in the following image.

3. Select the tool for the component you wish to create. You can also select a tool from the static Component Connector toolbar that displays in the Procedure window. The corresponding tool opens. For details, see *Component Types* on page 121.

4. After you have constructed a component using a tool, exit the tool. You are asked if you want to update the procedure. Answer yes, and your changes are saved and the component is added to the procedure as a graphical object of the type created (Report, Graph, Define, Join, and so on).
Working With a Component

You can perform the following tasks on a component:

- **Running.** You can run some types of components, for example, a report or graph. This shows you how the output appears from a visual point of view.

- **Editing.** You can edit a component within a graphical tool or with the text editor.
  
  Using the text editor requires that you have some knowledge of the WebFOCUS language. You must be familiar with the types of commands, how to use them, and where to place them in a request.

- **Checking.** Checking a component enables you to locate errors. Each component tool includes a check button that checks the syntax of the code against the server. For more information, see *How to Check a Component* on page 128.

- **Copying.** You can copy a component from one location to another within the same procedure, or from one procedure to another.

**Procedure: How to Run a Component**

1. Open a procedure in the Procedures window.
2. From the Procedure window:
   - Right-click the component and select *Run* from the shortcut menu.
   - or
   - Select the component and then click the *Run* button on the Main toolbar.
How to Edit a Component

Procedure:

1. From the Procedure window, choose one of the following methods to open a component:
   - Double-click the component.
   - Right-click the component and choose Open.

2. Make your changes to the selected component.

3. Exit the tool by clicking the control menu box. When asked if you want to update the procedure, click Yes. Your changes are saved.

Tip: You can edit a component with the text editor. Right-click the component and choose Edit Text. The text editor opens, displaying the component’s syntax. For information about the text editor, see Editing Application Components as Text in Developer Studio on page 269.

How to Check a Component

Procedure:

1. From the Procedure window, choose one of the following component tools:
   - Define
   - Set
   - Use
   - Allocation
     The appropriate tool opens.

2. Add your information to the tool, and click Check.
The Check window opens, displaying the code produced by the tool and any errors that exist. The source code for your component and any errors that may exist are displayed. The following is a Check window for the Define tool.

3. Click OK to return to your component tool.

**Procedure: How to Copy a Component From One Procedure to Another**

1. Open both procedures by right-clicking them and choosing Open from the shortcut menu.
2. Choose Tile from the Window menu to view both Procedure windows simultaneously.
3. In the first Procedure window, click and hold the component you wish to copy.
4. While holding the left mouse button, click the Ctrl key on your keyboard.
5. Drag the mouse pointer to the second Procedure window.
   
   **Note:** You must click the Ctrl key to copy. Omitting to click Ctrl before dragging moves the component.
6. Release the mouse button when the mouse pointer is over the second Procedure window. The component is copied to the second window.
When you drag and drop a component, its icon changes to show whether its target location is valid. If it is not valid, the cursor resembles a circle with a slash through the word Drop. When you have dragged the component to a valid location, the cursor resembles a small square with a plus sign inside it.

**Tip:** You can move a component from one procedure to another by holding the left mouse button, and dragging the mouse pointer to the second Procedure window. Holding down the Ctrl key copies the component, whereas clicking and dragging moves the component.

### Using the SQL Report Wizard

**How to:**

- Include SQL Commands From an External .sql File
- Pass SQL Commands to the RDBMS Using SQL Passthru
- Import SQL Commands From an Existing .sql File
- Add an .sql Extension as a Valid Filter

The SQL Report Wizard assists you with SQL passthru, which allows you to execute SQL code that retrieves data from an RDBMS. You can use the resulting extract file in the Report Painter or the Graph Assistant. The supported engines for the SQL Report Wizard are DB2, DB2 for System i, Microsoft SQL Server, Oracle, Sybase, and Teradata.

The SQL Report Wizard is available throughout all development areas of Developer Studio: Projects, Data Servers, and Managed Reporting. When working in Managed Reporting, the tool enables administrators to use SQL in Standard Reports available in the Domain or use procedures that reside on the WebFOCUS Reporting Server.

**Note:** It is recommended to use the Create Synonym Tool to execute a SQL Command for a stored procedure. For more information see the reference topic, Synonym Creation Parameters for Stored Procedures, in the Accessing Data and Creating Synonyms chapter of the Describing Data with Graphical Tools manual.

**Procedure:** How to Include SQL Commands From an External .sql File

1. Access the SQL Report Wizard by doing one of the following:
   - In the Explorer, right-click the Procedures folder and select New, Procedure. In the Add Procedure dialog box enter a file name and then select SQL Report Wizard from the Create with drop-down list. Click Open.
   - In the Procedure Viewer, click the component connector and then click SQL Report on the drop-down list.
You can click the SQL Report icon on the Procedure Viewer toolbar.
The SQL Report Wizard - Welcome window opens.

2. Click the option button to *Included from an external '.sql' file*. This enables you to browse and select external procedures that exist in the DB2 Web Query repository of the project. This enables sites to leverage pre-existing SQL procedures. It is not possible to execute procedures from Managed Reporting.

*Note:* You can only browse the files on the APP PATH if your .sql extension is a valid filter in the Properties dialog box. To do this, see *How to Add an .sql Extension as a Valid Filter* on page 137.
3. Click Next to see the SQL Report Wizard - Data access information window.

4. In the Select the SQL database engine area, select a database engine from the drop-down list. The list consists of available engines in the edasprof.prf file.

5. In the Select the connection area, choose a connection from the drop-down list generated from the engine that you selected. You can choose the default value, which is the first connection in the edasprof.prf file for the selected engine, or choose another connection defined in edasprof.prf.
6. Click Next to see the SQL Report Wizard - Include external SQL file window.

7. Enter the external SQL file name in the field, or click Browse to select it.
   
   **Note:** You can only browse the files on the APP PATH if your .sql extension is a valid filter in the Properties dialog box. To do this, see *How to Add an .sql Extension as a Valid Filter* on page 137.

8. Optionally, you can run with limited records by clicking Run SQL. By default, the *Run with limited records* check box is selected so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. 100 is the default limit.

9. Click Next to see the SQL Report Wizard - Summary of SQL options window.
   
   Do one of the following:
   
   - To create a report, select the *Create Report* option button. This option is selected by default.
   - To create a graph, select the *Create Graph* option button.

10. Click *Finish* to run the SQL procedure. The Report Painter or Graph Assistant opens. When you have completed the procedure, you can run it from the Procedures folder in the Explorer view.
Using the SQL Report Wizard

Procedure: How to Pass SQL Commands to the RDBMS Using SQL Passthru

1. Access the SQL Report Wizard by doing one of the following:
   - In the Explorer, right-click the Procedures folder and select New, Procedure. In the Add Procedure dialog box enter a file name and then select SQL Report Wizard from the Create with drop-down list. Click Open.
   - In the Procedure Viewer, click the component connector and then click SQL Report on the drop-down list.
   - You can click the SQL Report icon on the Procedure Viewer toolbar.
   The SQL Report Wizard - Welcome window opens.

2. Click the option button next to Type SQL statements in the report request. This enables you to enter SQL commands that will be passed on to the RDBMS with the SQL Passthru feature.

3. Click Next to see the SQL Report Wizard - Data access information window.

4. In the Select the SQL database engine area, select a database engine from the drop-down list. The list consists of available engines in the edasprof.prf file.

5. In the Select the connection area, choose a connection from the drop-down list generated from the engine that you selected. You can choose the default value, which is the first connection in the edasprof.prf file for the selected engine, or choose another connection defined in edasprof.prf.
6. Click Next to see the SQL Report Wizard - Enter SQL statements window.

7. In the field box, type the SQL statements you want to pass to the RDBMS.

8. Optionally, you can run with limited records by clicking Run SQL. By default, the Run with limited records check box is selected so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. 100 is the default limit.

9. Click Run SQL to run your report.

10. Click Next to see the SQL Report Wizard - Summary of SQL options window.

    Do one of the following:

    - To create a report, select the Create Report option button. This option is selected by default.

    - To create a graph, select the Create Graph option button.

11. Click Finish to run the SQL procedure. The Report Painter or Graph Assistant opens. When you have completed the procedure, you can run it from the Procedures folder in the Explorer view.
**Procedure: How to Import SQL Commands From an Existing .sql File**

1. Access the SQL Report Wizard by doing one of the following:
   - In the Explorer, right-click the Procedures folder and select New, Procedure. In the Add Procedure dialog box enter a file name and then select SQL Report Wizard from the Create with drop-down list. Click Open.
   - In the Procedure Viewer, click the component connector and then click SQL Report on the drop-down list.
   - You can click the SQL Report icon on the Procedure Viewer toolbar.

   The SQL Report Wizard - Welcome window opens.

2. Click the option button next to Import from an existing .sql file. This enables you to modify SQL code after importing it from an external file to the procedure being built. It enables you to modify the request using bits of code.

3. Click Next to see the SQL Report Wizard - Data access information window.

4. In the Select the SQL database engine area, select a database engine from the drop-down list. The list consists of available engines in the edasprof.prf file.

5. In the Select the connection area, choose a connection from the drop-down list generated from the engine that you selected. You can choose the default value, which is the first connection in the edasprof.prf file for the selected engine, or choose other connections defined in edasprof.prf.

6. Click Next to see the SQL Report Wizard - Import external SQL file window.
7. Type the SQL file name that you want to import or click Browse to select it.

   **Note:** You can only browse the files on the APP PATH if your .sql extension is a valid filter in the Properties dialog box. To do this, see *How to Add an .sql Extension as a Valid Filter* on page 137.

8. Optionally, you can run with limited records by clicking Run SQL. By default, the *Run with limited records* check box is selected so you can test your procedure with a readlimit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. 100 is the default limit.

9. Click Run SQL to run your report.

10. Click Next to see the SQL Report Wizard - Enter SQL statements window. You can edit the imported SQL code if necessary.

11. Click Run SQL to run your report.

12. Click Next to see the SQL Report Wizard - Summary of SQL options window.

   Do one of the following:

   - To create a report, select the *Create Report* option button. This option is selected by default.
   - To create a graph, select the *Create Graph* option button.

13. Click Finish to run the SQL procedure. The Report Painter or Graph Assistant opens. When you have completed the procedure, you can run it from the Procedures folder in the Explorer view.

**Procedure:** *How to Add an .sql Extension as a Valid Filter*

To browse the files on the APP PATH, your .sql extension must be a valid filter in the Properties dialog box.

1. Right-click the project, select Properties, then select the Edit Filters tab.

2. Scroll through the list until you find the .sql extension in the Extensions column.
   
   If the .sql file is not part of the displayed list in the Extensions column, click the Add new filter file type filter(s) icon.

3. Select the .sql extension.

4. Click OK.
Assigning a Logical Name With the Allocation Wizard

In this section:
Clearing Allocations

How to:
Define a Logical Name With the Allocation Wizard

Reference:
Advanced Allocation Dialog Box

For a file managed by the operating system, such as an ISAM or comma-delimited data file, the physical file name is the actual name of a file as it appears to the operating system. A logical name (or ddname) is a shorthand name that points to the physical file name. Logical names simplify code by allowing short names to be used in place of the longer physical file name.

The FILEDEF command assigns a logical name to a physical file name and specifies file attributes. You can explicitly define a file and its location to WebFOCUS using the Allocation Wizard. The Allocation Wizard generates platform independent file paths for all portable platforms by creating FILEDEF syntax with application names. An Allocation can be issued in a procedure and lasts for a single request.

It is recommended that instead of including an Allocation in each procedure, you include all FILEDEF commands in a single file that you call with the -INCLUDE command at the beginning of each procedure. This enables you to make changes to your FILEDEF commands globally instead of changing the Allocation information in each procedure.

The FILEDEF command is typically used in the following ways in operating systems that support this command:

- **Identifying data sources.** WebFOCUS Developer Studio automatically creates FILEDEF assignments for .DAT files and temporary files (.FTM) in the current search path. You must create FILEDEF assignments (or Use directories, in the case of FOCUS data sources) for all other data sources you wish to use.

- **Redirecting input and output.** Three ddnames are used for input and output. By reassigning these ddnames, you can redirect input or output:
  - SYSIN for input.
  - SYSPRINT for output displayed on the screen.
  - OFFLINE for output sent to the printer.
You can also use the Universal Naming Convention (UNC) to assign logical names to files that are located on a server. In order to take advantage of the UNC you must first attach to the server you want to use. For information on attaching to a server or mapping network drives, consult your Network Administrator.

Procedure: How to Define a Logical Name With the Allocation Wizard

1. Right-click the procedure in the Procedures folder and choose Open from the shortcut menu.

   The Procedure (FOCEXEC) window opens.

2. Click and hold a component connector (yellow diamond), and select Allocation from the Component Connector toolbar.

   The Allocation Wizard opens.

3. Click Next and specify the logical name for the allocation.
The name can be from one to eight characters. If you are identifying a data source, the name must be the same as the name of the Master File.

4. Click Next and select the device for the logical name.
   - Disk associates the logical name with a file. This is the default option.
   - Printer associates the logical name with a printer.
   - Http associates the logical name with a URL, by allocating the Master File to the result of running the URL.
   - Terminal indicates that the keyboard and monitor are the input source and output destination for the file.
Clear clears the allocation assigned to a file name.

5. Click Next to select additional options.

   **Note:** The options vary depending on which device is selected. The steps below detail how the Allocation Wizard continues with the selected device.

   **Tip:** At any time, you may click Back to go back and change the device, or any of the device settings.

6. When selecting Disk as the device:
   
a. Select the application folder that contains the physical file.
**Note:** The application folders from the Reporting Server are shown by default and depend on the area from which the Allocation Wizard is invoked. For example, Local Projects, Data Servers, or Managed Reporting. You may specify a file in a directory that is not in the Reporting Server Application Path by using the Advanced option from the completed Allocation. For more information, see *Advanced Allocation Dialog Box* on page 153.
b. Click Next and enter the new file name.

![Allocation Wizard](image1)

or

Click the Browse button to display the Open dialog box and select from the list of existing files.

![Allocation Wizard](image2)

**Note:** The default Files of type is Data Files (*.dat). You may also select Temporary Files (*.ftm), and All Files (*.*), as the type of file.

c. Click Next to specify additional options for the Disk device.
- Appends records to the end of the file (without this option, the file is overwritten).
- Fixed length records indicates that you are assigning a logical name to a file with a fixed record length.
- Record Size specifies the record length of the file, when using the Fixed length records option.
d. Click Next to view the summary of the allocation settings you selected for the Disk device.

7. When selecting Printer as the device:
   a. Select the application folder that contains the physical file.
**Note:** The application folders from the Reporting Server are shown by default and depend on the area from which the Allocation Wizard is invoked. For example, Local Projects, Data Servers, or Managed Reporting. You may specify a file in a directory that is not in the Reporting Server Application Path by using the Advanced option from the completed Allocation. For more information, see Advanced Allocation Dialog Box on page 153.
b. Click Next and enter the new file name.

![Allocation Wizard](image)

or

Click the Browse button to display the Open dialog box and select from the list of existing files.

![Allocation Wizard](image)

**Note:** The default Files of type is Data Files (*.dat). You may also select Temporary Files (*.ftm), and All Files (*.*), as the type of file.
c. Click Next to enter the name of the printer.

![Allocation Wizard](image)

Enter the name of the printer:

\bprint\28C1

---

d. Click Next to view the summary of the allocation settings you selected for the Printer device.

![Completing the Allocation Wizard](image)

Logical Name: USEFLST
Device: Printer
Application: dbamp
Filename: dmpers.dat
Printer: \bprint\28C1

To close this wizard and create a FILEDEF command based on the settings above, click Finish.
**Note:** The following syntax is applied to the FILEDEF statement when using a Printer device:

```plaintext
FILEDEF OFFLINE PRINTER BASEAPP/TEMPOFFLINE.FTM (PRINTER \ibiprint\28C1
```

8. When selecting Http as the device:
   a. Enter the full URL path.
Assigning a Logical Name With the Allocation Wizard

b. Click Next to view the summary of the allocation settings you selected for the Http device.

9. When selecting Terminal as the device:
   a. Click Next to specify additional options for the Terminal device.
You may choose to send the keyboard input to WebFOCUS as entered (lowercase or mixed-case). The default is mixed-case. Select this option to specify lowercase.

Click Next to view the summary of the allocation settings you selected for the Terminal device.
10. When selecting Clear as the device:
   a. Click Next to view the summary of the allocation settings that you cleared.

11. Click Finish to close the Allocation Wizard and create the FILEDEF command. The completed file name appears for the Allocation, as shown in the image below.
12. Optionally, click **Check** to validate the FILEDEF command.

A dialog box opens, displaying the component code and confirmation that no error exists.

13. To edit the selected allocation, double-click, or click **Edit**, to open the Allocation Wizard again.

14. Click **Advanced** to open the Advanced Allocation dialog box.

The Advanced Allocation dialog box enables you to enter the FILEDEF command without using the Allocation Wizard. For details, see *Advanced Allocation Dialog Box* on page 153.

15. Click **OK** to close the completed Allocation Wizard.

The Allocation component is added to the procedure. You may double-click the component to view the completed Allocation Wizard options again.

**Reference:** *Advanced Allocation Dialog Box*

When you click **Finish** from the Allocation Wizard, the Allocation Wizard dialog box displays the completed logical allocation name that you created. Click **Advanced** to open the Advanced Allocation dialog box to enter the FILEDEF command without using the Allocation Wizard.

**Note:** When you create a FILEDEF command with the Advanced option, the Allocation is added to the procedure as a component. You may double-click the component to view or edit the command in the Advanced Allocation dialog box.

### Name

Shows the logical (defined) name of a file or device (one to eight characters).
Assigning a Logical Name With the Allocation Wizard

**Device**

Identifies the type of device to associate with the logical name.

**Disk** associates the specified logical name with a file.

**Printer** associates the specified logical name with a printer.

**Http** associates the Hypertext Transfer Protocol link, by allocating the Master File to the result of running the URL.

**Terminal** indicates that the keyboard and monitor are the input source and output destination for the file.

**Clear** clears the FILEDEF command assigned to a file name. (Select the logical name in the list below the Name box.)

**File name**

Shows the full file name. It may include a drive and directory specification.

Enter a file name in the box, or click the **Browse** button to open a dialog box and select the file.

**Note:** You may use the File name field to specify a directory that is not in the Reporting Server Application Path.

**Fixed**

Indicates that you are assigning a logical name to a file with a fixed-record length.

Enter the length of the record in the Record Size box.

**Record Size**

Specifies the record length of the file.

**Append**

Appends records to the end of the file. Without this option, the file is overwritten.

**Lower Case**

Retains the case (lowercase or mixed-case) of keyboard input in the Command Console.

**OK**

Closes the Advanced Allocation dialog box and adds the FILEDEF command to the completed Allocation Wizard dialog box.

**Cancel**

Closes the Advanced Allocation dialog box without saving the allocation settings.
Help
Opens the online documentation for this topic.

Check
Displays the FILEDEF command in code, and specifies any errors.

New
Moves the cursor into the Name box for a new FILEDEF command entry.

Delete
Deletes the logical name you select from the list below the Name box.

Clearing Allocations

**How to:**
Clear an Allocation with the Allocation Wizard
Clear an Allocation with the Advanced Allocation Dialog Box

You can clear Allocations by using the Allocation Wizard and with the Advanced Allocation dialog box.

**Procedure:**  **How to Clear an Allocation with the Allocation Wizard**

1. Double-click the Allocation component to view the completed Allocation.
2. Select the Allocation name and double-click, or click Edit, to open the Allocation Wizard.
   **Note:** The Allocation opens with the tool you used to create the FILEDEF command, in this case, the Allocation Wizard.
3. Click Next twice, for the device options section of the Allocation Wizard, and select Clear.
4. Click Next and Finish to close the Allocation Wizard.
5. Click OK from the completed Allocation Wizard dialog box.

**Procedure:**  **How to Clear an Allocation with the Advanced Allocation Dialog Box**

1. Double-click the Allocation Component to view the completed Allocation.
2. Select the Allocation name and double-click, or click Edit, to open the Advanced Allocation dialog box.
   **Note:** The Allocation opens with the tool you used to create the FILEDEF command, in this case, the Advanced Allocation dialog box.
3. Select the Clear radio button under the Device section.
4. Click OK to close the Advanced Allocation dialog box.
5. Click OK from the completed Allocation Wizard dialog box.

**Calling a Procedure From the Current One**

**How to:**
- Call a Procedure With the Include Component
- Execute a Procedure With the Execute Wizard

The Execute Wizard and the Include component enable you to call another procedure from the current one.

The Execute component allows one procedure to execute or call another procedure. The redesigned Execute Wizard allows you to view and select available procedures and supply values for parameters in the called procedure and also test that called procedure. The called procedure behaves as a completely separate procedure, with its own context.

Execute Wizard allows you to select the procedure that you are supplying parameter values for. Once this procedure is specified, Execute Wizard displays these parameters and prompts you to provide values for them. You can provide values for all parameters in the procedure, or can choose to provide values for only some of the parameters. If you provide values for only some of the parameters, you must provide values for the other parameters using another method. (For example, values are passed from another part of the application, etc.) After providing parameter values, you can test the called procedure.

The Execute Wizard is available throughout all development areas of the product: Projects, Data Servers, and Managed Reporting. When working in Managed Reporting, the tool allows developers to use Standard Reports available in the Domain or use procedures that reside on the WebFOCUS Reporting Server.

Execute Wizard uses the WFDESCRIBE auto-prompting feature to locate and pass parameters. To activate the WFDESCRIBE feature, change the default value in the IBIF_wfdescribe setting in the cgivars.wfs file.

The Include component allows one procedure to run another procedure as if the second one were embedded in the first. In this case, the procedure being included (called) has full access to variables defined in the calling procedure. Using this tool, you can create an object that includes another procedure within a host procedure. There is no limit to the number of procedures that can be included.
**Procedure:** How to Call a Procedure With the Include Component

1. Open the host procedure.
2. Click and hold the component connector at the point at which you wish to include the existing procedure.
   The Component Connector toolbox opens.
3. Click the **Include** button on the toolbox.
   The Open dialog box opens.
4. Select the procedure you wish to embed from the list, or type the name of a new procedure in the File Name field.
5. Click **Open**.

**Tip:** The Include component inserts the -INCLUDE command into the procedure's code. Right-click the Include object and choose **Quick View** from the shortcut menu to see the -INCLUDE command.

---

**Procedure:** How to Execute a Procedure With the Execute Wizard

1. Click the component connector at the point at which you wish to include the existing procedure.
   The Component Connector toolbox opens.
2. Click the **Execute** button.
The Execute Procedure Wizard opens.
3. Select the procedure you want to execute from the list of available procedures, and click Next. The Execute Procedure Wizard - Enter Procedure Parameters window opens:

The Enter Procedure Parameters window contains a list of variables contained in the procedure you are executing, and fields in which to enter values for those variables. This screen will not appear if the procedure you are calling does not have parameters.

4. Enter the variable values you wish to use in the procedure you are executing. If you do not wish to provide a value for a variable, uncheck the Optional box next to the variable name.

5. Click Next.
The Execute Procedure Wizard - Summary window opens. This window allows you to view the name of the procedure you are calling and view the variables-value pairs.

6. Click Test Procedure to ensure that the procedure runs properly. If the procedure works properly, a report will display in your browser.

7. Click Finish to insert the selected procedure into the current procedure.

Tip: The Execute component inserts the EX command in the procedure’s code. Hover your cursor over the Execute object to see the EX command. If you wish to run the embedded procedure, right-click the Execute object and choose Run from the shortcut menu.

Example: Calling a Procedure From Another Procedure

In the following example, you will create two procedures. One procedure, named SALESREP, will contain a sales report with variables. The second procedure, names SVALUES, will contain the values for the variables in SALESREP, and will execute SALESREP. The result will be a sales report with variable values filled in.

Create the SALESREP procedure:

1. Create a new procedure named SALESREP using the SALES data source.

3. In the Page Heading section, type Monthly Report for and then select Variable Field from the Insert menu. The Variable Editor opens.

4. Create the &CITY variable by doing the following:
   a. Enter CITY in the Name input field.
   b. Select Single Select from the Variable Type drop-down list.
   c. Select Values from field from the Data Context drop-down list.
   d. Select CITY from the Fields from database drop-down list.
   e. Select ALL from the Values from field CITY box.
   f. Click OK.

5. Add the UNIT_SOLD and RETURNS fields to your report, then select both fields in the layout and click the Sum button in the toolbar.

6. Create the RATIO field by doing the following:
   b. Enter RATIO in the Field input box.
   c. Enter D5.2 in the Format input box.
   d. Enter 100 * (RETURNS/UNIT_SOLD) in the Expressions window.
   e. Click OK.

7. Add the PROD_CODE field to your report, and click By.

8. Filter the PROD_CODE field by doing the following:
   b. Select the IF tab.
   c. Enter the following in the Expressions window: IF PROD_CODE IS-FROM &CODE1 TO &CODE2
   d. Click OK.

9. Add the CITY field to your report, and click By.

10. Filter by the CITY field by doing the following:
b. Select the IF tab.
c. Enter the following in the Expressions window: \textit{IF CITY EQ \&CITY}
d. Click OK.

11. Save and close the procedure.

Create the SVALUES procedure:

1. Create a new procedure named SVALUES and from the Create with drop-down list, select \textit{Procedure Viewer}. The Procedure Viewer opens.

2. Click a component connector, and select \textit{Execute}. The Execute Wizard opens.

3. Select SALESREP from the list of procedures, and click Next.

   The Enter Procedure Parameters window opens.

4. Enter \textit{STAMFORD} in the Value field for CITY.

5. Enter \textit{B10} in the Value field for CODE1.

6. Enter \textit{B20} in the Value field for CODE2.

7. Click Next.
The Summary window opens.

8. Click Finish. You return to the Component Connector toolbox.

9. Close and save the procedure.

When the SVALUES procedure is run, the following report appears.

MONTHLY REPORT FOR STAMFORD

<table>
<thead>
<tr>
<th>PROD CODE</th>
<th>CITY</th>
<th>RETURNS</th>
<th>UNIT SOLD</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>B10</td>
<td>STAMFORD</td>
<td>10</td>
<td>60</td>
<td>16.67</td>
</tr>
<tr>
<td>B12</td>
<td>STAMFORD</td>
<td>3</td>
<td>40</td>
<td>7.50</td>
</tr>
<tr>
<td>B17</td>
<td>STAMFORD</td>
<td>2</td>
<td>29</td>
<td>6.90</td>
</tr>
</tbody>
</table>
Using the Engine Tool

How to:
Create an Engine Statement

Reference:
ENGINE Dialog Box

The Engine tool exposes the FOCUS ENGINE SET commands and enables you to enter ENGINE commands or connection attributes, and override parameters.

Note: The Engine tool is only used to create ENGINE statements. You are responsible for having the knowledge of the ENGINE SET command or the Connection Attribute needed.

Procedure: How to Create an Engine Statement

1. From the Procedure Viewer, click and hold a component connector (yellow diamond), then click the Engine button on the Component Connector toolbox.
   
or
   Select Engine from the Insert menu.
   
The ENGINE dialog box opens.

2. Select an Engine from the Engine drop-down list or manually type in an Engine name.

3. Enter either the Connection string or the SET parameters.

4. Click the TEST button to check the ENGINE statement.

5. Click OK to save the ENGINE statement in the Procedure Viewer.
   
   You may click the ENGINE file to update the existing ENGINE statement.
Reference: ENGINE Dialog Box

The following image shows the Engine Dialog box.

![Engine Dialog Box]

The ENGINE dialog box has the following fields:

**Engine**

Contains a list of current adapters for which ENGINE statements exist. This field is editable. You may manually type a new engine name.

**Connection**

Contains any existing connections found in your environment.

**SET parameters**

Type in a parameter statement.

For more information about Adapters, see the *Adapter Administration for Windows, UNIX, OpenVMS, i5/OS, and z/OS* manual.

Managing Flow of Control

You can use Dialogue Manager to control the execution of procedures. Among other things, Dialogue Manager commands:

- Display forms.
- Set and test variables.
- Execute procedures.
- Execute operating system commands.
- Read and write disk files.

Dialogue Manager commands are stored in procedures. Their execution can be triggered by actions on a form, such as clicking a button. Procedures containing Dialogue Manager commands can be reused by other procedures in a project.
The ability of Dialogue Manager commands to display forms and the ability of a form's objects to call Dialogue Manager commands are two of the most important aspects of application development. The most powerful applications are a combination of graphical and command-driven components.

For related information, see Managing Flow of Control in an Application in the Developing Reporting Applications manual.

Working With a Full Procedure

How to:

- Access ReportCaster to Schedule a Report
- Encrypt or Decrypt a Procedure
- Edit a Procedure as Text

Although most of procedure development happens at the component level, there are several tasks that you can perform on a full procedure:

- **Activating OLAP.** If a procedure is set to run in HTML format and it uses an OLAP-enabled Master File, you can activate OLAP for that procedure and use the OLAP interface to manipulate data in a report or graph. For details on OLAP, see the Creating Reports With Graphical Tools manual.

- **Scheduling a report for distribution.** You can schedule a report for distribution to a single user or multiple users at specific times. For details on scheduling a report, see the Creating Reports With Graphical Tools manual.

  You can schedule an HTML-formatted report that has been OLAP-enabled, but the OLAP feature will not be active. See How to Access ReportCaster to Schedule a Report on page 167.

- **Create a Web-based user interface.** You can use the HTML Composer to create a launch page (Web-based interface) that makes a report available to browser users. For more information on your options, see Understanding User Interface Basics on page 17.

- **Securing a procedure's contents by encrypting or decrypting it.** You can restrict a user from viewing a procedure by encrypting it. To restore access to a procedure, decrypt it. See How to Encrypt or Decrypt a Procedure on page 167.

- **Editing a procedure.** You can edit a procedure as text. See How to Edit a Procedure as Text on page 168.

- **Debugging a Procedure.** Debugging enables you to check a procedure for any errors by viewing the logical execution of the request line-by-line in the Command Console.
**Procedure:** How to Access ReportCaster to Schedule a Report

1. Expand the *WebFOCUS Environments* node.
2. Select a server (for example, localhost).
3. Click the *ReportCaster* icon on the Main toolbar.
   
   The ReportCaster Signon dialog box opens.

   ![ReportCaster Signon Dialog](image)

4. Enter a valid user name and password.
5. Click OK. If the user name and password for ReportCaster are incorrect, you are prompted to log on again.

   The ReportCaster dialog box opens. For information about scheduling a report, see the *Creating Reports With Graphical Tools* manual.

**Procedure:** How to Encrypt or Decrypt a Procedure

To encrypt a procedure:

1. From an open project, right-click a procedure in the Procedures folder, then select *Properties*.
   
   The Project Properties dialog box opens.

2. Select the *Encrypted* check box, and click *Apply*.
3. From an open project, right-click a procedure in the Procedures folder, then select *Properties*.
   
   The Project Properties dialog box opens.

4. Deselect the *Encrypted* check box, and click *Apply*. 
**Procedure:** How to Edit a Procedure as Text

Right-click the procedure and select *Edit As Text* from the shortcut menu, or select the procedure and choose *Edit As Text* from the File Menu. The text editor opens, displaying the procedure's syntax.

For details on using the text editor, see *Editing Application Components as Text in Developer Studio* on page 269.

**Running a Procedure**

**How to:**

Run a Procedure on a Local Server

You can run a procedure on the local host at any point during the development process.

**Procedure:** How to Run a Procedure on a Local Server

1. Select the procedure.
2. Do one of the following:
   - Right-click the procedure, then choose *Run* from the shortcut menu.
   - Click the *Run* button on the Main toolbar.

**Canceling a Running Procedure**

You can cancel any procedure while it is running. This may be necessary if you run a procedure and find that it takes too long to process or that it is the wrong one. To cancel a procedure at any time, you must use the Web Console. For more information, see the *Server Administration for UNIX, Windows, OpenVMS, i5/OS, and z/OS* manual.
A WebFOCUS StyleSheet enables you to specify a hyperlink to a JavaScript function. You will apply this feature to perform data calculations in a drill-down report.

How you enable report manipulation is an important part of a user interface. This topic describes how to use the WebFOCUS Viewer in an application for navigation of a long report. You will also see how to use an external Cascading Style Sheet to add scrolls bars to report display.

You can implement advanced design features on a launch page. An example is a menu with customized options for an individual user.

This topic describes coding capabilities that extend the functionality and usability of an interface.

Report presentation is not limited to HTML. You will see how to display:

- A report in a format other than HTML, such as PDF (Portable Document Format) or PostScript, using a helper application.
- Multiple reports in different formats from a single launch page.

**Topics:**

- Displaying a Report in a Helper Application
- Controlling Multiple Reports
- Adding JavaScript for Drill-Down Reporting
- Facilitating Report Manipulation
- Using a Cascading Style Sheet to Standardize Display
- Displaying a Previously Run Report
- Passing a User ID From HTML for a Custom Menu
Displaying a Report in a Helper Application

How to:
Specify a MIME Type in Microsoft Windows Explorer
Display a Report in a Helper Application

Reference:
Supported MIME Types

A helper application or plug-in is a desktop program used by a browser to open a file with a format other than standard HTML. An example of a helper application is Adobe® Acrobat® Reader, which opens a PDF file.

When a Web server sends a special type of file to a browser, it includes MIME (Multipurpose Internet Mail Extensions) information, indicating the file format. WebFOCUS determines which helper application to run based on MIME type. You must associate the content (MIME) type of a file with a helper application using Windows Explorer.

This feature applies to WebFOCUS. You can create the procedure and launch page in Developer Studio, and deploy them with the Deploy Wizard. Run them in WebFOCUS.

Procedure: How to Specify a MIME Type in Microsoft Windows Explorer

1. Open Windows Explorer.
2. Select Folder Options from the Tools menu.
3. Select the File Types tab.
4. Select a file type in the Registered File Types list box, then click the New Type button. The Add New File Type dialog box opens.
5. Enter the following information.
   - In the Description of Type field, enter a description of your choice.
   - In the Associated Extension field, enter the extension for the file type.
   - In the Content Type (MIME) field, enter the MIME type for the file type and extension.
6. Click OK to save your changes.
**Reference: Supported MIME Types**

WebFOCUS supports the following MIME types.

<table>
<thead>
<tr>
<th>MIME Type</th>
<th>Application</th>
<th>File Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>application/postscript</td>
<td>GhostView</td>
<td>.PS</td>
</tr>
<tr>
<td>application/x-prn</td>
<td>Printer (using PCL printer driver)</td>
<td>.PRN</td>
</tr>
<tr>
<td>application/x-dif</td>
<td>Microsoft Excel or other spreadsheet</td>
<td>.DIF</td>
</tr>
<tr>
<td>application/x-doc</td>
<td>Microsoft Word</td>
<td>.DOC</td>
</tr>
<tr>
<td>application/x-xls</td>
<td>Microsoft Excel</td>
<td>.XLS, .E97, .WK1, .XHT</td>
</tr>
<tr>
<td>application/vnd.ms-excel</td>
<td>Microsoft Excel</td>
<td>.XLS, .E97, .WK1, .XHT</td>
</tr>
<tr>
<td>application/pdf</td>
<td>Adobe Acrobat Reader</td>
<td>.PDF</td>
</tr>
<tr>
<td>text/plain</td>
<td>Microsoft Notepad or other text editor</td>
<td>.WP or .HTS</td>
</tr>
<tr>
<td>text/html</td>
<td>Native browser format</td>
<td>.HTML</td>
</tr>
<tr>
<td>text/htm</td>
<td>Native browser format</td>
<td>.HTM</td>
</tr>
<tr>
<td>image/gif</td>
<td>Native browser format</td>
<td>.GIF</td>
</tr>
<tr>
<td>image/jpeg</td>
<td>Native browser format</td>
<td>.JPG, .JPEG</td>
</tr>
<tr>
<td>XML</td>
<td>Native browser format</td>
<td>.XML</td>
</tr>
</tbody>
</table>

**Procedure: How to Display a Report in a Helper Application**

1. Create a procedure that includes the command

   ```
   ON TABLE PCHOLD FORMAT fmt
   ```

   where:

   `fmt`

   Is the file format for the data.

   For details on formats and complete syntax, see the Creating Reports With WebFOCUS Language manual.

   **Caution:** Keep in mind that a valid format for browser display in a helper application requires a supported MIME type.
2. Create a launch page from which a user can run the report.

3. Deploy the procedure and launch page using the Deploy Wizard. See Step 2: Create a Deployment Scenario on page 241 for instructions. When you run the launch page, WebFOCUS returns the data to the browser in the format specified. The browser displays it using the appropriate helper application.

Example: Displaying a Report In Excel and Offering a Choice of Display Formats

The following are examples of displaying a report in an Excel spreadsheet and offering a choice of display formats.

Displaying a Report in an Excel 2000 Spreadsheet

The following is an example of displaying a report in an Excel 2000 spreadsheet.

1. Create a procedure named HELPER, which generates a report on product orders:

   ```
   TABLE FILE GGORDER
   SUM QUANTITY BY PCD
   WHERE PCD IS 'B141' OR 'B142' OR 'F101' OR 'F102'
   ON TABLE PCHOLD FORMAT EXL2K
   END
   ```

2. Create a launch page from which a user can run the report.

3. Deploy the procedure and launch page using the Deploy Wizard. See Step 2: Create a Deployment Scenario on page 241 for instructions.
4. Run the launch page, and click the link. WebFOCUS returns the data to the browser in EXL2K format. The browser calls the helper application, Microsoft Excel 2000, and displays the report:

![Excel Spreadsheet Image]

**Offering a Choice of Display Formats**

The following is an example of offering a choice of display formats.

1. Create a procedure named QASTATUS, which generates a quality assurance report on a plant (PLANT) of interest, in a format (FMT) of choice:

   ```sql
   TABLE FILE CENTQA
   ON TABLE SET PAGE-NUM OFF
   SUM CNT.PROBNUM AS 'Total Number, of Problems'
   SUM CNT.PROBNUM AS 'Problem by, Product' BY PLANT NOPRINT BY PRODNAME
   WHERE PLANT EQ '&PLANT'
   HEADING CENTER
   "QA Report for &PLANT"
   ON TABLE PCHOLD FORMAT &FMT
   END
   ```
Create a launch page from which a user can run the report. The following sample launch page is named FORMATS. The letter on the left corresponds to the note explaining the code.

```html
<HTML>
<HEAD>
<TITLE>Multiple Display Formats</TITLE>
</HEAD>
<BODY BGCOLOR="#E3E3E3">
<FONT FACE="Arial">
<CENTER>
<FONT SIZE="+1">Welcome to the QA Database</FONT>
</CENTER>
<FORM ACTION="/ibi_apps/WFServlet" METHOD="get">
<INPUT NAME="IBIF_ex" VALUE="qastatus" TYPE="hidden">
<P>
Select a plant to generate a report on product problems:
</P>
SELECT NAME="PLANT">
<OPTION>BOS
<OPTION>DAL
<OPTION>LA
<OPTION>ORL
<OPTION>SEA
<OPTION>STL
</SELECT>
<P>
<HR>
a.  
<TABLE CELLPADDING="2">
<TR>
<TD CLASS="LABEL">Select a display format:</TD>
<TD><INPUT TYPE="RADIO" NAME="FMT" VALUE="HTML" CHECKED>HTML</TD>
<TD><INPUT TYPE="RADIO" NAME="FMT" VALUE="PDF">PDF</TD>
<TD><INPUT TYPE="RADIO" NAME="FMT" VALUE="EXL2K">Excel 2000</TD>
</TR>
</TABLE>
<P>
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Submit">
<INPUT TYPE="RESET" NAME="RESET" VALUE="Reset">
</FORM>
</BODY>
</HTML>
```

a. Radio buttons in a table prompt the user for the format (FMT), and the value is passed to the procedure.
3. Deploy the procedure and launch page using the Deploy Wizard. See Step 2: Create a Deployment Scenario on page 241, for instructions.

4. Run the launch page.

![Image of a launch page showing a drop-down list and input fields for selecting a format]

5. Select ORL from the drop-down list. Select PDF for the format. Click Submit.
WebFOCUS returns the data to the browser in PDF. The browser calls the helper application, Adobe Acrobat Reader, and displays the report:

WebFOCUS Client variables give you control over the display of multiple reports on the same HTML page with multiple frames without coding HTML FRAME syntax. It also allows you to link multiple reports with a single Table of Contents (TOC) that calls them. For example, you can use these variables to display different report formats, such as HTML and PDF, on a single launch page.
Note:

- Table of Contents (TOC) functionality is not supported for ReportCaster distribution because the WebFOCUS Client is controlling the display of multiple reports.

- For current implementation of Table of Contents (TOC) functionality, see the Navigating Within an HTML Report chapter of the Creating Reports With WebFOCUS Language manual. In Developer Studio, see Adding a Table of Contents Page in the Creating Reports With the Document Composer chapter or Navigating Sort Groups From a Table of Contents in the Creating Reports with the Report Painter chapter of the Creating Reports With Graphical Tools manual.

You can:

- Pass the variables on a call to the WebFOCUS Client from a hyperlink.

- Set the variables in a procedure with the Dialogue Manager -TYPE command. If you use this method, test your application in WebFOCUS. Either create the procedure with a text editor in WebFOCUS, or create it in Developer Studio, deploy it, and then run it.

Use the variables described in this topic with procedures that generate more than one report. See Managing Flow of Control in an Application in the Developing Reporting Applications manual for details on the use of -RUN, -INCLUDE, and EX to code multiple report requests.

WebFOCUS Client variables apply to WebFOCUS. Developer Studio does not support multiple reports generated by these variables. Instead, multiple reports can be created with the Developer Studio HTML Composer. These reports must be run in WebFOCUS.

Reference: Formats of Multiple Reports

Two formats of multiple reports are supported:

- Multiple reports displayed in a frame set.

- Table of content reports where all reports in the procedure are listed in a menu on the left side, and the first report is displayed on the right. When the user clicks on a report in the menu, it is executed and appears by default on the right side of the window.

Reference: WebFOCUS Client Variables

The following are the variables that control multiple reports.

**IBIWF_mreports**

Controls the multiple report option and creates either a TOC or a frameset with one report for each frame in the browser.
The syntax is

```
IBIWF_mreports = {OFF | INDEX | FRAME}
IBIWF_mrcolumns = {1 | n}
IBIWF_mrrows = n
IBIWF_mprefix = {Report | text}
IBIWF_morder = {FORWARD | REVERSE}
IBIWF_mframename = {MREPORT | text}
IBIWF_index = {ON | OFF}
```

where:

**OFF**

Disables the multiple report option. OFF is the default value.

**INDEX**

Creates a TOC that lists all reports in a procedure. Default sequence numbers are from 1 (for the first report generated) to \( n \) (for the last report generated). Used with IBIWF_mprefix.

**FRAME**

Creates a default frameset. The number of reports in the frameset is determined by the IBIWF_mrcolumns variable.

**IBIWF_mrcolumns**

Is the number of side-by-side reports from left to right across the page when IBIWF_mreports is set to FRAME. If this variable is not set, reports are displayed top to bottom.

The syntax is

```
IBIWF_mrcolumns = {1 | n}
```

where:

\( n \)

Is the number of reports. The default value is 1. The maximum value is 9.

**IBIWF_mrrows**

Is the number of vertically stacked reports when IBIWF_mreports is set to FRAME.

The syntax is

```
IBIWF_mrrows = n
```

where:

\( n \)

Is the number of reports desired from top to bottom.
**IBIWF_mprefix**

Is descriptive text that precedes a sequence number and identifies a report on a TOC. WebFOCUS appends the number 1 (for the first report generated) to n (for the last report generated), as set by the index value on IBIWF_mreports.

Do not use this variable if IBIWF_mreports = FRAME.

The syntax is

\[ \text{IBIWF_mprefix} = \{\text{Report} | \text{text}\} \]

where:

- **text**
  
  Is a character string, up to 50 characters long. The maximum length does not include the number appended by WebFOCUS. The default value is Report.

**IBIWF_morder**

Is the order in which reports display in the browser. Applies only when IBIWF_mreports = FRAME. Ignored when IBIWF_mreports = INDEX.

The syntax is

\[ \text{IBIWF_morder} = \{\text{FORWARD} | \text{REVERSE}\} \]

where:

- **FORWARD**
  
  Displays reports in the order in which they were coded and executed. This value is the default.

- **REVERSE**
  
  Displays reports in the reverse order in which they were coded and executed. This is especially useful if the last report is a summary report you would like to display on the Web page first.

**IBIWF_mframename**

Is a name for a frame when IBIWF_mreports = FRAME. If you do not code this variable, WebFOCUS internally names the frames MREPORT1 through MREPORTn, which may conflict with other HTML code.

The syntax is

\[ \text{IBIWF_mframename} = \{\text{MREPORT} | \text{text}\} \]

where:

- **text**
  
  Is a character string, up to 20 characters long.
**IBIWF_index**

Controls whether a sequence number (1, 2,...,n) is appended to the end of the names on the TOC when IBIWF_mreports = INDEX.

The syntax is

\[
\text{IBIWF_index} = \{\text{ON} | \text{OFF}\}
\]

where:

- **ON**
  - Appends a sequence number of 1 (for the first report generated) to n (for the last reported generated). ON is the default value.

- **OFF**
  - Omits a sequence number. Only the text specified by IBIWF_mprefix applies.

**Syntax:**

How to Control Multiple Reports From a Hyperlink

\[
<A \text{ HREF="/alias/ibiweb.exe?IBIF_ex=procedure[&var=value [&var=value]...]> text</A> \]

\[
<A \text{ HREF="/alias/webapi.dll?IBIF_ex=procedure[&var=value [&var=value]...]> text</A> \]

\[
<A \text{ HREF="http://server_name/servlet?WFServlet=procedure[&var=value [&var=value]...]> text</A> \]

where:

- **alias**
  - Points to the directory in which the WebFOCUS CGI (ibiweb.exe) or ISAPI (webapi.dll) is located. A Web server uses an alias to provide a logical name for a physical directory. The default alias is cgi-bin/ibi_cgi. It is set during WebFOCUS installation and configuration.

  To call WebFOCUS on another Web server, specify a fully qualified URL, for example:

  \[
  \text{http://web_server/alias/ibiweb.exe...} \]

- **server_name**
  - Is the name of the Web server on which WebFOCUS is installed.

- **procedure**
  - Is the name of the procedure to run.

- **var=value**
  - Is a WebFOCUS Client variable and its corresponding value.
You can pass more than one variable-value pair, but do not include a space between pairs. Use an ampersand (&) as a delimiter to separate each variable-value pair. A value can be a maximum of 80 characters long.

If a value contains an embedded blank, substitute a plus sign (+) or the characters %20 for the blank.

See *WebFOCUS Client Variables* on page 177 for a list of variables and valid values.

**Example: Displaying Two Reports With an Index Value**

The following is an example of displaying two reports with an index value.

1. Create a procedure named TWORPTS, which consists of two requests. The first generates a report on total dollar sales; the second, on total unit and dollar sales.

   **Procedure:**

   ```
   TABLE FILE GGSALES
   SUM DOLLARS BY PRODUCT
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF, $
   ENDSTYLE
   END
   -RUN
   
   TABLE FILE GGSALES
   SUM UNITS DOLLARS BY PRODUCT
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF, $
   ENDSTYLE
   END
   -RUN
   ```

2. Create a launch page named TWOLNCH. It contains a hyperlink that calls the WebFOCUS Servlet and passes it the name of the procedure to run. It also sets the variables that provide an identifier (Sales Analysis Report) and sequence number (1 and 2) to each report.
Launch Page:

```html
<HTML>
<BODY>
<A HREF="/ibi_apps/WFServlet?IBIF_ex=tworpts
&IBIWF_mreports=index&IBIWF_mprefix=Sales+Analysis+Report">
Run report.</A>
</BODY>
</HTML>
```

3. Deploy the procedure and launch page using the Deploy Wizard. See *Step 2: Create a Deployment Scenario* on page 241, for instructions.

4. Run the launch page in the browser, and click *Run report* to receive the report on total dollar sales:

```
<table>
<thead>
<tr>
<th>Product</th>
<th>Dollar Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biscotti</td>
<td>5263317</td>
</tr>
<tr>
<td>Capuccino</td>
<td>2381590</td>
</tr>
<tr>
<td>Coffee Grinder</td>
<td>2337567</td>
</tr>
<tr>
<td>Coffee Pot</td>
<td>2449585</td>
</tr>
<tr>
<td>Croissant</td>
<td>7749902</td>
</tr>
<tr>
<td>Espresso</td>
<td>3906243</td>
</tr>
<tr>
<td>Latte</td>
<td>10943622</td>
</tr>
<tr>
<td>Mug</td>
<td>4522521</td>
</tr>
<tr>
<td>Scone</td>
<td>4216114</td>
</tr>
<tr>
<td>Thermos</td>
<td>2385829</td>
</tr>
</tbody>
</table>
```
Click Sales Analysis Report 2 for the report on total unit and dollar sales:

<table>
<thead>
<tr>
<th>Product</th>
<th>Unit Sales</th>
<th>Dollar Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biscotti</td>
<td>421377</td>
<td>5263317</td>
</tr>
<tr>
<td>Capuccino</td>
<td>189217</td>
<td>2381590</td>
</tr>
<tr>
<td>Coffee Grinder</td>
<td>186534</td>
<td>2337567</td>
</tr>
<tr>
<td>Coffee Pot</td>
<td>190695</td>
<td>2449585</td>
</tr>
<tr>
<td>Croissant</td>
<td>630054</td>
<td>7749902</td>
</tr>
<tr>
<td>Espresso</td>
<td>308986</td>
<td>3906243</td>
</tr>
<tr>
<td>Latte</td>
<td>878063</td>
<td>10943622</td>
</tr>
<tr>
<td>Mug</td>
<td>360570</td>
<td>4522521</td>
</tr>
<tr>
<td>Scone</td>
<td>333414</td>
<td>4216114</td>
</tr>
<tr>
<td>Thermos</td>
<td>190081</td>
<td>2385829</td>
</tr>
</tbody>
</table>

**Syntax:**

**How to Control Multiple Reports From a Procedure**

- **TYPE** WEBFOCUS CGIVAR var=value
  .
  .
  -RUN

where:

**var=value**

Is a WebFOCUS Client variable and its corresponding value. You can include only one variable-value pair per each -TYPE command.

See *WebFOCUS Client Variables* on page 177 for a list of variables and valid values.

**Note:** Include the command -RUN at the end of each request to execute the previous set of -TYPE commands.

**Example:**

**Displaying Side-By-Side Reports**

The following is an example of displaying side-by-side reports.

1. Create a procedure named SIDERPTS, which consists of two requests. The first generates a report on sales by store; the second, on sales by product.
   - -TYPE commands create a two-frame page on which the reports display side-by-side.
**Procedure:**

- `TYPE WEBFOCUS CGIVAR IBIWF_mreports=FRAME`
- `TYPE WEBFOCUS CGIVAR IBIWF_mrcolumns=2`
- `TABLE FILE CENTORD`
- `HEADING "Sales By Store"
  SUM LINEPRICE AS 'Sales'
  BY SNAME
  ON TABLE SET PAGE-NUM OFF`
- `ON TABLE SET STYLE *
  TYPE=REPORT, GRID=OFF,$`
- `ENDSTYLE
  END
- `RUN`
- `TABLE FILE CENTORD`
- `HEADING "Sales By Product"
  "$ "
  SUM LINEPRICE AS 'Sales'
  BY PRODCAT AS 'Product'
  ON TABLE SET PAGE-NUM OFF`
- `ON TABLE SET STYLE *
  TYPE=REPORT, GRID=OFF,$`
- `ENDSTYLE
  END`

2. Create a launch page that runs the procedure.

3. Deploy the procedure and launch page using the Deploy Wizard. See Step 2: Create a Deployment Scenario on page 241 for instructions.
4. Access the launch page in the browser, and run the reports:

**Example:** **Displaying Two Reports With Descriptive Names and Sequence Numbers**

The following is an example of displaying two reports with descriptive names and sequence numbers.

1. **Create a procedure named HTMPDF1, which consists of two requests.** The first generates a report on total dollar sales in HTML format; the second, on total dollar sales in PDF format.

   - **TYPE commands set the variables that provide an identifier (HTML Report and PDF Report) and sequence number (1 and 2) to each report.** The command that creates the TOC is required only once at the beginning of the procedure.
Controlling Multiple Reports

Procedure:

- TYPE WEBFOCUS CGIVAR IBIWF_mreports=INDEX
- TYPE WEBFOCUS CGIVAR IBIWF_mprefix=HTML Report
  TABLE FILE GGSALES
  SUM DOLLARS BY PRODUCT
  ON TABLE SET PAGE-NUM OFF
  ON TABLE SET STYLE *
  TYPE=REPORT, GRID=OFF,$
  ENDSTYLE
  END
  -RUN

  TABLE FILE GGSALES
  -TYPE WEBFOCUS CGIVAR IBIWF_mprefix=PDF Report
  SUM DOLLARS BY PRODUCT
  ON TABLE PCHOLD FORMAT PDF
  END

2. Create a launch page that runs the procedure.

3. Deploy the procedure and launch page using the Deploy Wizard. See *Step 2: Create a Deployment Scenario* on page 241 for instructions.
4. Run the launch page in the browser. The report in HTML format displays.

<table>
<thead>
<tr>
<th>HTML Report 1</th>
<th>Product</th>
<th>Dollar Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bascom</td>
<td>5263317</td>
<td></td>
</tr>
<tr>
<td>Capuccino</td>
<td>2381590</td>
<td></td>
</tr>
<tr>
<td>Coffee Grinder</td>
<td>2337567</td>
<td></td>
</tr>
<tr>
<td>Coffee Pot</td>
<td>2449585</td>
<td></td>
</tr>
<tr>
<td>Croissant</td>
<td>7749902</td>
<td></td>
</tr>
<tr>
<td>Espresso</td>
<td>3906243</td>
<td></td>
</tr>
<tr>
<td>Latte</td>
<td>10943622</td>
<td></td>
</tr>
<tr>
<td>Mug</td>
<td>4522521</td>
<td></td>
</tr>
<tr>
<td>Soothe</td>
<td>4216114</td>
<td></td>
</tr>
<tr>
<td>Thermos</td>
<td>2365829</td>
<td></td>
</tr>
</tbody>
</table>
5. Click PDF Report 2 for the report in PDF using the Acrobat Reader.

Example: Displaying Two Reports With Descriptive Names Only

The following is an example of displaying two reports with descriptive names only.

1. Create a procedure named HTMPDF2, which consists of two requests. The command
   -TYPE WEBFOCUS CGIVAR IBIWF_index=OFF omits sequence numbers from the names
   in the TOC.

Procedure:

   -TYPE WEBFOCUS CGIVAR IBIWF_mreports=INDEX
   -TYPE WEBFOCUS CGIVAR IBIWF_index=OFF
   -TYPE WEBFOCUS CGIVAR IBIWF_mprefix=HTML Report
   TABLE FILE GGSALES
   SUM DOLLARS BY PRODUCT
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF,$
   ENDSTYLE
   END
   -RUN
2. Create a launch page that runs the procedure.

3. Deploy the procedure and launch page using the Deploy Wizard. See Step 2: Create a Deployment Scenario on page 241 for instructions.

4. Run the launch page in the browser. The reports have descriptive names without sequence numbers.
**Example: Displaying Two Reports With Sequence Numbers Only**

The following is an example of displaying two reports with sequence numbers only.

1. Create a procedure named HTMPDF3, which consists of two requests. The command
   `-TYPE WEBFOCUS CGIVAR IBIWF_mprefix=` omits descriptive names for the reports.

   ```
   -TYPE WEBFOCUS CGIVAR IBIWF_mreports=INDEX
   -TYPE WEBFOCUS CGIVAR IBIWF_mprefix=
   TABLE FILE GGSALES
   SUM DOLLARS BY PRODUCT
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF,$
   ENDDOCUMENT
   END
   -RUN
   TABLE FILE GGSALES
   SUM DOLLARS BY PRODUCT
   ON TABLE PCHOLD FORMAT PDF
   END
   ```

2. Create a launch page that runs the procedure.

3. Deploy the procedure and launch page using the Deploy Wizard. See *Step 2: Create a Deployment Scenario* on page 241 for instructions.
4. Run the launch page in the browser. The reports have sequence numbers without descriptive names:

![Add image]

Adding JavaScript for Drill-Down Reporting

**How to:**
Specify a Hyperlink to a JavaScript Function

This topic illustrates the use of JavaScript to create a drill-down report. It describes how to call a JavaScript function and pass values to it from the summary component of the report, to dynamically determine the content of the detailed component.

You will see how to specify a hyperlink to a JavaScript function in a procedure’s StyleSheet. Once a hyperlink is defined, a user can select the associated object in the report to execute the function.

For more information on StyleSheets, see the Creating Reports With WebFOCUS Language manual.

For details on JavaScript capabilities and syntax, see your JavaScript documentation.

**Syntax:**

```
TYPE=type, (COLUMN|ACROSSCOLUMN)=fieldname, JAVASCRIPT=func[(value)],$``
where:

**type**

Is the report component that serves as the hyperlink. The TYPE attribute and its value must be first in the StyleSheet.

For example, use TYPE=DATA to set up a hyperlink from a data field in a report, or use TYPE=REPORT to set up a hyperlink from any component in a report.

**fieldname**

Is the name of the field in the data source whose value, when selected, executes the hyperlink.

**func**

Is the name of the JavaScript function.

The maximum length of the code for JAVASCRIPT=func, including any passed values, is 800 characters. The code can span more than one line. If you split it across a line, use a backslash at the end of the first line as the continuation character. If you split a line at a point at which a space is required, the space must be before the backslash, or must be the first character on the next line.

In this example,

```javascript
JAVASCRIPT=myfunc(COUNTRY\ CAR MODEL),$
```

the code correctly spans two lines.

**value**

Is a value or values passed to the JavaScript function. Specify a value in one of the following ways:

- As the name of a report column.

- As a constant. You must enclose the value in single quotation marks.

- As a Dialogue Manager amper variable. You can only specify an amper variable in a StyleSheet that is part of the procedure (inline).

An amper variable is typically used to pass a constant value, in which case it must be enclosed in single quotation marks, for example, '&ABC'.

*Adding JavaScript for Drill-Down Reporting*
**Example:** Creating a Drill-Down Report With JavaScript

In this example, the summary component of a drill-down report displays orders per month for each store code. When the user selects a particular store code, a hyperlink calls a JavaScript function that performs calculations on the data and displays detailed information for the selected store.

1. Create a procedure named JDRILL. The letters on the left correspond to the notes explaining the code.

   ```javascript
   SET MESSAGE = OFF
   TABLE FILE GGORDER
   SUM QUANTITY BY STORE_CODE ACROSS ORDER_DATE
   IF ORDER_DATE GT 12/31/96
   a. ON TABLE HOLD
   END
   -RUN
   TABLE FILE HOLD
   HEADING CENTER
   "Store Sales Analysis Using JavaScript"
   b. PRINT E01 AS 'Store,Code' E02 AS 'Jan' E03 AS 'Feb' E04 AS 'Mar'
      E05 AS 'Apr' E06 AS 'May' E07 AS 'Jun'
      E08 AS 'Jul' E09 AS 'Aug' E10 AS 'Sep'
      E11 AS 'Oct' E12 AS 'Nov' E13 AS 'Dec'
   FOOTING CENTER
   "Please click on the store code to summarize the store's data."
   c. ON TABLE HOLD AS JAVATEMP FORMAT HTMTABLE
   d. ON TABLE SET STYLE *
      TYPE=DATA,COLUMN=STORE_CODE,$
      JAVASCRIPT=conprint(E01 E02 E03 E04 E05 E06 E07 E08 E09 E10 E11 \ E12 E13),$ COLOR=GREEN,STYLE=ITALIC,$
      TYPE=TITLE, COLOR=RED,STYLE=BOLD,$
      TYPE=HEADING, COLOR=BLUE,STYLE=ITALIC,SIZE=11,$
      TYPE=FOOTING, COLOR=BLUE,STYLE=ITALIC,$
      END STYLE
   END
   e. -HTMLFORM BEGIN
      <HTML>
      "Developing Reporting Applications With Graphical Tools"
<SCRIPT LANGUAGE="JavaScript"> 
var spacer = "................................."; var pos=0;
var aaaa;
function conprint(aaax,lsyr,aa1,aa2,aa3,aa4,aa5,aa6,aa7,
    aa8,aa9,aa10,aa11,aa12) 
{
    var spacer="   ";

    lsyrave=parseFloat(lsyr);
a1=parseFloat(aa1);
a2=parseFloat(aa2);
a3=parseFloat(aa3);
a4=parseFloat(aa4);
a5=parseFloat(aa5);
a6=parseFloat(aa6);
a7=parseFloat(aa7);
a8=parseFloat(aa8);
a9=parseFloat(aa9);
a10=parseFloat(aa10);
a11=parseFloat(aa11);
a12=parseFloat(aa12);
gotota=eval(a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 +
    a11 + a12);
goavea=gotota/12;
gotot=Math.round(gotota);
goave=Math.round(goavea);
diffavea=goavea-lsyrave;
diffave=Math.round(diffavea);

    document.form1.text1.value=gotot;
document.form1.text2.value=goave;
document.form1.store.value=aaax;
arraygo = new Array(13);
arraygo[1]=a1;
arraygo[2]=a2;
arraygo[3]=a3;
arraygo[4]=a4;
arraygo[5]=a5;
arraygo[6]=a6;
arraygo[7]=a7;
arraygo[8]=a8;
arraygo[9]=a9;
arraygo[10]=a10;
arraygo[12]=a12;
minval=100000;
maxval=0;
mnmmax=0;
mnmmin=0;
</SCRIPT>
for(i = 1; i <= 12; i++)
{
  if(arraygo[i] > maxval)
  {
    mnmmax=i;
    maxval = arraygo[i];
  }
  if(arraygo[i] < minval)
  {
    mnmmin=i;
    minval = arraygo[i];
  }
}
rng1=maxval - minval;
rng=Math.round(rng1);
mnms=new Array(13);
mnms[1]="January";
mnms[2]="February";
mnms[3]="March";
mnms[4]="April";
mnms[5]="May";
mnms[6]="June";
mnms[7]="July";
mnms[8]="August";
mnms[9]="September";
mnms[10]="October";
mnms[12]="December";
document.form1.themax.value=maxval;
document.form1.themin.value=minval;
document.form1.range.value=rng;
document.form1.mnmmax.value=mnms[mnmmax];
document.form1.mnmmin.value=mnms[mnmmin]; }
a. This command saves the report output with 1997 data to a temporary file named HOLD in native machine format. Since this is the only data necessary for the report, this server extract file is created to speed subsequent processing.

b. This code formats the report, providing descriptive column titles. The store code is the first column. The following columns contain total monthly quantity for each store.

c. This command saves the report output to a temporary file in HTML format. The file is named JAVATEMP. It will be merged with the HTML page created later (see item e).

d. The StyleSheet specifies a hyperlink to a JavaScript function named conprint. The code passes the store code and monthly values to the function.

e. The Dialogue Manager command -HTMLFORM BEGIN indicates the start of an HTML page in which the JavaScript function is defined. The report output will be embedded on this page.

f. The HTML code declares the JavaScript function and passes values to it.

g. JavaScript assigns variable names to values displayed on the HTML page.

h. WebFOCUS reads the HTML comment and replaces it with the report output held in JAVATEMP.

i. The Dialogue Manager command -HTMLFORM END indicates the end of the HTML page.

2. Create a launch page from which a user can run the report.
3. Run the launch page, and click the link. The summary component displays.

![Store Sales Analysis Using JavaScript](image)

<table>
<thead>
<tr>
<th>Store Code</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1019</td>
<td>3961</td>
<td>3119</td>
<td>3951</td>
<td>3982</td>
<td>4057</td>
<td>2888</td>
<td>4996</td>
<td>3943</td>
<td>4526</td>
<td>3962</td>
<td>3317</td>
<td>5305</td>
</tr>
<tr>
<td>R1029</td>
<td>4152</td>
<td>3944</td>
<td>3927</td>
<td>3866</td>
<td>3697</td>
<td>3454</td>
<td>4449</td>
<td>4785</td>
<td>4461</td>
<td>4588</td>
<td>4454</td>
<td>4139</td>
</tr>
<tr>
<td>R1040</td>
<td>4189</td>
<td>3766</td>
<td>4342</td>
<td>3741</td>
<td>4681</td>
<td>3953</td>
<td>3775</td>
<td>4655</td>
<td>3731</td>
<td>4127</td>
<td>3021</td>
<td>3794</td>
</tr>
<tr>
<td>R1041</td>
<td>4650</td>
<td>4818</td>
<td>5061</td>
<td>3000</td>
<td>4528</td>
<td>4751</td>
<td>5396</td>
<td>4918</td>
<td>3442</td>
<td>4995</td>
<td>3873</td>
<td>5611</td>
</tr>
<tr>
<td>R1044</td>
<td>3836</td>
<td>5590</td>
<td>3068</td>
<td>4356</td>
<td>4126</td>
<td>4414</td>
<td>4412</td>
<td>5038</td>
<td>4044</td>
<td>4109</td>
<td>4010</td>
<td>3498</td>
</tr>
<tr>
<td>R1085</td>
<td>4039</td>
<td>5224</td>
<td>2700</td>
<td>3586</td>
<td>3435</td>
<td>3947</td>
<td>4083</td>
<td>3311</td>
<td>4886</td>
<td>3710</td>
<td>4382</td>
<td>4256</td>
</tr>
<tr>
<td>R1100</td>
<td>4190</td>
<td>3652</td>
<td>4244</td>
<td>3146</td>
<td>4068</td>
<td>3848</td>
<td>4348</td>
<td>3760</td>
<td>4312</td>
<td>4151</td>
<td>3987</td>
<td>3963</td>
</tr>
<tr>
<td>R1109</td>
<td>4216</td>
<td>4031</td>
<td>3350</td>
<td>3894</td>
<td>4336</td>
<td>4225</td>
<td>4221</td>
<td>4661</td>
<td>4421</td>
<td>3447</td>
<td>4161</td>
<td>4175</td>
</tr>
<tr>
<td>R1206</td>
<td>3842</td>
<td>4604</td>
<td>3550</td>
<td>3657</td>
<td>4237</td>
<td>4688</td>
<td>4493</td>
<td>4663</td>
<td>3902</td>
<td>2995</td>
<td>3701</td>
<td>4305</td>
</tr>
<tr>
<td>R1244</td>
<td>3169</td>
<td>3903</td>
<td>4014</td>
<td>4027</td>
<td>4049</td>
<td>4028</td>
<td>4111</td>
<td>4299</td>
<td>4061</td>
<td>4913</td>
<td>4730</td>
<td>3966</td>
</tr>
<tr>
<td>R1248</td>
<td>3961</td>
<td>3283</td>
<td>4502</td>
<td>3854</td>
<td>3038</td>
<td>5224</td>
<td>4411</td>
<td>3658</td>
<td>3596</td>
<td>4586</td>
<td>4374</td>
<td>4398</td>
</tr>
<tr>
<td>R1250</td>
<td>3195</td>
<td>4102</td>
<td>4492</td>
<td>4270</td>
<td>3570</td>
<td>4466</td>
<td>3781</td>
<td>3488</td>
<td>4233</td>
<td>4309</td>
<td>3759</td>
<td>3946</td>
</tr>
</tbody>
</table>

*Please click on the store code to summarize the store’s data.*

<table>
<thead>
<tr>
<th>Units sold for store</th>
<th>Monthly average of</th>
</tr>
</thead>
</table>

Top selling month is [ ] with [ ] units
Slowest month was [ ] with [ ] units
Range between best and slowest months [ ] units

4. Click store code R1019 for the detail component.

<table>
<thead>
<tr>
<th>Units sold for store</th>
<th>Monthly average of</th>
</tr>
</thead>
</table>

Top selling month is [ ] with [ ] units
Slowest month was [ ] with [ ] units
Range between best and slowest months [ ] units
Facilitating Report Manipulation

In this section:

- Using the WebFOCUS Viewer
- Opening and Closing the WebFOCUS Viewer
- Printing With On-Demand Paging
- WebFOCUS Viewer Navigation

Normally, a Web server returns an entire report to a browser. The browser waits until it receives all of the report before displaying it.

On-demand paging allows you to download one page of a report to a browser instead of the entire report. The Web server holds the remaining pages until the user requests them. This feature shortens the time the user waits to see the first page of a report. It is especially effective for long reports.

On-demand paging is implemented in the WebFOCUS Viewer. It requires that report output be formatted as HTML, which is the default setting for a request submitted through the WebFOCUS Client.
The following is a report displayed in the WebFOCUS Viewer.

![WebFOCUS Viewer](image)

**Using the WebFOCUS Viewer**

**How to:**

Enable the WebFOCUS Viewer

The WebFOCUS Viewer is a browser window divided into two frames:

- The Report Frame is the larger upper frame that contains one page of output.
- The Control Frame contains the controls used to navigate the report and to search for a string in the report. The navigational controls allow you to display the next or previous page, the first or last page, or a specific page.
**Syntax:**

How to Enable the WebFOCUS Viewer

```plaintext
SET WEBVIEWER = {OFF | ON}
ON TABLE SET WEBVIEWER {OFF | ON}
```

or

```plaintext
ON TABLE SET WEBVIEWER {OFF | ON}
```

where:

**OFF**

Disables on-demand paging; WebFOCUS downloads the entire report to a standard browser window. OFF is the default value.

**ON**

Enables on-demand paging. WebFOCUS downloads the first page of a report to the browser in the WebFOCUS Viewer. The number of lines displayed at one time depends on Windows desktop settings (resolution).

**Example: Enabling the WebFOCUS Viewer**

The procedure and launch page for this example are run in WebFOCUS. They must be tested and run in this environment.

1. Create a procedure named ONDEMAND, which displays an order report for a store in the WebFOCUS Viewer.

   **Procedure:**

   ```plaintext
   SET WEBVIEWER=ON
   TABLE FILE CENTORD
   PRINT ORDER_NUM ORDER_DATE
   BY STORE_CODE
   WHERE STORE_CODE EQ '1003DC'
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF,$
   ENDSPECIAL
   END
   ```

2. Create a launch page from which a user can run the report.
3. Run the launch page, and click the link. The report displays in the WebFOCUS Viewer.

Opening and Closing the WebFOCUS Viewer

**How to:**
Open the WebFOCUS Viewer in a Target Frame
Display a Home Page When You Close the WebFOCUS Viewer

**Reference:**
Closing the WebFOCUS Viewer

You can specify a target frame in which to open the WebFOCUS Viewer, and a home page that displays when you close the WebFOCUS Viewer.
### Syntax: How to Open the WebFOCUS Viewer in a Target Frame

SET WEBVIEWTARG = \{target\_frame|OFF\}

where:

`target_frame`
- Is the name of an existing frame in the browser or one of the following reserved HTML target frames:
  - `_blank` opens the WebFOCUS Viewer in a new browser window. This is the default.
  - `_self` opens the WebFOCUS Viewer in the same frame as the anchor.
  - `_parent` opens the WebFOCUS Viewer in the immediate parent frame that contains the anchor.
  - `_top` opens the WebFOCUS Viewer in the current browser window.
- `OFF` Opens the WebFOCUS Viewer in the frame from which you ran the report.

### Syntax: How to Display a Home Page When You Close the WebFOCUS Viewer

SET WEBVIEWHOME = \{home\_URL|OFF\}

where:

`home_URL`
- Is a valid URL that displays an HTML page when you close the WebFOCUS Viewer.
- `OFF` Displays a blank browser window when you close the WebFOCUS Viewer. You must enter another URL to run another report. OFF is the default value.

### Reference: Closing the WebFOCUS Viewer

The Close button, located on the Control Frame, closes the WebFOCUS Viewer and removes the report from the Web server. The page the browser displays next depends on the WEBVIEWTARG and the WEBVIEWHOME settings, as follows:

- If you set WEBVIEWTARG to `_blank`, the window that contained the WebFOCUS Viewer is removed. The browser does not display any page in any frame, and the WEBVIEWHOME setting has no effect.
- If you set WEBVIEWTARG to any other value, the result of clicking Close depends on the WEBVIEWHOME setting:
  - If you set WEBVIEWHOME to a URL, the browser displays the page associated with the URL in the frame that the WebFOCUS Viewer occupied.
If you set WEBVIEWHOME to OFF, the browser displays a blank page.

**Printing With On-Demand Paging**

**How to:**
Clear the Cache in Microsoft Internet Explorer

You must clear the browser cache before printing a report locally using the Print button on the browser toolbar. You must also activate the window by clicking it before using the Print button.

Use the procedure that applies to your browser to clear the cache.

**Procedure:** How to Clear the Cache in Microsoft Internet Explorer

1. Select *Internet Options* from the Tools menu.
2. On the General tab, for Temporary Internet Files, select *Delete Files*.

**WebFOCUS Viewer Navigation**

**How to:**
Enable the Goto Last Page Button

To enable the Goto Last Page button in the WebFOCUS Viewer with certain versions of Microsoft Internet Explorer (for example, Version 4.01 with Service Pack 2), complete the following procedure.

**Procedure:** How to Enable the Goto Last Page Button

1. From Internet Explorer, click the *Tools* menu and select *Internet Options*.
2. The Internet Options dialog box opens.
3. From the Temporary Internet files box, click *Settings*.
   The Settings dialog box opens.
4. Click the *Every visit to the page* radio button.
5. Click *OK* to apply the change and exit the Settings dialog box.
6. Click *OK* to exit the Internet Options dialog box.
Using a Cascading Style Sheet to Standardize Display

A Cascading Style Sheet is an extension to HTML that allows you to specify formatting for an HTML page. A Cascading Style Sheet can reside either in the HTML page that it formats, or in a separate file (with the extension .CSS), which can be shared by multiple pages. When it is in a separate file, it is known as an external Cascading Style Sheet.

This topic illustrates the use of an external Cascading Style Sheet to add scroll bars to reports when necessary and set standard fonts, font sizes, colors, and other display characteristics.

For details on Cascading Style Sheets, see the Creating Reports With WebFOCUS Language manual.

**Example:** Adding Scroll Bars to a Report

The following is an example of adding scroll bars to a report.

1. Create a procedure named SCROLL, which consists of two report requests and two graph requests. Each request uses a WebFOCUS StyleSheet to add individual styling features. The WebFOCUS StyleSheets for the reports call an external Cascading Style Sheet to add standardized application styling. See the Creating Reports With WebFOCUS Language manual for details on WebFOCUS StyleSheets and graph formatting options.

The letters on the left correspond to the notes explaining the code.

**Procedure: SCROLL**

```
SET PAGE-NUM=OFF

TABLE FILE CENTORD
HEADING
"Sales By Store"
SUM LINEPRICE AS 'Sales'
BY SNAME
a. ON TABLE SET STYLE *
b. TYPE=HEADING, CLASS=HEAD, $
```
5. Enhancing a User Interface

b. TYPE=TITLE, CLASS=TITLE, $
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=SNAME, COLOR=RED, STYLE=BOLD,$
WHEN=LINEPRICE LT 10000000,$
TYPE=DATA, COLUMN=LINEPRICE, COLOR=RED, STYLE=BOLD,$
WHEN=LINEPRICE LT 10000000,$
ENDSTYLE
ON TABLE HOLD AS CREPORT1 FORMAT HTMTABLE
END

GRAPH FILE CENTORD
SUM LINEPRICE
ACROSS SNAME

a. ON GRAPH SET STYLE *
UNITS=IN, LEFTMARGIN=0.250000,
     RIGHTMARGIN=0.250000, TOPMARGIN=0.250000,
     BOTTOMMARGIN=0.250000, SQUEEZE=ON, ORIENTATION=PORTRAIT,$
DEFMACRO=COND0001, MACTYPE=RULE, WHEN=N1 LE 100000.00,$
TYPE=REPORT, FONT='VERDANA', SIZE=10, BACKCOLOR=NONE,$
     STYLE=NORMAL, $
TYPE=DATA, ACROSSCOLUMN=N1, COLOR=RGB(144 24 24), $
TYPE=DATA, ACROSSCOLUMN=N1, COLOR=YELLOW, MACRO=COND0001,$
ENDSTYLE
ON GRAPH SET LOOKGRAPH 3D_BAR
ON GRAPH SET GRAPHEDIT OFF
ON GRAPH SET GRAPHSTYLE *
setConnectLineMarkers(true);
setO1LabelDisplay(true);
setO1AxisSide(0);
setO1MajorGridDisplay(false);
setO1MinorGridDisplay(false);
setY1LabelDisplay(true);
setY1AxisSide(0);
setY1MajorGridDisplay(false);
setY1MinorGridDisplay(false);
setPieFeelerTextDisplay(1);
setPieLabelDisplay(0);
setTextFormatPreset(getPieSliceLabel(),1);
setLegendDisplay(true);
setFootnoteString("Store Sales");
setTextJustHoriz(getFootnote(),1);
setFontStyle(getFootnote(),2);
ENDSTYLE
ON GRAPH SET BARNUMB OFF
ON GRAPH SET 3D ON
ON GRAPH SET GRID ON
ON GRAPH SET VAXIS 200
ON GRAPH SET HAXIS 300
ON GRAPH HOLD AS CGRAPH1 FORMAT HTMTABLE
END

SET LOOKGRAPH=PIE
SET GRAPHEDIT=OFF
SET GRID=ON
SET BARNUM=ON
SET 3D=ON
SET VAXIS=250
SET HAXIS=250
GRAPH FILE CENTORD
SUM LINEPRICE
BY PRODCAT

a. ON GRAPH SET GRAPHSTYLE *
setLegendDisplay(true);
setTitleDisplay(true);
setTextRotation(getO1Label(),0);
setY1LabelFormat(10);
setO1LabelAutofit(false);
setO1LabelStagger(false);
setTextRotation(getO1Label(),0);
setFontSizeVC(getO1Label(),1500);
setY1LabelAutofit(false);
setFontSizeVC(getY1Label(),1800);
setTextWrap(getLegendText(0),true);
setRect(getLegendArea(), new Rectangle(9000, -8000,8000, 15000));
setAutofit(getLegendText(0),false);
setFontSizeVC(getLegendText(0),850);
setGroupLabel(0,"Sales By Product");
setPieTilt(45);
ENDSTYLE
ON GRAPH HOLD AS CGRAPH2 FORMAT HTMTABLE
END

TABLE FILE CENTORD
HEADING
"Sales By Product"
SUM LINEPRICE AS 'Sales'
BY PRODCAT AS 'Product'

a. ON TABLE SET STYLE *
b. TYPE=HEADING, CLASS=HEAD, $
   TYPE=TITLE, CLASS=TITLE, $
   TYPE=REPORT, GRID=OFF, $
ENDSTYLE
ON TABLE HOLD AS CREPORT2 FORMAT HTMTABLE
END

c. -HTMLFORM scrollpg
a. These commands indicate the start of a WebFOCUS StyleSheet.

b. The CLASS attribute in the WebFOCUS StyleSheet refers to a set of styling characteristics in an external Cascading Style Sheet (SCROLLSS.CSS, which you create in step 3).

c. This command calls an HTML display page named SCROLLPG, which you create in step 2. It will incorporate the report output.

2. Create an HTML display page named SCROLLPG.HTM. The WebFOCUS Reporting Server must be able to locate this page using APP PATH or EDAPATH. See WebFOCUS Application Logic in the Developing Reporting Applications manual for details on search paths.

In Windows and UNIX, an HTML file called by a -HTMLFORM command must have the extension .HTM.

The display page links to an external Cascading Style Sheet you create in step 3.

**HTML Display Page: SCROLLPG.HTM**

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Cent Corp Demo KPI Template</title>
<link rel="stylesheet"
    href="/ibi_html/NewCentCorp/scrollss.css" type="text/css">
</head>
<body>
<div class="report1">!IBI.FIL.CREPORT1;</div>
<div class="graph1">!IBI.FIL.CGRAPH1;</div>
<div class="graph2">!IBI.FIL.CGRAPH2;</div>
<div class="report2">!IBI.FIL.CREPORT2;</div>
<div id="toolbar">
<img src="/ibi_html/NewCentCorp/images/wflogo.gif" width="96"
    height="14" border="0" alt="" style="position: relative;
top: 2px;">
</div>
</body>
</html>
```

3. Create a Cascading Style Sheet named SCROLLSS, which positions the reports and graphs on the display page, creates the scroll bars for report display, and sets text and background colors. The Web server must be able to locate the Cascading Style Sheet.

The letters on the left correspond to the notes explaining the code.
Cascading Style Sheet: SCROLLSS.CSS

a. { color: Navy;
  font-weight : bold;
}
/* the following rule controls the default styling for fonts in the application */
body, td {
  font-family : Verdana, Geneva, Arial, Helvetica, sans-serif;
  font-size : 12px;
  font-style : normal;
  font-variant : normal;
  font-weight : normal;
/* width : 10%; */
/* the rule below allows the scroll bars to be customized */
scrollbar-base-color : Blue;
scrollbar-arrow-color : white;
}
#toolbar {
  position: relative;
  top: 500;
  left: 0;
  width: 430px;
  height: 20pt;
  padding: 6px;
  background: 0033ff;
  border : thin outset;
}
/* the following defines position and properties of top left report */
.report1 { position: absolute;
  top: 30;
  left: 60;
  width: 200;
  height: 200;
  border: thin outset;
}
/* the following defines position and properties of top right graph */
.graph1 { position: absolute;
5. Enhancing a User Interface

c. top: 0;
   left: 440;
   overflow: auto;
}

/* the following defines position and properties of bottom left graph */
.graph2 { position: absolute;
   top: 246;
   left: 60;
   overflow: auto;
}

/* the following defines position and properties of bottom right report */
.report2 { position: absolute;
   top: 245;
   left: 455;
   width: 200;
   height: 200;
   border: thin outset;
   overflow: auto;
}

/* this class sets the styling for a WebFOCUS heading */
.HEAD {
  font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
  font-size: 14pt;
  font-weight: bold;
  color: #333366;
  position: relative;
  top: -13px;
}

/* this class sets the styling for WebFOCUS column titles */
.TITLE {
  font-family : Verdana, Geneva, Arial, Helvetica, sans-serif;
  font-size : 11pt;
  font-weight : bold;
  color: Navy;
  /*position: relative;
  top: 3px; */
}

/* this rule controls the defaults for all tables */
TABLE {
  border-collapse: collapse;
  /* table-layout : fixed; */
}

/* the two rules below controls styling for various headings */
h2 { font-size: 14pt; color: gold; }
h3 { font-size: 14pt; color: white; }

/* this rule sets the style for the mouse over effect on an
anchor */
a:hover { color: red; }

a.
Position the first report on the HTML page.

b.
Create scroll bars if the report cannot be entirely displayed at one time in a frame.

c.
Position the first graph on the HTML page.

d.
Set styling characteristics that are applied to WebFOCUS report headings.

e.
Set styling characteristics that are applied to WebFOCUS column titles.

4. Create a launch page that runs the procedure SCROLL to generate the following:

![Sales By Store]

**Sales By Store**

**Store**

**Name:**

- AV VideoTown: $175
- Audio Expert: $267
- City Video: $11
- Consumer Merchandise: $35
- TV City: $94
- **Web Sales:** $6,568

![Sales By Product]

**Sales By Product**

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD Players</td>
<td>$44</td>
</tr>
<tr>
<td>Camcorders</td>
<td>$406</td>
</tr>
<tr>
<td>Cameras</td>
<td>$15.5</td>
</tr>
<tr>
<td>DVD</td>
<td>$56</td>
</tr>
<tr>
<td>Digital Tape Recorders</td>
<td>$44</td>
</tr>
<tr>
<td>PDA Devices</td>
<td>$255</td>
</tr>
<tr>
<td>VCRs</td>
<td>$39.5</td>
</tr>
</tbody>
</table>
Displaying a Previously Run Report

**How to:**
Set Report Duration

You can display a previously run report in a browser without re-executing the request. WebFOCUS holds a report in a cache for a set period of time. If the output for a new request is the same as the output for a previous request, and it is still in the cache, the browser displays the previous report with the *Back*, *Refresh*, or *Reload* button.

The EXPIRE_REPORTS parameter sets the period of time for which a report is held in the cache. It is customizable for a WebFOCUS installation and affects all WebFOCUS output. You can change the EXPIRE_REPORTS parameter in the following ways:

- Set the EXPIRE_REPORTS value in the CGIVARS.WFS file, which is located by default as follows:
  - **Windows:** `install_drive:\ibi\client76\conf\etc`
  - **UNIX:** `/ibi/client76/conf/etc`
  - **z/OS:** `/ibi/client76/conf/etc`

- Add EXPIRE_REPORTS and its revised value to the SITE.WFS file, which overrides values in other .WFS files.

**Syntax:**

```
How to Set Report Duration

EXPIRE_REPORTS = {n|300}
```

where:

\( n \)

Is the number of seconds for which a report is held in the cache. The default value is 300.

To ensure that a report is re-executed, set EXPIRE_REPORTS to 1.

To view a browser’s cached output using the *Back*, *Refresh*, or *Reload* button, set EXPIRE_REPORTS to a large number, such as 4,000,000,000 seconds.
Passing a User ID From HTML for a Custom Menu

You can capture a user ID on an HTML logon page and pass it to a procedure. Use this technique to create menus that display only those application functions that a user is permitted to execute. For instance, one user may access menu options that apply to a particular role in the organization, and another user may access certain sensitive capabilities. The choice of menu options is based on the user ID.

The procedure in this topic runs on WebFOCUS for Windows.
Customizing a Menu

How to:
Create the HTML Logon Page (Step 1)
Create the Home Page (Step 2)
Create the Frame Pages (Step 3)
Modify the SITE.WFS Files (Step 4)
Create the User Data Source (Step 5)
Create the Menu Data Source (Step 6)
Create the Procedure (Step 7)
Create the HTML Display Page (Step 8)
Create the Forms That Call the Report Selections on the Menu (Step 9)
Create the Procedures (Step 10)

The following is a summary of steps that you will complete in the example. You can modify the steps for your own application requirements. The specified file locations are the Windows file locations used in this procedure.

<table>
<thead>
<tr>
<th>Step</th>
<th>File Name</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create an HTML logon page. A cookie is created after successful logon, containing the user ID (IBIC_user) entered on the page.</td>
<td>SIGNON.HTM</td>
</tr>
<tr>
<td>Step</td>
<td>File Name</td>
<td>File Location</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>2</td>
<td>HOME.HTM</td>
<td>WebFOCUS76\ibi_html</td>
</tr>
<tr>
<td>3</td>
<td>MENU.HTM LOADMEM.HTM WELCOME.HTM</td>
<td>WebFOCUS76\ibi_html</td>
</tr>
<tr>
<td>4</td>
<td>CGIVARS.WFS</td>
<td>ibi\client76\conf\etc</td>
</tr>
<tr>
<td>5</td>
<td>USERLST.MAS USERLST.DAT</td>
<td>srv76\ggdemo</td>
</tr>
<tr>
<td>Step</td>
<td>File Name</td>
<td>File Location</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>6</td>
<td>MENUST.MASMENULST.DAT</td>
<td>srv76\ggdemo</td>
</tr>
<tr>
<td>7</td>
<td>FHSUB.FEX</td>
<td>srv76\ggdemo</td>
</tr>
<tr>
<td>8</td>
<td>SUBMENU.HTM</td>
<td>srv76\ggdemo</td>
</tr>
<tr>
<td>9</td>
<td>1MYGRAPH.THM2MYGRAP.HTM</td>
<td>WebFOCUS76\ibi_html</td>
</tr>
<tr>
<td>10</td>
<td>FHSUB.FEX</td>
<td>srv76\ggdemo</td>
</tr>
</tbody>
</table>
Note:

- Procedures used by the WebFOCUS Reporting Server to create output, and HTM files in which the output is embedded, must be accessible to the WebFOCUS Reporting Server.
- Launch pages, menus that request reports, and images are stored in the WebFOCUS Client on the Web server.

Procedure: How to Create the HTML Logon Page (Step 1)

The following file is named SIGNON.HTM. Modify the lines in bold to apply to the Web server at your site.

SIGNON.HTM:

```html
<html>
<head>
<meta http-equiv="Content-Language" content="nl-be">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Please Identify yourself!</title>
</head>
<body stylesrc="submenu.htm" bgcolor="#CECF9C">
<p>&nbsp;</p>
<form method="POST" action="/ibi_apps/WFServlet" name="Signon">
<table border="0" width="100%" height="112">
<tr>
<td width="66%" height="19" align="right" colspan="2">
<p align="center"><img border="0" src="http://localhost/ibi_html/IB_LOGO.GIF"></p>
</td>
</tr>
</form>
</body>
</html>
```
Log on to WebFOCUS reporting

<table>
<thead>
<tr>
<th>User ID</th>
<th>Input type: text</th>
<th>Name: IBIC_user</th>
<th>Size: 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Input type: password</td>
<td>Name: IBIC_pass</td>
<td>Size: 20</td>
</tr>
</tbody>
</table>

<input type="reset" value="Reset" name="B2" />

<input type="submit" value="Submit" name="B1" />

<input type="hidden" name="IBIWF_action" value="WF_SIGNON" />
<input type="hidden" name="IBI_random" value="" />
<input type="hidden" name="WF_SIGNON_MESSAGE" value="http://localhost/IBI_HTML/HOME.HTM" />

Passing a User ID From HTML for a Custom Menu

The following is the SIGNON.HTM page when accessed in the Web browser.

![SIGNON.HTM page](image1)

The following is the SIGNON.HTM page when accessed in the Web browser.

![SIGNON.HTM page](image2)
**Procedure:** How to Create the Home Page (Step 2)

In this example, you create the home page with a frameset. The following HTML files supply the content for each frame used: MENU.HTM (banner frame), LOADMEN.HTM (contents frame), and WELCOME.HTM (query frame).
The following file is named HOME.HTM. Modify the lines in bold to apply to the Web server at your site.

**HOME.HTM:**

```html
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Webfocus Reporting</title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta name="Microsoft Border" content>
<meta base href="http://localhost/">
</head>

<frameset id="header" rows="68,*" framespacing="0" border="0" frameborder="0">
  <frame name="banner" scrolling="no" noresize target="contents" src="MENU.HTM" marginwidth="0" marginheight="0" NOBORDER>
  <frameset id="bodypart" cols="146,*">
    <frame name="contents" target="main" src="LOADMEN.HTM" scrolling="auto" marginwidth="3" marginheight="11">
    <frameset id="querypart" rows="80,*">
      <frame name="QueryFrame" src="WELCOME.HTM" target="bottom" marginwidth="0" marginheight="0" scrolling="no" noresize>
      <frame name="ReportFrame" src="WELCOME.HTM" scrolling="auto" noresize marginwidth="12" marginheight="1" target="_self">
    </frameset>
  </frameset>
</frameset>
<noframes>
<body>
  <p>This page uses frames, but your browser doesn't support them.</p>
</body>
</noframes>
</frameset>
</html>
```

**Procedure:** How to Create the Frame Pages (Step 3)

1. The following file is named MENU.HTM. It supplies the content for the top frame (banner). Modify the lines in bold to apply to the Web server at your site.
Passing a User ID From HTML for a Custom Menu
The following code validates the query.
function dovalidation(Checktype){
var boodschap=''
var frm=parent.frames('QueryFrame').frminput
// Check Required !!!
for (x=0;x<Checktype.length;x++) {
if (Checktype[x][2]=='R') {
if (frm.elements(Checktype[x][0]).value == '') {
boodschap=boodschap + '"'+Checktype[x][1]+'" is mandatory ! \n'
} }
}
// End Check Required
if (boodschap != '') {return boodschap};
// Check Type !!!
for (x=0;x<Checktype.length;x++)
{
if (Checktype[x][3]=='N')
{
var anum=/(^\d+$)|(^\d+\,\d+$)/
if (anum.test(frm.elements(Checktype[x][0]).value))
{
if ((Checktype[x][4] > frm.elements(Checktype[x][0]).value) ||
(Checktype[x][5] < frm.elements(Checktype[x][0]).value))
{
boodschap=boodschap+'"'+Checktype[x][1]+'" must be within range
('+Checktype[x][4]+'-'+Checktype[x][5]+')! \n';
}
else
{}
}
else
{
boodschap=boodschap+'"'+Checktype[x][1]+'" must be a number !
\n';
}
}
else
{
if (Checktype[x][3]=='T')
{
if (frm.elements(Checktype[x][0]).value.length < Checktype[x][4]
||
frm.elements(Checktype[x][0]).value.length > Checktype[x][5])

222

WebFOCUS


```javascript
{ boodschap=boodschap+""'+Checktype[x][1]+'"length must be within range ('+Checktype[x][4]+'-'+Checktype[x][5]+')! \n'; }

} else 
{

if((Checktype[x][3]=='D')&&(frm.elements(Checktype[x][0]).value.length > 0))
{
    var dateStr = frm.elements(Checktype[x][0]).value;
    var datePat = /^([d1,2])(\-\d(1,2)\2(\d2)\2(\d4)\2)/;
    // To require a 4 digit year entry, use this line instead:
    // var datePat = /^([d1,2])(\-\d(1,2)\2(\d2)\2(\d4)\2)/;

    var matchArray = dateStr.match(datePat);
    if (matchArray == null) {
        boodschap=boodschap+""'+Checktype[x][1]+'" is not a valid date!
            \n';
        return boodschap;
    }
    day = matchArray[1];
    month = matchArray[3];
    year = matchArray[4];
    if (month < 1 || month > 12) {
        boodschap=boodschap+""'+Checktype[x][1]+'" :month must be
            between 1 and 12.';
        return boodschap;
    }
    if (day < 1 || day > 31) {
        boodschap=boodschap+""'+Checktype[x][1]+'" :day must be between
            1 and 31.';
        return boodschap;
    }
    if (((month==4 || month==6 || month==9 || month==11) && day==31) {
        boodschap=boodschap+""'+Checktype[x][1]+'" : month '+month+' does not have 31 days!';
        return boodschap;
    }
    if (month == 2) {
        var isleap = (year % 4 == 0 && (year % 100 != 0 || year % 400
            == 0));
        if (day>29 || (day==29 && !isleap)) {
            boodschap=boodschap+""'+Checktype[x][1]+'" : february ' + year
                + ' does not have ' + day + ' days!';
            return boodschap;
        }
```

Developing Reporting Applications With Graphical Tools 223
Passing a User ID From HTML for a Custom Menu

function breakout() {
    mywindow=window.open("","Titel","scrollbars=yes,status=no");
    mywindow.location.href=parent.frames('ReportFrame').location.href;
}

</script>

<base target="contents">
</HEAD>

<BODY style="font-family: Verdana; font-size: 8pt" class="regular"
    bgcolor="#000000" link="#CECF9C" vlink="#CECF9C" alink="#CECF9C">

<table border="0" width="600" bgcolor="#000000" cellspacing="0"
cellpadding="0" bordercolor="#000000" height="62">
    <tr>
        <td>&nbsp;</td>
        <td width="57" align="center" height="1">
            <p align="center"><font color="#CECF9C" size="1">224 WebFOCUS</font></p>
        </td>
    </tr>
</table>
2. The following file is named LOADMEN.HTM. It supplies the content for the left-hand frame (contents). Modify the lines in bold to apply to the Web server at your site.

**LOADMEN.HTM:**

```html
<html>
<head>
<title>Welcome text</title>
<SCRIPT LANGUAGE="JavaScript">
<!--
function loadmenu() {
IBI_random=Math.random()*Math.random()*Math.random()*Math.random()*100 ;
loadurl="http://localhost/ibi_apps/
WFServlet?IBIF_ex=fhsub&IBI_random"+IBI_random
parent.frames[1].location.href=loadurl
}
// -->
</SCRIPT>
</head>
<body bgcolor="#E0E0BE" onload="loadmenu()">
<p align="center">Loading Menu ....</p>
<p align="center">Stand by ..</p>
</body>
</html>
```
3. The following file is named WELCOME.HTM. It provides the initial blank frame that will be replaced by report output. No modification is required.

**WELCOME.HTM:**

```
<html>
<head>
<meta http-equiv="Content-Language" content="nl-be">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Welcome text</title>
<base target="_self">
</head>
<body bgcolor="#E0E0BE"></body>
</html>
```

**Procedure:** How to Modify the SITE.WFS Files (Step 4)

The following is an example of modifying the SITE.WFS files.

1. Open the file SITE.WFS, located by default as follows
   
   **Windows:** `drive:\ibi\client76\conf\etc`
   
   **UNIX:** `/ibi/client76/conf/etc`
   
   **z/OS:** `/ibi/client76/conf/etc`
   
   **Note:** At many locations, developers will not have access to the .WFS files on the Web server. Contact your systems administrator or other appropriate staff.

2. In IBICOMMD.WFS, add the following lines after the code `<ifndef>EDACS3... <endif>`:

   ```
   <sendvar>
   RUSER=&IBIC_user
   </sendvar>
   ```

   If you are modifying SITE.WFS, add these lines after the comment.
   
   This step will send the user ID to a Dialogue Manager variable named &RUSER, which will be used for selecting menu options in the FHSUB procedure.

3. Save the file and exit.
Procedure: **How to Create the User Data Source (Step 5)**

The following Master File for the user data source is named USERLST.MAS. This file must be created in the WebFOCUS Reporting Server’s search path, either the APP PATH or EDAPATH in EDASPROF.PRF. For details, see *WebFOCUS Application Logic* in the *Developing Reporting Applications* manual.

```plaintext
FILENAME=USERLST, SUFFIX=FIX,
SEGNAME=USERLST,SEGTYPE=S0,$
  FIELD=USER ,ALIAS=USER, USAGE=A8 , ACTUAL=A8 ,$ user ID
  FIELD=MKEY ,ALIAS=MKEY, USAGE=A2  , ACTUAL=A2  ,$ menu key
  FIELD=DUMM ,ALIAS=DUMM, USAGE=A70 , ACTUAL=A70 ,$ filler
```

The data source is USERLST.DAT:

- CEO 01
- ADMIN 02

The user ID CEO has access only to menu option 01. The user ID ADMIN has access to menu option 02.

Procedure: **How to Create the Menu Data Source (Step 6)**

The following Master File for the menu data source is named MENULST.MAS:

```plaintext
FILENAME=MENULST, SUFFIX=FIX,
SEGNAME=TCSTAB04,SEGTYPE=S0,$
  FIELD=MKEY , ALIAS=MKEY, USAGE=A2  , ACTUAL=A2  , INDEX=I ,$ menu key
  FIELD=RAP  , ALIAS=RAP , USAGE=A25 , ACTUAL=A25 ,$ menu text
  FIELD=HTM  , ALIAS=HTM , USAGE=A35 , ACTUAL=A35 ,$ HTML page
  FIELD=FRM  , ALIAS=FRM , USAGE=A12 , ACTUAL=A12 ,$ target frame
  FIELD=DUMM , ALIAS=DUMM , USAGE=A6  , ACTUAL=A6  ,$ filler
```

The data source is MENULST.DAT:

- 01REPORT1  ibi_html/1myqrap.htm  QueryFrame
- 02REPORT2  ibi_html/2myqrap.htm  QueryFrame

Procedure: **How to Create the Procedure (Step 7)**

The following file is named FHSUB.FEX. The JOIN command associates users with their permitted menu options. The DEFINE command dynamically defines the hyperlinks that will appear on the custom menu.

The value for the Dialogue Manager variable &RUSER is passed to the procedure by the `<sendvar>` block in the file SITE.WFS.
Modify the lines in bold to apply to your site.

```
-************************************************************************
-* Calling from : LOADMEN.htm
-* Files called : sub.htm
-* Used &VARS   : &RUSER -> Web user ID (IBIC_user or equivalent)
-* Files used   : MENULIST.DAT -> fixed file with menu items
-*                USERLST.DAT  -> fixed file with users and privileges
-* Files created: rlist.htm -> temporary file with menu list
-*                H1.FOC    -> temporary file used in JOIN
-************************************************************************

-************************************************************************
-* Change these for customization:
-* &HOMEURL : Web server URL
-* &LSTDIR  : Location of userlst.dat and menulst.dat
-************************************************************************

-SET &ECHO=ALL;
-SET &HOMEURL='HTTP://localhost/';
-SET &LSTDIR='drive\IBI\APPS\GGDEMO\';
-SET &EDAHTMDIR='drive\IBI\APPS\GGDEMO\';
-SET &USERDAT='&LSTDIR.EVAL' || 'userlst.dat' ;
-SET &MENUDAT='&LSTDIR.EVAL' || 'menulst.dat' ;
-SET &LISTLOC='&EDAHTMDIR.EVAL' || 'rlist.htm' ;
-SET &RUSERL=&RUSER.LENGTH;
-SET &OLENGTH='A' || &RUSERL.EVAL;
-SET &UPUSER=UPCASE(&RUSERL.EVAL,'&RUSER.EVAL','&OLENGTH.EVAL');
FILEDEF USERLST DISK &USERDAT.EVAL
FILEDEF MENULST DISK &MENUDAT.EVAL
FILEDEF RLIST   DISK &LISTLOC.EVAL
-Run

************************************************************************
- This code creates temporary file containing all menu items
- with index on MKEY fields needed in JOIN.
************************************************************************
TABLE FILE MENULST
PRINT *
ON TABLE HOLD AS H1 FORMAT FOCUS INDEX MKEY
END
-Run

************************************************************************
- This will build the HTML syntax needed for dynamic user menu
- with selection on user ID.
************************************************************************
```
JOIN MKEY IN USERLST TO ALL MKEY IN H1 AS A
DEFINE FILE USERLST
TAGP1/A25='<TR> <TD WIDTH="100%" >' ;
TAGP2/A150=
' <A HREF=' || '&HOMEURL.EVAL' || HTM ||
' TARGET="' || FRM || '" ONCLICK="Prepare();" >' ;
TAGP3/A80='<FONT SIZE="1" COLOR="#000000" >'|| RAP ||
' </FONT></A></TD></TR>' ;
END
TABLE FILE USERLST
PRINT TAGP1 TAGP2 TAGP3
WHERE USER EQ '&UPUSER.EVAL';
ON TABLE HOLD AS RLIST FORMAT ALPHA
END
-RUN
******************************************************************************
* This will call the HTM file that contains the !IBI.FIL syntax.
******************************************************************************
-HTMLFORM SUBMEN

Procedure: How to Create the HTML Display Page (Step 8)

The HTML display page is called by the procedure, and must be accessible by the WebFOCUS Reporting Server. In Windows and UNIX, the extension is always .HTM.
The following file is named SUBMEN.HTM. It is merged with the report output generated by the procedure FHSUB. Modify the lines in bold to apply to the Web server at your site.

```html
<html>
<head>
<meta http-equiv="Content-Language" content="nl-be">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Submenu</title>

<SCRIPT LANGUAGE='JavaScript'>
<!-- shields up!
function Prepare () {
parent.document.all("header").all("bodypart").all("QueryPart")
 .all("Reportframe").src="welcome.htm"
parent.document.IBI_random=Math.floor((Math.random()*100000));
}

function Hide_query()
{
parent.document.all("header").all("bodypart").all("QueryPart")
 .rows="26,*";
parent.document.all("header").all("bodypart").all("QueryPart")
 .all('QueryFrame').src='welcome.htm';
}

function set_rand()
{
parent.document.IBI_random=Math.floor((Math.random()*100000));
// -->
</SCRIPT>

<style fprolloverstyle>A:hover {color: #808000; font-size: 10pt;
 font-family: Verdana; font-weight: bold}
</style>
<meta base href="http://localhost/ibi_html" target="main">
<base target="main">
</head>

<body style="font-family: Verdana; font-size: 8pt" bgcolor="#CECF9C"
onload="set_rand()">
```
Procedure: How to Create the Forms That Call the Report Selections on the Menu (Step 9)

1. The following file is named 1MYQRAP.HTM. No modification is required. The file must be stored in the WebFOCUS client under the directory ibi\WebFOCUS\ibi_html.
function Form1_Validator(theForm) {
    var boodschap = ''
    var terug = true
    //
    //Checktype([[FieldName,Text, Required, Type, rangefrom, rangeto]])
    // Type=(T,N,D), T -> Rangefrom & rangeto = length, N > range= real
    // range !!! - Daterange not yet supported
    //
    var Checktype=[[[],[]]];
    //
    // End of Validation definition
    //
    //
    parent.frames("Reportframe").document.write('
    <b><font face="Verdana" size="2">Please Wait...<BR>WebFocus is
    working for you</font></b></p>');</n
    fun1()
    return (terug)
}

1MYQRAP.HTM:
function Hide_query()
{
parent.document.all("header").all("bodypart").all("QueryPart")
.rows="26,*";
}

function resize()
{
var x=document.all("Query").offsetHeight-10;
parent.document.all("header").all("bodypart").all("QueryPart")
.rows=x+",*";
}

function checkaction()
{
if
(parent.document.all("header").all("bodypart").all("QueryPart").rows
== '26,*')
{ resize()
} else {
parent.document.all("header").all("bodypart").all("QueryPart")
.rows='26,*';
}
}

function fun1()
{
document.frminput.IBI_random.value=Math.floor((Math.random()*100000));
}
//--></script>

<form method="GET" action="/ibi_apps/WFServlet"
style="font-weight:bold" name="frminput"
onsubmit="return Form1_Validator(this)" target="ReportFrame">
<input type="hidden" name="IBIF_ex" value="REPORT1">
<input type="hidden" name="IBI_random" value="">
<table name="tble" border="0" width="759" cellspacing="1"
style="font-family:Verdana; font-size: 8pt" height="94">
<tr>
<td width="759" align="right" bgcolor="#CFD09F" colspan="4"
height="11">
<p align="left"><font face="Verdana" size="1">Please provide Parameters for Report 1</font></p>
</td>
</tr>
</table>
</form>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION</td>
<td>ACTION</td>
</tr>
<tr>
<td>MYSTERY</td>
<td>MYSTERY</td>
</tr>
<tr>
<td>CLASSIC</td>
<td>CLASSIC</td>
</tr>
<tr>
<td>CHILDREN</td>
<td>CHILDREN</td>
</tr>
<tr>
<td>MUSICALS</td>
<td>MUSICALS</td>
</tr>
</tbody>
</table>

<input type="submit" value="Run" name="B1">
2. The following file is named 2MYQRAP.HTM. No modification is required.

**2MYQRAP.HTM:**

```html
<html>
<head>
<title>Query</title>
<meta name="Microsoft Border" content>
</head>

<body name="test" topmargin="1" style="font-family:Verdana;
font-size:8pt" bgcolor="#D8D9B0" onload="resize()">

<DIV ID="Query" style="width: 713; height: 159">
<script Language="JavaScript">

function Form1_Validator(theForm){
  var boodschap = ''
  var terug = true

  //Checktype[[Fieldname,Text, Required, Type, rangefrom, rangeto]]
  // Type=(T,N,D), T -> Rangefrom & rangeto = length, N > range= real
  // Range !!! - Daterange not yet supported

  var Checktype=[[[]],[[]]];

  // End of Validation definition

  parent.frames("Reportframe").document.write('<br>
<p align="center"><b><font face="Verdana" size="2">Please
Wait...<BR>WebFocus is working for you</font></b></p>');

  fun1()
  return (terug)
}

function Hide_query()
{
parent.document.all("header").all("bodypart").all("QueryPart")
.rows="26,*";
}

function resize()
{var x=document.all("Query").offsetHeight-10;
parent.document.all("header").all("bodypart").all("QueryPart")
.rows=x"," *
}

function checkaction()
```
function fun1() {
    document.frminput.IBI_random.value=Math.floor((Math.random()*100000));
}

<form method="GET" action="/ibi_apps/WFServlet"
    style="font-weight: bold" name="frminput"
onsubmit="return Form1_Validator(this)" target="ReportFrame">
    <input type="hidden" name="IBIF_ex" value="REPORT2">
    <input type="hidden" name="IBI_random" value="">
    <table name="tble" border="0" width="759" cellspacing="1"
        style="font-family:Verdana; font-size: 8pt" height="94">
        <tr>
            <td width="759" align="right" bgcolor="#CFD09F" colspan="4"
                height="11">
                <p align="left"><font face="Verdana" size="1">
                    <u><b>Please provide Parameters for Report 2</b></u></font>
                </p>
            </td>
        </tr>
        <tr>
            <td width="82" align="right" bgcolor="#D8D9B0" height="25">
                <font face="Verdana" size="1">Variable 1</font>
            </td>
            <td width="212" bgcolor="#D8D9B0" height="25">
                <select size="1" name="AMP_VAR1" style="font-family:Verdana;
                    font-size: 8pt">
                    <option value="ACTION">ACTION</option>
                    <option value="MUSICALS">MUSICALS</option>
                    <option value="CLASSIC">CLASSIC</option>
                    <option value="CHILDREN">CHILDREN</option>
                    <option value="FOREIGN">FOREIGN</option>
                    <option value="MYSTERY">MYSTERY</option>
                </select>
            </td>
        </tr>
    </table>
</form>
Variable 2

<table>
<thead>
<tr>
<th>Variable 1</th>
<th>ACTION</th>
</tr>
</thead>
</table>

Run
**Procedure:** How to Create the Procedures (Step 10)

The following file is named REPORT1.FEX.

```
DEFINE FUNCTION SUBTRACT (VAL1/D8, VAL2/D8)
SUBTRACT/D8.2 = VAL1 - VAL2;
END
TABLE FILE MOVIES
PRINT TITLE LISTPR IN 35 WHOLESALEPR AND COMPUTE
PROFIT/D8.2 = SUBTRACT(LISTPR,WHOLESALEPR);
BY CATEGORY
WHERE CATEGORY EQ '&AMP_VAR1';
ON TABLE HOLD FORMAT HTMTABLE AS REPORT1
END

-HTMLFORM BEGIN
<HTML>
<BODY>
<H2>MOVIE SALES PROFIT</H2>
<!--WEBFOCUS TABLE REPORT1>
<HR>
<P>Parameters selected </P>
<HR>
<UL>
<LI> CATEGORY: &AMP_VAR1
<LI> CURRENT DATE : &DATEDMYY
<LI> CURRENT TIME : &TOD
<LI> CURRENT USER : &RUSER
</UL>
</BODY>
</HTML>
-HTMLFORM END
```
After you create, test, and debug a project in the development environment, you are ready to make it available as a live application on the Web environment. This process involves moving certain project files to selected target servers in a WebFOCUS environment. Developer Studio automates the process for you.

This section describes the steps you will perform for successful deployment of your project files to the Web environment.

Topics:
- Deployment Basics
- Summary of Steps
- Step 1: Identify the Target Servers
- Step 2: Create a Deployment Scenario
- Step 3: Partition the Project Files
- Step 4: Deploy the Project Files
- Configuring the Target Servers (Optional)
Deployment Basics

Once you create project files as described in Creating a Reporting Application on page 27, you can partition and deploy them to a WebFOCUS environment.

A WebFOCUS environment consists of:

- A Web server
- A WebFOCUS Reporting Server
- Alternate server nodes defined in the Data Servers area

**Note:** The Web environment may be a Web server or application server or a combination of both, depending on your environment. This topic refers to deployment while using a Web server to hold the static content, for example, .html and .gif files and required configuration for such an environment.

If you need to place static content on the application server, you must define an Application Root directory (APPROOT) context root for deploying files and perform required configuration steps. The APPROOT context root is used during the deploy process and when you run the deployed Web-based application.

**Partitioning** the files means that you identify the target Web server on which the Web-based files will reside, and the WebFOCUS Reporting Server on which files such as procedures will reside. You define the partitioning of files in a deployment scenario.

**Deployment** means that Developer Studio moves the files to the target servers, where they make up an application that users run on the Web.

The Developer Studio deployment feature allows you to:

- Create multiple deployment scenarios. You can define multiple deployment scenarios and save them for future use. For example, you might have two deployment scenarios for a project, one that maps the project files to a test environment, and another that maps the files to a production environment.

- Deploy files to multiple servers. You can deploy project files to multiple WebFOCUS Reporting Servers or Maintain servers. This capability enables you to access data on multiple servers, run report components in the most suitable environment, and accelerate application processing.

**Summary of Steps**

To take your project live, you will do the following:

- **Step 1: Identify the Target Servers** on page 241
- **Step 2: Create a Deployment Scenario** on page 241
Step 3: Partition the Project Files on page 249

Step 4: Deploy the Project Files on page 256

You can optionally configure the target servers for your site. For details, see Configuring the Target Servers (Optional) on page 261.

Step 1: Identify the Target Servers

You must identify the target servers that are going to receive the project files. For details on identifying servers, see Managing the WebFOCUS Environment in the Developing Reporting Applications manual, and the installation and configuration documentation for Developer Studio.

Step 2: Create a Deployment Scenario

How to:

Create a Deployment Scenario
View the Deploy Namespace Properties

A deployment scenario provides a means of identifying project files for distribution to target servers in a WebFOCUS environment.

You can create more than one deployment scenario for a project, and you can view and modify the properties of an existing scenario.

Developer Studio supplies a deployment scenario named Local Deploy, which it uses to prepare files to run on a local server. Do not modify or remove Local Deploy.

Procedure: How to Create a Deployment Scenario

1. Select and right-click the project.
2. From the shortcut menu, select Project Deployment, then select New Scenario.
The New Deployment Scenario dialog box opens.

3. Respond as follows:
   - **Scenario name.** Type a name for the new deployment scenario. You can change the name at a later time with the Rename option.
- **Use smart deploy.** Choose this option to deploy only the files that have changed since the last time you deployed the application. This option can significantly increase the speed of deployment.

  When Smart Deploy is active for a Deployment Scenario, after its first successful deployment, date/time information is stored in your Project GFA file for each file being deployed. Each subsequent Deploy action then compares the date-time stamp of the source files in your project against the date/time stamps stored in the Deployment Scenario. With Smart Deploy, only newer files will be deployed to the destination, where the source file has a newer time stamp than the one stored in the GFA.

  **Note:**

  The date/time information is stored uniquely in each deployment scenario, so if your project has multiple deployment scenarios, there is no ambiguity about when the files were last deployed using a specific scenario. Take note that each time you first use a new deployment scenario, Smart Deploy performs a full deployment of all files referenced for deployment in your project; subsequently only changed files will be deployed.

  Also note that Smart Deploy stores file date/time stamps in the GFA after a successful deployment. If an error occurs during a first deployment of a new scenario, the date/time stamps are rolled back.

  If you add a new file to be deployed to a Deployment Scenario, the GFA has no date/time stamp for a file, so it will always be deployed the first time.

- **Ignore unresolved items.** This option ensures that deployment is completed in a case where the project deployment scenario contains an unresolved file. An unresolved file can be either:

  A project file that was used in a deployment scenario and deleted outside of Developer Studio (or from another project in which it was shared).

  or

  A file located in a directory that changed and can no longer be found.

  If this option is selected, the deployment scenario will not fail if it contains unresolved files. However, when the deployment process completes, it will display information about files skipped during the process because they were unresolved.

- **Compile maintain procedures.** Choose this option to compile Maintain procedures during deployment. Compilation improves application performance. For more information, see the *Developing WebFOCUS Maintain Applications* manual.
Compile for verbose maintain trace (debug). This option is reserved for debugging purposes. It can affect performance. For more information, see the Developing WebFOCUS Maintain Applications manual.

Include Server Paths. Choose this option to allow the deployment process to include server paths in the Call and Exec statements within Maintain procedures. Do not choose this option when deploying Maintain applications to platforms that do not support APP PATH commands (for example, VM).

Keep the form generation code in deployed .MNT file. Choose this option to deploy .MNT files with any original form code intact. This option is enabled by default. A decrease in the speed of deployment may be noticed if a project contains many forms.

Although this code (XML) is not needed at run time once the .WFM files have been created during the deploy process, keeping it in the remote copy of the file can be an added safeguard should anything happen to the original version. (MNT files deployed without the form code cannot be reopened in an MDE session.)

Note: The XML code for the form should never be edited manually; use the MDE to open and edit .MNT files containing forms.

Click Next to proceed to the next Scenario Wizard window.
4. Respond as follows:

- **Default partition environment.** From the drop-down list, select a WebFOCUS environment that will be used by default when you partition files. Developer Studio automatically partitions project files for you based on the following:

  HTML, GIF, JPG, VBScript, and JavaScript files are partitioned to the default Web server. Executable WebFOCUS procedures (.FEX and .MNT files) are partitioned to the default WebFOCUS Reporting Server.

  By default, Developer Studio does not partition Master Files and Access Files in order not to overwrite production files with the same name on the deployment server.

  Selecting an environment from this list does not force the deployment of files as defined in the scenario.

  You can change the definition of the scenario.

- **Default reporting server.** From the drop-down list, you can select which WebFOCUS Reporting Server to use as your Default partition environment.

  This list is not active if you choose the value NONE for the Default partition environment.

- **Target application name.** Optionally, type an application name that will be used on the Web server and WebFOCUS Reporting Server deployment paths, or use the default value (the current project name).

  If you are deploying files to your development server, you must provide a target application name other than your development directory.

  If you are deploying files to a WebFOCUS Reporting Server on z/OS, the target application name can be up to 8 characters. For deployment servers on all other supported platforms, the name can be up to 18 characters. Spaces are not allowed. (If a space is entered, it is converted to an underscore.)

- **Automatically partition the project files.** This sets default locations based on file types.
Click Next to proceed to the next Scenario Wizard window.

5. Respond as follows:

- **No starting object.**
  - **Select from project items.** From the drop-down list, you can select a file from the application as the starting object.
    
    This option is enabled only when files are added to the application and when the option to partition files has been selected.

- **Select From WebFOCUS Environments.** Allows you to select a starting object from an existing configured WebFOCUS Environment, for example, a general launch page.

  You must ensure that the deployed application has access to the file to avoid an error at run time.

- **Select a web page.** Allows you to select a URL as a starting object. To specify a Web page, type the URL as a complete path to a file.

  You must ensure that the deployed application has access to the file to avoid an error at run time.
6. Click Next to proceed to the next Scenario Wizard window.

6. Click Finish to complete the deployment process. The following screen appears.
7. Close or minimize the deployment scenario window, known as the Scenario Editor to return to the Explorer. The project displays a new folder labeled Deploy. Later, when you have defined deployment scenarios, you can display them by clicking the folder.

**Procedure: How to View the Deploy Namespace Properties**

1. Right-click the Deploy namespace and choose *Properties*.

   The Deploy Properties dialog box opens, displaying the General and Comment tabs.

   The General tab is selected by default. It displays the CLSID, ProgID, and a description.

2. Optionally, type a descriptive comment for the deploy namespace in the Comment field.
Step 3: Partition the Project Files

In this section:
- Working in a Default Partition Environment
- Setting Running Paths

How to:
Partition the Project Files

Reference:
Guidelines for Deploying Files

After creating a deployment scenario, you define the details of that scenario by identifying project files for distribution to target servers in a WebFOCUS environment.

This topic applies to deployment scenarios that you create in Step 2: Create a Deployment Scenario on page 241. It does not apply to the supplied scenario Local Deploy. Developer Studio uses Local Deploy to prepare files for Maintain and to run on a local server. This option is visible only after you create the first deployment scenario. Do not modify or remove Local Deploy.

Procedure: How to Partition the Project Files

1. Open the Deploy folder for the project. Right-click the deployment scenario that you are defining, and select Open.

2. The left window pane displays information about the individual files that are visible from the project’s search path.

   The icons next to some files are grayed out. This indicates that the files are available, but not active for your project. You can click the binoculars icon to show the files that you added to your project and hide the inactive files. However, you can deploy even the files that you did not add to your project.

   The value of Assigned is No if the file has not been assigned to a server for deployment. Assigned is Yes if you have assigned the file to a server for deployment.

   Note: If you cannot see the value in the Assigned column or the full application path in the Location column, drag your cursor to the right to expand the Application Files window.

3. The right window pane displays the environments you have identified and optionally configured to accept project files.
Step 3: Partition the Project Files

For a Web server, you can create one or more subfolders on the deployment path. For example, you might separate HTML files from image files by creating an individual folder, or central location, for each type:

a. Select and right-click the current Web server deployment path (for example, WEB/APPROOT/Sales).

b. Select New Folder from the shortcut menu. At the cursor location in the entry field, enter the name of the new folder.

Note: You must design the application properly in order to take advantage of a multilevel directory structure on the deployment environment.

4. In the right pane, open the environment on which you are partitioning the files. In this example, the Test Environment includes a Web server and a server node that can be used for deployment.

In the example:

- The Web server is defined by the node WEB, followed by the descriptive name APPROOT and the name of the directory that will be created when the files are deployed (Sales).
A WebFOCUS Reporting Server is defined by the node EDA and the name of the server (for example, EDASERVE), followed by the descriptive name APPPATH and the name of the directory that will be created when the files are deployed (Sales).

The developer selected NONE as the default partition environment when creating the new deployment scenario. For an example of the window that opens when you select a default partition environment, see Working in a Default Partition Environment on page 253.

5. Drag one or more files from the left window pane and drop them into the right window pane on the applicable server.

See Guidelines for Deploying Files on page 252 for information that will help you deploy files to the applicable server.

To remove a partitioned file from the right window pane, right-click it and select Remove.

In this example, the file named launch_qty.htm is an HTML page from which a user can run a report named stockqty.fex. To partition these files appropriately, you would drag launch_qty.htm to the Web server, and stockqty.fex to the WebFOCUS Reporting Server.

On the toolbar, these buttons enable you to do the following while you remain in the Scenario Editor:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Deploy scenario" /> - <strong>Deploy scenario</strong></td>
<td>Deploys (moves) the files to the servers based on the partitioning defined in the current deployment scenario.</td>
</tr>
<tr>
<td><img src="image" alt="Deploy and run scenario" /> - <strong>Deploy and run scenario</strong></td>
<td>Deploys the files to the servers and then runs the starting object defined in the current deployment scenario.</td>
</tr>
<tr>
<td><img src="image" alt="Run scenario" /> - <strong>Run scenario</strong></td>
<td>Runs the starting object defined in the current deployment scenario.</td>
</tr>
<tr>
<td><img src="image" alt="Displays all files in the project path" /> - <strong>Displays all files in the project path</strong></td>
<td>Displays available items in the project’s path. The icons next to some files are grayed out. This indicates that the files are available, but not active for your project. You can click the binoculars icon to show the files that you added to your project and hide the inactive files. However, you can deploy even the files that you did not add to your project.</td>
</tr>
</tbody>
</table>
These buttons are also available on the toolbar in the Explorer.

6. When you have finished partitioning files, choose one of the options in step 5, or close the window. You can perform any of the options in step 5 outside the Scenario Editor. See Step 4: Deploy the Project Files on page 256.

**Reference:** Guidelines for Deploying Files

- Typically you will deploy HTML, GIF, JPG, VBScript, and JavaScript files to a Web server. Executable WebFOCUS procedures (.fex, .mnt, and .sty files) will go to a WebFOCUS Reporting Server.

  If the deployed FOCUS procedure has a reference to a stylesheet (.sty) file and the generated procedure uses a stylesheet different from the default, you must deploy the stylesheet to the Reporting Server, or else the report will not run.

  To view and deploy the .sty file, you must add the filter for this file to the project level or an available virtual folder (for example, Procedures) to view files of this type in the project and in the Scenario Editor.

  Usually you will not deploy Master Files (.MAS) or Access Files (.ACX). Master Files and Access Files may already reside on the deployment WebFOCUS Reporting Server or subserver in applications configured to contain them. If you deploy these types of files, you may overwrite production files with the same name on the deployment server.

  Proprietary FOCUS data sources (.FOC) files cannot be deployed. If you try to drag and drop an invalid file type to the Reporting Server, an error message appears.

- For files containing the command -HTMLFORM `filename`:
  - If you are deploying a procedure that contains -HTMLFORM `filename` (including procedures created with the HTML Composer), also deploy the referenced HTML file to the WebFOCUS Reporting Server.
    
    In this scenario, the procedure must be launched in order to execute the report.

  - When you use the HTML Composer to create reports with parameters, you must deploy the referenced HTML file containing the launch form to both the Web server and the WebFOCUS Reporting Server.
    
    In this scenario, the HTML file must be launched from the Web server.
Working in a Default Partition Environment

How to:
Modify Properties of Partitioned Files

In this example, the developer selected the Test Environment as the default partition environment when creating the deployment scenario.

Developer Studio automatically partitions the .jpg and .htm files to the Web server, and the .fex files to the WebFOCUS Reporting Server.

Files in the Name display field that have a check \( \checkmark \) icon next to them have been partitioned.
Procedure: How to Modify Properties of Partitioned Files

1. In the Scenario Editor, right-click a file that has been assigned to a server in the Available Servers window and select Properties.

2. Click the Advanced tab to see the following two options:
   - Always deploy this file.
   - Never deploy this file.

3. Choose one of the options. They are unchecked by default.

   Always deploy this file. Use this option when Smart Deploy is in effect to designate specific files to be deployed even though they have not changed since the last deployment. You can use this option for a file that has not changed but references another file that has been assigned to a different server within the scenario.

   For example, if Smart Deploy is enabled and a WebFOCUS Maintain procedure with forms has not changed since the last deployment, by default it would not be redeployed. But if a new Web server has been assigned for a Web-based file that is used by one of the Maintain forms, the Maintain file (.mnt) with the form needs to be updated with the new URLs. In this case, you would select this option for the .mnt file, so that the form would be generated with the new Web server information.

   The following are examples of other situations when you might want to use this option:
   - When script files are embedded in Maintain forms rather than linked, and changes are made to the script code but not to the Maintain procedure and forms.
When CALLeD or EXECed procedures are assigned to different WebFOCUS servers since the last deployment, but the calling procedure has not changed.

**Never deploy this file.** Use this option when you do not want to deploy a file but information about the assigned server location is needed by other files in the application.

In WebFOCUS Maintain applications, this option is useful when Web-based files already reside on the target Web server and you do not need to edit files there or do not have write access. The correct URLs will be used in the forms, without writing the Web resource files to the Web server.

If you are also using Smart Deploy, you would need to assign the *Always deploy files* option to any .mnt files that have not changed but have any associated forms that need to reference the new Web server location.

### Setting Running Paths

**How to:**

Set Running Paths

You can set the execution search path for a deployed application on a Reporting Server. The Reporting Server will use the path specified to locate the resources for that application.

This feature is useful if you set up common resources shared by multiple applications. It allows you to point a server to a common location to find the resources it needs to run a specific application.

**Procedure: How to Set Running Paths**

1. In the Scenario Editor, expand a WebFOCUS environment under Available Servers in the right pane. Then select and right-click a WebFOCUS Environment.
Select Paths from the shortcut menu. The Set Running Paths for Server dialog box opens.

![Set Running Paths for Server dialog box]

2. The Available Paths list box displays the application directories you can add to the execution search path for the current application.

   Select a directory or directories and click Add, or click Add All to include all directories in the execution search path.

   You can also use the Remove and Remove All buttons to make the desired assignments.

3. In the Additional Path field, optionally enter another path to add to the execution search path.

4. Click OK to close the dialog box.

Step 4: Deploy the Project Files

How to:
Deploy Project Files in the Explorer Using the Shortcut Menu

Reference:
Scenario Editor
Deploying Application Dialog Box

Developer Studio automates the process of moving the files to the target servers.
In addition to deploying files as described in this topic, you can deploy them in the Scenario Editor. See *Step 3: Partition the Project Files* on page 249. The buttons on the toolbar in the Scenario Editor are also available in the Explorer.

**Procedure:** How to Deploy Project Files in the Explorer Using the Shortcut Menu

1. Select the project. From the drop-down list on the toolbar, select the deployment scenario.

2. Right-click the project. Select *Deploy* from the shortcut menu. Then select:
   - *Deploy* to move the files to the servers based on the partitioning defined in the deployment scenario.
   - *Deploy and Run* to move the files to the servers and then run the starting object defined in the deployment scenario.

3. During deployment, a dialog box tracks the status. You can click *Cancel* to terminate the process.
The following image shows the Deployment Progress window detailing successful deployment of HTML files to the Web server and procedures to the WebFOCUS Reporting Server.

**Tip:** You can click the Details button if you do not want to view messages during the deployment process.

After deployment, the ACTION field indicates that one of the following occurred:

- Deployment Status: Succeeded, 0 error(s), 0 warning(s).
- Deployment Status: Failed to complete. (The reasons for failure are shown, such as unresolved files, etc.)
- Deployment Status: Cancelled by user.

During the deployment process, the File field shows the names of objects as they are deployed to a server.

The progress bar indicates how much of the process is complete, as it takes place.

The Status field displays the events that took place in the deployment process.
Reference: Scenario Editor

The following image is the Scenario Editor.

**Application Files**

Displays the files in the selected project and their properties.

The value of Assigned is No if the file was not assigned to a server for deployment. Assigned is Yes if you assigned the file to a server for deployment.

**Available Servers**

Displays the WebFOCUS environments to which you can deploy files.
Step 4: Deploy the Project Files

**Reference:** Deploying Application Dialog Box

The following image is the Deploying Application dialog box.

![Deploying Application Dialog Box](image)

**Tip:** You can click the *Cancel* button during the deployment process. When deployment ends, the Cancel button becomes a Close button.

- After deployment the ACTION field indicates that one of the following occurred:
  - Deployment Status: Succeeded, 0 warning(s), 0 error message(s).
  - Deployment Status: Failed to complete. (The reasons for failure are shown, such as unresolved files, etc.)
  - Deployment Status: Cancelled by user.
- During the deployment process, the File field shows the names of objects as they are deployed to the server.
The progress bar indicates how much of the process is complete, as it takes place.
The Status field displays the events that took place in the deployment process.

**Configuring the Target Servers (Optional)**

**How to:**
- Set the Web Server Deployment Path Using the WebFOCUS Administration Console (Windows)
- Set the WebFOCUS Reporting Server Deployment Path Using the Reporting Server Console (Windows)
- Set the Web Server and WebFOCUS Reporting Server Deployment Path Using a Text Editor (Windows and UNIX)
- Set the Deployment Path (z/OS)
- Set the Deployment Path (AS/400)
- Create an Alias on the Web Server (Windows 2000)
- Restrict Authorization to Deploy Files Using the WebFOCUS Administration Console (Windows)
- Restrict Authorization to Deploy Files Using a Text Editor (Windows and UNIX)

**Reference:**
- APPROOT Variable for Setting the Deployment Path
- APPROOT Alias
- DEVELOPERS Variable for Restricting Authorization

During installation, Developer Studio and WebFOCUS set certain variables that affect the implementation of the deployment feature. You have the option of changing the values for these variables to meet site-specific needs. Most sites use the default values supplied during installation.

You can optionally specify the path in which files will reside on the Web server and the WebFOCUS Reporting Server. The path is defined by the APPROOT variable, which you can set in two different files, one for the Web server and one for the Reporting Server.

You also have the option of restricting the users authorized to deploy files. You can set the DEVELOPERS variable to restrict authorization.

Follow the instructions in this topic to review or change the settings. You can use the WebFOCUS Administration Console and the Reporting Server Console to edit the files that contain the variables, or you can edit them manually.
You need read and write access to the Web server and WebFOCUS Reporting Server in order to deploy files.

**Note:** The Application Root (APPROOT) directory of the WebFOCUS Client and Reporting Server do not need to point to the same physical location and you can install them on different servers or platforms.

**Reference:**  **APPROOT Variable for Setting the Deployment Path**

The `cgivars.wfs` file contains the APPROOT variable for the path in which deployed files will reside on the Web server. Typically these files include HTML pages, graphic images, Cascading Style Sheets, and JavaScript files.

The `cgivars.wfs` file is located in:

<table>
<thead>
<tr>
<th>Type of Installation</th>
<th>Location of cgivars.wfs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Studio with localhost</td>
<td><code>install_drive:\ibi\srv76\wfs\etc</code></td>
</tr>
<tr>
<td>WebFOCUS</td>
<td><code>install_drive:\ibi\client76\wfc\etc</code></td>
</tr>
</tbody>
</table>

The `edaserve.cfg` file contains the APPROOT variable for the path in which deployed files will reside on the WebFOCUS Reporting Server. Typically these files include procedures, WebFOCUS StyleSheets, and customized HTML pages that require processing on the Reporting Server.

The `edaserve.cfg` file is located in:

<table>
<thead>
<tr>
<th>Type of Installation</th>
<th>Location of edaserve.cfg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Studio with localhost</td>
<td><code>install_drive:\ibi\srv76\wfs\bin</code></td>
</tr>
<tr>
<td>WebFOCUS</td>
<td><code>install_drive:\ibi\srv76\wfs\bin</code></td>
</tr>
</tbody>
</table>

**Procedure:**  **How to Set the Web Server Deployment Path Using the WebFOCUS Administration Console (Windows)**

1. Access the WebFOCUS Administration Console from the Start menu, choose *Programs*, *WebFOCUS*, and *WebFOCUS Administration Console*.

2. On the WebFOCUS Administration Console, click *Configuration* from the left pane and under *Client Settings*, click *General*.

3. Locate the APPROOT variable in `cgivars.wfs`. Change the setting to a directory that exists on the Web server.
Ensure that this directory is also configured as a virtual directory on the Web server, to allow users access to the application. For details, see APPROOT Alias on page 265 and How to Create an Alias on the Web Server (Windows 2000) on page 266.

When specifying the setting, use forward slashes (/) to separate directories, as shown in the default setting:

```
APPROOT=install_drive:/ibi/apps
```

An example is:

```
APPROOT=d:/ibi/apps
```

During deployment Developer Studio creates a new subdirectory on the Web server under APPROOT. For example, d:\ibi\apps\Sales.

You can add one or more subfolders to the Web server deployment path. For details, see Step 3: Partition the Project Files on page 249.

4. Click Save and exit the WebFOCUS Administration Console.

**Note:** You must have Administrator privileges in order to modify WebFOCUS Client configuration privileges.

---

**Procedure:** How to Set the WebFOCUS Reporting Server Deployment Path Using the Reporting Server Console (Windows)

1. Access the Reporting Server Console:
   - From the Start menu, choose Programs, WebFOCUS Server, and Web Console.
   - or
   - In Developer Studio, expand the applicable environment under WebFOCUS Environments. Expand Data Servers. Select the remote WebFOCUS Reporting Server.
     
     Click the WebFOCUS Reporting server console icon on the toolbar.

2. Select Workspace from the left pane, then select Configure.

3. On the Workspace Configuration window, click the Edit edaserve.cfg button in the upper part of the right pane.

4. Locate the APPROOT variable. Change the setting to a directory that exists on the WebFOCUS Reporting Server.

   During deployment Developer Studio creates a new subdirectory on the WebFOCUS Reporting Server under APPROOT. For example, c:\ibi\apps\Sales.

5. Click Save and Restart and exit the Reporting Server Console.
**Note:** You must have Administrator privileges in order to modify WebFOCUS Reporting Server configuration privileges.

**Procedure:** How to Set the Web Server and WebFOCUS Reporting Server Deployment Path Using a Text Editor (Windows and UNIX)

1. Open the cgivars.wfs file in a text editor such as Windows Notepad.

2. Locate the APPROOT variable. Change the setting to a directory that exists on the Web server.

   Ensure that this directory is also configured as a virtual directory on the Web server, to allow users access to the application. For details, see APPROOT Alias on page 265 and How to Create an Alias on the Web Server (Windows 2000) on page 266.

   When specifying the setting, use forward slashes (/) to separate directories, as shown in the default setting:

   ```
   APPROOT=install_drive:/ibi/apps
   ```

   An example is:

   ```
   APPROOT=d:/ibi/apps
   ```

   During deployment Developer Studio creates a new subdirectory on the Web server under APPROOT. For example, d:\ibi\apps\Sales.

   You can add one or more subfolders to the Web server deployment path. For details, see Step 3: Partition the Project Files on page 249.

3. Save the changes and close cgivars.wfs.

4. Open the edaserve.cfg file in the text editor.

5. Locate the APPROOT variable. Change the setting to a directory that exists on the WebFOCUS Reporting Server.

   During deployment Developer Studio creates a new subdirectory on the WebFOCUS Reporting Server under APPROOT. For example, c:\ibi\apps\Sales.

6. Save the changes and close edaserve.cfg.

**Procedure:** How to Set the Deployment Path (z/OS)

To configure the WebFOCUS Reporting Server, set the APPSNS (application name space) parameter to YES in the EDACFGF configuration routine. In the qualif.INSTALL.DATA(FFSSERV) server configuration file (service block EDAAPPS), the APPROOT variable will be set to the qualif.APP used to install and configure the server.
For example:

```
APPROOT=EDAYHC.V5R2M01.APP
```

For more information on the APPSNS parameter, see the *Server Installation, Configuration and Operations for MVS* manual.

If you are using WebFOCUS Reporting Server for z/OS for deployment, the base application files are allocated during configuration (APPSNS=YES).

**Example:**  **Allocating Base Application Files (z/OS)**

The following is an example of base files allocated for the application:

```
EDAYHC.V5R2M01.APP.BASEAPP.ACCESS.DATA
EDAYHC.V5R2M01.APP.BASEAPP.FOCEXEC.DATA
EDAYHC.V5R2M01.APP.BASEAPP.FOCSTYLE.DATA
EDAYHC.V5R2M01.APP.BASEAPP.GIF.DATA
EDAYHC.V5R2M01.APP.BASEAPP.HTML.DATA
EDAYHC.V5R2M01.APP.BASEAPP.MASTER.DATA
```

**Procedure:**  **How to Set the Deployment Path (AS/400)**

1. In the CGI_BIN/EXPORT(WEBEXPORT) directory, set the APPROOT variable to the APPNAME defined during installation of the WebFOCUS Reporting Server for AS/400. This value is the same one defined for APPROOT in the xxxBIN/CONFIG(EDASERVE) file.

2. In the Web server’s HTTP configuration file, add the following lines for APPROOT:

```
Pass /APPROOT/* /IBI/APPS/*
Pass /approot/* /ibi/apps/*
```

**Reference:**  **APPROOT Alias**

After you set the APPROOT variable, you may need to create an alias for it. The alias allows browsing of the application directory and read permission.

The following will help you determine if you need to create an alias:

- If you are using Internet Information Server (IIS), an alias is automatically created for you. During installation of Developer Studio or WebFOCUS, you can choose automatic configuration of IIS.

- The APPROOT alias must correspond to the APPROOT path in cgivars.wfs.
**Procedure:** How to Create an Alias on the Web Server (Windows 2000)

1. From the Start menu, select **Settings** to open the Control Panel.
2. Choose **Administrative Tools**.
3. Choose **Internet Services Manager**. This selection opens a tool that allows you to manage the IIS (the Web server).
4. In the Internet Information Services window, locate and expand the **Default Web Site** node.
   
   Right-click the **Default Web Site** node and choose **New**, followed by **Virtual Directory**.
5. Follow the instructions of the Virtual Directory Creation Wizard to create a new virtual directory for APPROOT on the selected Web site.

**Note:** The Windows Challenge/Response is supported.

**Reference:** DEVELOPERS Variable for Restricting Authorization

The **ibiweb.cfg** file contains the DEVELOPERS variable, which identifies users with authorization to deploy files. By default, all users can deploy files:

**DEVELOPERS=**

The ibiweb.cfg file is located in:

<table>
<thead>
<tr>
<th>Type of Installation</th>
<th>Location of ibiweb.cfg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Studio with localhost</td>
<td><code>install_drive:\ibi\srv76\wfs\web\cgi</code></td>
</tr>
<tr>
<td>WebFOCUS</td>
<td><code>install_drive:\ibi\client76\wfc\web\cgi</code></td>
</tr>
</tbody>
</table>

**Procedure:** How to Restrict Authorization to Deploy Files Using the WebFOCUS Administration Console (Windows)

1. Access the WebFOCUS Administration Console from the Start menu, choose **Programs**, **WebFOCUS**, and then **WebFOCUS Administration Console**.
2. The WebFOCUS Administration Console opens. Select **Configuration** from the options on the left.
3. Select **Startup Parameters**.
4. On the ibiweb.cfg configuration window, select the DEVELOPERS variable and click **Modify**.
5. In the Usage field on the next window, enter the new value for the variable.
For an individual user, supply the user's logon ID. The user must have write access. For example:

```
DEVELOPERS=admin
```

For multiple users, separate IDs with a semicolon. For example:

```
DEVELOPERS=admin;power_user1;power_user2
```

6. Click Save and exit the WebFOCUS Administration Console.

**Procedure:** How to Restrict Authorization to Deploy Files Using a Text Editor (Windows and UNIX)

1. Open the ibiweb.cfg file in a text editor such as Windows Notepad.

2. Locate the DEVELOPERS variable.

   For an individual user, supply the user's logon ID. The user must have write access. For example:

   ```
   DEVELOPERS=admin
   ```

   For multiple users, separate IDs with a semicolon. For example:

   ```
   DEVELOPERS=admin;power_user1;power_user2
   ```

3. Save and close ibiweb.cfg.
Configuring the Target Servers (Optional)
Editing Application Components as Text in Developer Studio

Developer Studio provides a fully integrated text editor that you can use to create, view, and edit the source code for procedures, procedure components, Master and Access files, and other types of files required by your applications. The text editor enables you to use familiar Windows editing techniques, such as cut, copy, paste, undo/redo, and drag-and-drop.

In addition, you can:

- Take advantage of color-coded syntax designed to make writing, editing, and debugging procedures easier.
- Bookmark lines of a file for quick editing and easy reference.
- Find and replace text.
- Run procedures or procedure components directly from the Editor.

Within applications, you can open and edit multiple procedure components in separate Editor windows. You can also open a supplementary text editing window called the Other component in order to type fragments of code (such as Dialogue Manager syntax) or text (such as comments) that you may wish to incorporate into a procedure.

Topics:
- Text Editor
- The Other Component
- The Comment Component
- Creating a Text File or a Procedure Component as Text
- Opening Application Components as Text
- Finding and Replacing Text
- Changing Text Color and Case Size
- Adding Headings and Footings
- Using Bookmarks to Move Within a File
- Running a Procedure From the Editor
- Opening a Graphical Tool From the Text Editor
Text Editor

Since the text editor is fully integrated with the graphical toolset, changes you make to source code are immediately reflected in the graphical tools, and vice versa.

The text editor window includes:

- A menu bar, which provides access to all options available in the text editor.
- The editor window, which contains the open file, displaying color-coded text.
- The editor toolbar, which contains icons for frequently performed tasks such as saving and printing; cutting, copying, and pasting; undoing and redoing; finding text; toggling bookmarks, finding bookmarks, and clearing bookmarks; and accessing online help.
- When you are editing a procedure, the following additional icons appear: Report, Graph, Define, Join, Set, Allocation, Use. These icons enable you to launch graphical tools from the Text Editor. For details, see Opening a Graphical Tool From the Text Editor on page 286.

- A status bar that shows a help message, line and column positions, and the status of four functions—insert, caps lock, num lock, and scroll lock.

When you open a Master File, procedure, or HTML file in the Editor window, syntax elements in the text appear color-coded for easy viewing and editing. For example, the commands in a procedure appear in red. You can change default colors or remove text coloring to suit your preferences. For details, see Changing Text Color and Case Size on page 280.

Depending on where you place the cursor in the editor window, you can click the Open FOCUS tool button on the text editor toolbar or right-click and select Open tool whenever you want to edit code in a graphical tool. See Opening a Graphical Tool From the Text Editor on page 286 for details.

You can access any text editor that is registered with your operating system based on the program that is set up to be the default program that opens the selected file type. For example, in Windows Explorer, select a .fex file, then right-click and select Properties from the context menu. Click Change and select Notepad as the default program to open this file type if it is not already associated with an application. The option to open procedures (.fex files) with Notepad will now be available through Developer Studio.

This option is valuable for developers who want to take advantage of the WebFOCUS language and use the editor of their choice.
Accessing Text Editors

How to: Set Up the Default Action for the Open Option

You can access any text editor by right-clicking a file within the Developer Studio Explorer. A context menu for the file appears with the following default options:

- **Open in (Task Viewer/Text Editor/Registered Program)** to edit the procedure's components. Depending on this setting, the following options order will change.
  - **Open in (Task Viewer).** Depending on the product settings, this option may show **Open in (Text Editor)** or **Open in (Registered Program)**, for example, Notepad.
  - **Open in Text Editor** to edit the procedure's code using the Developer Studio text editor.
  - **Open in (Registered Program)** to edit the procedure's code using an external editor, such as Notepad.
  - **Open in auto detected tool** to open the procedure in the tool used to create it (for example, the Report Painter, HTML Composer, and so on). You must use this option if you need to open the file in the HTML Composer because it is the only way to open procedures created in the tool used to create them. You cannot open procedures created with those tools from the Task Viewer.

**Procedure:** How to Set Up the Default Action for the Open Option

1. In the Explorer window, select Options from the Window menu. The Developer Studio Options dialog box opens at the General tab.

2. From the Default file editor drop-down list, select one of the following:
   - **Edit in DevStudio tool** to edit a procedure in the Developer Studio Task Viewer.
   - **Edit in Text Editor** to edit a procedure in Developer Studio's text editor.
   - **Edit in Windows registered tool** to edit a procedure in Notepad, Wordpad, or other program as registered with the operating system.
The Other Component

How to: Use the Other Component

The Other component uses the text editor as a scratch pad on which you can create and edit a procedure or a component of a procedure. You can access the Other component from the following icon on the component toolbar.

The Other component is particularly useful when you want to add code that is not represented by a graphical tool—for example, Dialogue Manager code for managing the flow of control within a procedure. For information on Dialogue Manager, see Managing Flow of Control in an Application in the Developing Reporting Applications manual.

If Developer Studio recognizes the code you type as a report or graph request or a Define or Join command, the next time the procedure opens, the appropriate component type (for example, Report, Graph, Define, or Join) displays in the Procedures window. Similarly, if you type text that is preceded by the characters -*, which denote a comment, when you update the FOCEXC, the text appears as a Comment component (see How to Create a Comment Component on page 273). However, if you type Dialogue Manager code, it appears in a component called Other. For related information, see Creating a Reporting Procedure on page 111.

Although you cannot create and save complete text files in the Other component, this facility gives you the same text editing options as the Editor. You can also cut or copy text to the clipboard, then paste it into a text file in the Editor window or into another Other window.

Procedure: How to Use the Other Component

1. Right-click the procedure in the Procedures folder and choose Open from the shortcut menu. The Procedure window opens.

2. Click and hold a component connector (yellow diamond), then drag the arrow pointer to the Other button on the component toolbar.

   The text editor opens.

3. Type your code or other text. When you close the tool, your text is saved as a procedure component.
The Comment Component

When you create and open a WebFOCUS graphical procedure, a component called Comment appears by default in the Procedure window. It contains the name you assigned to the procedure.

The type of component and the information contained in it are previewed below the -* icon. The characters -* are required to identify text as a comment.

By default, the Comment component includes a comment containing the name of the procedure. This comment is not necessary for your application, and if you wish, you can delete it. You can also expand it by typing additional comments, introducing each new line with the comment characters -*.

**Procedure: How to Create a Comment Component**

In the Procedure window:

1. Click and hold a component connector (yellow diamond), then drag the arrow pointer to the Other button on the component toolbar. The text editor opens.

2. Type your comment, beginning with the characters -*. Each line of the comment must begin with -*.
3. Close the editor and update the procedure when prompted. The text is added to the procedure as a new Comment component.

**Adding and Removing Comments in the Text Editor**

A comment option is available in the text editor, which enables you to add and remove comments for selected lines of WebFOCUS code. You may use the text editor to create, view, and edit the source code for procedures.

The comment option gives you the ability to comment out an entire block of code, without manually typing in a dash/asterisk on every line.

**Procedure: How to Add/Remove Comments in the Text Editor**

The option to add/remove comments is available from the stand-alone Text Editor and the Text View of the Procedure Viewer.

**Note:** The comment option is not available while viewing the Source code from the Report Painter.

1. To open the procedure in a Text Editor:
   a. In the Explorer view, right-click a Procedure file (.fex) and select *Edit in Text Editor.*
b. In the Procedure Viewer, right-click a procedure component (for example, Report, Graph, Other), and select *Edit Text.*

![Procedure Viewer screenshot](image)

c. In the Explorer view, select *Text Editor* from the Command menu.

![Explorer view screenshot](image)

Select a procedure from the Open dialog box, or create a new procedure file (.fex).

The Text Editor opens.

2. Select the portion of WebFOCUS code that you would like to comment and select *Add Comment* from the Edit menu.
Creating a Text File or a Procedure Component as Text

**Note:** You may apply comments to a block of code or to a single line.

![Developer Studio text editor](image)

A dash/asterisk(-*) is added to the selected block of code.

```sql
BY @ProductCategory, (OR(<CD Players, CD Players>, <Camcorders, Camcorders>, <Camera, Camera>)) . Select Product Type.
```

**Note:** Comments (-*) are added to the beginning of the selected lines.

3. To remove the comment, select the commented lines of code and select Remove Comment from the Edit menu.

The comment (-*) is removed from the beginning of the selected lines of code.

**Tip:** You may also select the Add and Remove Comment selections from the right-click menu and by using the shortcut keys. Ctrl+M adds a comment. Ctrl+R removes a comment.

---

### Creating a Text File or a Procedure Component as Text

**How to:**

- Create a Text File
- Create a Text Component in a Procedure

You can use the Developer Studio text editor to create a text file from scratch using any language. If the file you create contains an executable procedure, you can run it directly from the text editor. See *Running a Procedure From the Editor* on page 285.

You also can create a text component in a Procedure using the Other component.
Procedure: **How to Create a Text File**

1. Choose TED (text editor) from the Command menu.

2. In the Open dialog box, enter a new file name in the File name box. For a file other than a procedure (.FEX) or a Master File (.MAS) include an extension.

3. Choose a file type from the File of type list. For a procedure (.FEX) or a Master File (.MAS), the file extension is picked up from this selection.

4. Click the Open button. The text editor opens.

   Type your text and close the text editor. When prompted, save your file.

   The new file is saved in the selected folder with the appropriate extension.

Procedure: **How to Create a Text Component in a Procedure**

1. Create a new procedure:
   
   - With the Procedures folder highlighted, select New from the File menu.
   
   or

   - Right-click the Procedures folder and select New from the pop-up menu, then select Procedure.
The Add Procedure dialog box opens.

2. Enter a name for the new procedure in the File name field.
3. Select Procedure Viewer from the Create with drop-down list.

The Component Connector toolbox opens.

4. Click and hold a component connector (yellow diamond), then drag the arrow pointer to the Other button on the component toolbar. The text editor opens.
5. Type your text and close the Other component. When prompted to update the procedure (FOCEXEC), click Yes.
6. Close the procedure.
Opening Application Components as Text

### How to:
- View an Application Component as Text
- View a Procedure Component as Text

You can view or edit one or more procedures or other text files in the text editor. If you open multiple editor windows at the same time, you can move between them as you work. For example, you can drag-and-drop text between open windows. You can open existing files from the TED (text editing) option in the Command menu or from a project folder.

#### Procedure: How to View an Application Component as Text
You can use any of the following three methods to expose the underlying code or text for the selected file in the editor.

1. Right-click a file, such as a procedure or Master File, in an Explorer folder.
2. Select Edit As Text from the shortcut menu.
3. Highlight a file, such as a procedure or Master File, in an Explorer folder.
4. Select Edit as Text from the File menu.
5. Choose TED from the Command menu. The Open dialog box appears.
6. Specify the type of file you want to view or edit. FOCEXEC is the default.
7. Select a file, then click Open.

#### Procedure: How to View a Procedure Component as Text
In addition to viewing the code for an entire procedure from a folder, you can examine and edit the code for an individual procedure component.

1. Right-click a procedure folder in the Explorer and select Open from the shortcut menu.
2. Right-click a procedure component (for example, Report, Graph, Other), then select Edit As Text.

You will see the underlying code for the selected procedure component in an Editor window.

### Finding and Replacing Text
The Find dialog box enables you to search a file for specified text or numbers. You can also include special characters—for example, paragraph marks and symbols—in your search criteria. Replace enables you to search for a word and replace it with another.
Options available to narrow or speed your search are:

- **Match Case.** Finds only those occurrences with the exact combination of uppercase and lowercase letters specified in the Find What box.

- **Regular Expression.** Searches for text using the wildcard characters "*" (matches any number of characters) and "?" (matches any single character).

- **Wrap around search.** Searches the entire file from the current insertion point.

- **Direction (to search).** "Up" searches from the insertion point to the beginning of the document. "Down" searches from the insertion point to the end of the document.

- **Mark All.** Marks all of the lines with instances of your search text with a bookmark. For more information see *Using Bookmarks to Move Within a File* on page 283.

## Changing Text Color and Case Size

**How to:**

- Change the Foreground and Background Text Color
- Change the Font
- Change Text to Use Uppercase Letters
- Change Text to Use Lowercase Letters

You can change the text editor fonts, default colors and case size in a file. This is referred to as syntax coloring and is used in three file types: Master File (.MAS), procedure (.FEX) and HTML (.HTM).

In addition, you can apply text color to uppercase or lowercase text using the case-sensitive option.

<table>
<thead>
<tr>
<th>Syntax Coloring Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Regular or default text in the Text Editor.</td>
</tr>
<tr>
<td>Text Selection</td>
<td>Text that has been highlighted in the Text Editor.</td>
</tr>
<tr>
<td>Number</td>
<td>Numbers in the text editor.</td>
</tr>
<tr>
<td>Operator</td>
<td>Operators in the text editor.</td>
</tr>
<tr>
<td>Comment</td>
<td>Any text that follows a hyphen (-) and an asterisk (*). For Master Files (.MAS), any text that follows a dollar sign ($).</td>
</tr>
</tbody>
</table>
### Syntax Coloring Option

<table>
<thead>
<tr>
<th>Syntax Coloring Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword</td>
<td>Words in the FOCUS language.</td>
</tr>
<tr>
<td>End of Line</td>
<td>Text designating the last line of syntax.</td>
</tr>
<tr>
<td>Double Quoted String</td>
<td>Any text between double quotation marks.</td>
</tr>
<tr>
<td>Single Quoted String</td>
<td>Any text between single quotation marks.</td>
</tr>
<tr>
<td>Quotes</td>
<td>Any text between a single or double quotation marks.</td>
</tr>
<tr>
<td>Default Text</td>
<td>All other text.</td>
</tr>
<tr>
<td>Background</td>
<td>The space that all text displays on.</td>
</tr>
</tbody>
</table>

**Procedure: How to Change the Foreground and Background Text Color**

1. With the text editor open, select Options from the View menu. The Editor Options dialog box opens.
2. Click Font/Color. The Font and Color Settings dialog box opens.
3. Under the Color group box, highlight the syntax coloring option you want to change.
4. Deselect the Automatic check box next to the Foreground and Background drop-down lists. The drop-down lists become active.
5. Select a Foreground color and Background color from the respective drop-down lists.
6. Click OK.
7. Click OK to close the Font and Color Settings dialog box.
8. Click OK to close the Editor Options dialog box.

**Procedure: How to Change the Font**

1. With the text editor open, select Options from the View menu. The Editor Options dialog box opens.
2. Click Font/Color. The Font and Color Settings dialog box opens.
3. Click Choose Font. The Font dialog box opens.
4. Select a font and type from the Font drop-down list and a size from the Size drop-down list.

5. Click OK.

6. Click OK to close the Font and Color Settings dialog box.

7. Click OK to close the Editor Options dialog box.

**Procedure: How to Change Text to Use Uppercase Letters**

1. In the text editor, highlight the text you want to change.

2. Do one of the following:
   - Select *Upper Case* from the Edit menu.
   - or
   - Click the AB button on the Text Editor toolbar.

**Procedure: How to Change Text to Use Lowercase Letters**

1. In the text editor, highlight the text you want to change.

2. Do one of the following:
   - Select *Lower Case* from the Edit menu.
   - or
   - Click the Ab button on the Text Editor toolbar.

**Adding Headings and Footings**

**How to:**

Add Headings and Footings

You can add text that repeats at the top (heading) or bottom (footing) of every page in a file. You can also insert the file time (the time a file was changed) and system time (the current time) or page numbers in a file heading and/or footing.

These headings and footings settings apply to reports displayed on a printed page. They do not appear when the report is displayed in the Web browser.
Procedure: How to Add Headings and Footings

1. From the File menu, select Header and Footer. The Header and Footer dialog box displays.
2. Enter the heading text in the Header field or the footing text in the Footer field. You can do this by typing text you want included, or by clicking one of the buttons located on the right of the dialog box:
   - Insert Filename includes the name of the procedure.
   - Insert Page Number includes the page number.
   - Insert Number of Pages includes the total number of pages.
   - Insert Date includes the date.
   - Insert Time includes the time.
   - Default restores the Header and Footer fields to the default display.
3. Select a justification for the heading and footing from the Header and Footer section:
   - Left left-justifies the heading or footing.
   - Center centers the heading or footing.
   - Right right-justifies the heading or footing.
4. Click OK.

Using Bookmarks to Move Within a File

How to:
- Apply or Remove a Bookmark
- Jump to a Bookmark
- Apply Multiple Bookmarks
- Remove Multiple Bookmarks

The bookmark option places a bullet next to or removes a bullet from any line in a file. You can place multiple bookmarks in a file by using the Mark All function in the Find dialog box. Once a bookmark is added, you can jump to that bookmark from anywhere in a file. If you have multiple bookmarks, the cursor jumps to the very next bookmark.

Bookmarks are temporary; they disappear when you close the text editor.
**Procedure: How to Apply or Remove a Bookmark**

1. Position the cursor anywhere in a line that you want the bookmark to appear next to.

2. Click the Toggle Bookmark icon on the editor toolbar.

   or

   Position the cursor on the line to be marked, then press and hold down the Ctrl key and the F2 function key simultaneously.

**Procedure: How to Jump to a Bookmark**

To jump to another bookmark, click the Next Bookmark or Previous Bookmark icon on the editor toolbar or press the F2 function key.

The cursor jumps to the very next bookmark in that file.

**Procedure: How to Apply Multiple Bookmarks**

1. From the Edit menu, select Find.

   The Find dialog box opens.

2. In the Find What field, type the word or phrase you want to find, then click Mark All to mark all lines that contains this text.

**Procedure: How to Remove Multiple Bookmarks**

Click the Clear All Bookmarks icon on the editor toolbar or close the editor window and reopen it.
Running a Procedure From the Editor

**How to:**
- Run a Full Procedure as Code
- Run a Procedure Component as Code
- Run Several Components From the Text Editor

You can run a procedure from the text editor. You also can run executable procedure components (such as Defines and Joins) by placing the cursor within the specific section of code or by highlighting all code applying to each component. The results of a report or graph are displayed in the browser. For other types of components, results simply take effect (for example, virtual fields and fields from the joined data sources will be available for selection in the Fields windows).

**Procedure:**  **How to Run a Full Procedure as Code**

1. Open the procedure in the text editor.
2. Run the procedure using any of the following methods.
   - Select *Run* from the File menu.
   - Click the *Run* icon on the toolbar.
   - Press the F5 shortcut key.
3. The results appear in the appropriate viewer.

**Procedure:**  **How to Run a Procedure Component as Code**

1. Select the procedure, right-click it and select *Edit Text* from the pop-up menu.
2. Place the cursor within or highlight the specific text you wish to run.
3. Right-click and select *Run tool* (where the tool corresponds to the type of code you have selected, for example, Report, Define, Join, Set, and so on).

**Procedure:**  **How to Run Several Components From the Text Editor**

1. Open a procedure in the text editor or Other component.
2. Highlight the code to be run.
3. Right-click and select *Run*. 
Opening a Graphical Tool From the Text Editor

Since the text editor is fully integrated with the graphical toolset, you can open the following tools from the editor or the Other component: Set, Allocation, Use, Join, Define, Report, Graph. The location of the cursor in the procedure code determines where the code is returned in the editor window. When you close a tool that you opened from the Text editor, you return to the text editor. Any edits you saved in the graphical tool are reflected in the code.

Procedure: How to Add Code With a Graphical Tool From the Text Editor

In the text editor or the Other component:

1. Open a procedure or a procedure component.

2. Position the cursor on a blank line at the point where you want to insert the code generated by the tool you are about to open.
   
   **Tip:** If the cursor is not located on a blank line, the new code will be inserted before the block of contiguous code in which the cursor is positioned.

3. Click one of the following icons on the toolbar: Set, Allocation, Use, Join, Define, Report, Graph. The corresponding graphical tool opens.

4. Create the procedure component and close the tool.

5. When you are asked if you want to update the procedure, click Yes.

You return to the text editor, where the new code has been inserted.
Procedure: How to Edit Code With a Graphical Tool From the Text Editor

In the text editor or the Other component:

1. Open a procedure or a procedure component.

2. Position the cursor on the line containing the code you want to edit, and click the *Launch FOCUS Tool* icon on the toolbar.

   The appropriate graphical tool opens.

3. Edit the procedure component and close the tool.

4. When you are asked if you want to update the procedure, click *Yes*.

   You return to the text editor, where the code has been revised.
Creating an Update Application With Update Assist

Update Assist provides a simple way to create Web-based data source file browsers and data maintenance applications in just a few minutes without having to write code.

You can create applications with the Update Assist that add records, update records, or perform a combination of add, update, delete, and search functions against any data source for which you have read/write access. Data navigation and input validation are automatic. This means you get an update application with no need to design forms, or to write navigation, validation or update procedures.

To start using Update Assist, do the following:

1. Open a project.
2. Right-click any Master File in the project’s path and click Update Assist.
3. Follow the directions in the Update Assist dialog boxes.

The six Update Assist dialog boxes correspond to the six steps for generating Update Assist applications.

Topics:
- Update Assist (Step 1 of 6): Selecting Segments to Update
- Update Assist (Step 2 of 6): Selecting Fields to Update
- Update Assist (Step 3 of 6): Selecting Navigation Options
- Update Assist (Step 4 of 6): Selecting a Color Scheme
- Update Assist (Step 5 of 6): Selecting Output File Options
- Update Assist (Step 6 of 6): Confirming Selections
- About Your Update Assist Application
- Editing Your Update Assist Application
- Calling an Update Assist Procedure From a WebFOCUS Report
- Usage Notes
Update Assist (Step 1 of 6): Selecting Segments to Update

The first window of Update Assist shows you the structure of the master file and prompts you to select the segments you want to update.

This window contains the following fields/options:

1. **Select lowest segment or hierarchy you would like to auto update...**
   
   Contains a list of all the segments in the Master File you selected. If this is a flat file or a relational file, you will only see one segment.

   If this is a hierarchical file or a joined view, you will see more than one segment. Select the lowest segment containing fields that you want to update (the one farthest down in the list).

2. **Check the segment(s) to generate auto update logic...**

   Contains the selected segment, plus any segments that are above it. If this is a:
   
   - Flat file or a relational file, only one segment will appear. Select the segment.
   
   - Hierarchical file, there may be more than one segment. Select the segments that contain the fields you want to update.
Update Assist (Step 2 of 6): Selecting Fields to Update

In this section:
Change in When User-Supplied Data Is Validated

In the second window of Update Assist, you select the fields you want to update.

This window contains the following fields/options:

**Update Segment Options**

Contains a list of the segments you selected in the previous window. In order to enable changes to any of the fields in a segment, select the segment and turn on the Add, Update, Delete and Partial Key options.

If you want to add a Search button to your application, turn on Partial Key. For more information, see *How to Enable the Search Button in Your Application* on page 294.
Field View Options

Contains the fields in the segment that you selected in the Update Segment Options section. Once you have enabled changes to the segment as a whole, you can set change options for each individual field in the segment.

Visible

Determines whether the field is visible to the user.

Changeable

Determines whether the user can change the field. This option is available only if Add or Update was selected in the Update Segment Options section.

Note: A key field cannot be changed.

Tip: You can select multiple fields and then click once to change the Visible or Changeable settings.

Validation

Applies a validation technique which verifies the value a user enters in the field. This option is available only if Changeable is set to Yes.

The options for Validation are:

Automatic, the default validation option, validates the user's entry against the field format defined in the Master File. This automatically supports validation for Alphanumeric, Numeric (including Floating Point and Integer), and Date formats. The validation is performed using client-side JavaScript and doesn't require the server to validate the data.

Range allows you to define a numeric range between which data is valid for the field. See How to Use a Range to Validate a Field on page 294 for details. This option is best used for numeric fields.

Static List allows you to supply a list of valid values from which the user selects at run time. When Static List is selected, the Field Validation - List dialog box opens. See How to Use a Static List to Validate a Field on page 295 for details.

Dynamic List allows you to supply a list of valid values for the field that are retrieved from a specified data source at run time. When Dynamic List is selected, you are prompted for the Master File and field from which to retrieve values. See How to Use a Dynamic List to Validate a Field on page 296 for details.

None does not perform a validation.

Required field specifies that the user must supply a value for the field.
Change in When User-Supplied Data Is Validated

In earlier releases, applications generated with Update Assist performed field-level validation of user entries any time a user tabbed or clicked their mouse to move from one field to another on the form. Currently, rather than performing validation as users enter data into each field, applications generated with Update Assist validate data only when the user submits the information, for example, by clicking Save if the form contains a Save button. If the form does not contain a Save button, data is submitted when the user clicks Next to go to the next step.

**Procedure: How to Rename a Segment or a Field**

You can easily rename a segment or field as it is displayed to the user (this is called the Display Name).

1. Right-click the segment or field.
2. Click Rename.
3. Type the new name and press Enter.

**Procedure: How to Re-sort Fields in the Segment & Field Options Window**

You can change the order of the fields as they appear in the window.

1. In the upper right corner of the Field View Options pane, click the pull-down arrow to the right of the the alphabetical sort icon.
2. To sort fields by:
   - Display name, select Display Name from the drop-down menu.
   - Original name, select Name from the drop-down menu.
The order in which they appear in the Master File, select Original order from the drop-down menu.

**Note:** This does not affect the order of the fields in your application. In the application, the fields are sorted according to their order in the Master File.

**Procedure: How to Enable the Search Button in Your Application**

You can use the Partial-Key option to place a Search button in your application. At run time, the user can click the Search button, enter a key value to select a record, and then display field values for that record.

1. Select the segment for which you want to enable Search.
2. Set Partial-Key to Yes.
   A Partial-Key View tab is added to the window.
3. Click the Partial-Key tab.
4. Select the fields that you want to appear in the table at run time.

**Tip:** If you entered Display Names in the Single Record View tab, you can click any entry in the Display Name column here to use those entries.

**Procedure: How to Use a Range to Validate a Field**

When you choose the Range option to validate a field, Update Assist opens the Field Validation - Range dialog box. You use this option with numeric fields to specify a range of values for any information the user enters.

1. Enter a From value to indicate the beginning of the acceptable range of values.
2. Enter a To value to indicate the end of the acceptable range of values.
3. Click OK.
Procedure: How to Use a Static List to Validate a Field

When you choose the Static List option to validate a field, the Field Validation - List dialog box opens. Use this option to specify a static list of values that the user can select from a pull-down list.

1. To enter new acceptable field values, click the Add New Item button, type the text for the value and press Enter.

2. To edit an existing value, select it, make any changes, and press Enter.

3. To delete an existing value, select it and click the Delete selected items button.

4. To change the order of the values, use the move item up in the list and the move item down in the list buttons.

5. When you are done, click OK.

Note: When populating a Static list, make sure you scan the data source for all possible values and enter them into the list. If you leave a value off the list that’s in a current record and that record is selected for update, the value for the bound column will change to the first item on the Static List.
Tip: If a field is not required and you want to give your user the option to leave it blank, put an empty entry in as the first item in your Static list.

Procedure: How to Use a Dynamic List to Validate a Field

When you choose the Dynamic List option to validate a field, you specify a field in the data source that contains the possible values. At run time, a list of values is retrieved from that data source and the user can then select one of these values from a pull-down list.

The real power of Dynamic Lists is that you can add items to the lists in your Update Assist applications without having to make changes to the application’s forms code. Static lists require you to edit the forms in your Maintain application using the Maintain Development Environment. For example, if you choose to use a flat file as the source of items in your lists, you can simply add items to the flat file or export a new flat file from your data source to change the list; you do not need to change a line of application code.

1. In the Open dialog box, select a Master File for the data source containing the values for the field and click OK. You can use any data source type supported by WebFOCUS.

2. The Field Validation - File dialog box opens. Select the name of the field in the data source that contains the values you want to validate against and click OK. (If you want to select a different data source, click Browse.)
Update Assist (Step 3 of 6): Selecting Navigation Options

The third window of Update Assist is where you determine what the user interface for your Update Assist application will look like.

This window contains the following fields/options:

Do not generate any user interface for this project

Generates a project with no user interface.

Prompt user to enter database security information (DBA)

Generates a page prompting the user to enter a password to access the data in the data source. Use this option if data source security is enabled.

The application will store the password in a cookie, so the user will only be prompted for it once.
**Generate results page**

Displays a new window with the phrase "Your changes have been saved" when a user performs a Save or Delete action (which is what most users expect from a Web application).

This option is most appropriate when you select *Do not generate any user interface for this project*. The best use of the results page is in an application that calls the Maintain procedure directly using a URL and displays the Maintain form in a frame or iFrame of the application's workspace, as opposed to presenting a full navigation using the Tree, Combo, or Edit box. The results page might seem counterintuitive for users who are using Tree, Combo, or Tidbits navigation, as it takes the user out of the context of the edit forms and requires them to re-navigate to their information or click the browser's Back button to return to a form with controls.

**Key values selected via tree**

Generates a form in which the user selects records using a hierarchical tree control.

**Key values selected via combobox**

Generates a form in which the user selects records using a combo box.

**Key values entered by user**

Generates a form in which the user selects records by entering key values.

*Note:* This requires that the user knows the actual values for the key values.

**No key values required**

Generates a Maintain procedure to be called by another procedure (usually a WebFOCUS report) with the appropriate values to fill out the screen. To see how to create the WebFOCUS report that calls this type of Update Assist project, see *Calling an Update Assist Procedure From a WebFOCUS Report* on page 312.
Update Assist (Step 4 of 6): Selecting a Color Scheme

The fourth window of Update Assist is where you select the color scheme for your Update Assist application.

Select a color scheme and click Next. To see a preview of the color scheme and more information about it, select the particular color scheme and click Show details.

Tip: If you have detailed knowledge of WebFOCUS Maintain, are familiar with Cascading Style Sheets, and are able to write in XML, you may choose to create a customized color scheme and appearance for your application. For technical documentation on how to do so, contact your Information Builders representative.
Update Assist (Step 5 of 6): Selecting Output File Options

In the fifth window of Update Assist, the names of the files that Update Assist will generate are listed. You can rename these files if necessary.

This window contains the following fields/options:

**Output Filenames:**

Lists the files that will be created by Update Assist.

**Rename**

Allows you to change the selected file name.
Update Assist (Step 6 of 6): Confirming Selections

In the last window of Update Assist, you confirm all of your settings and generate the application files.

This window contains the following fields/options:

**Project summary**

Lists the files that Update Assist is creating and summarizes the operations the project can perform on the data source. To make changes, press the Back button until you reach the screen containing the information you want to change.

**Make this project use compiled Maintain procedures**

Select this option when you are finished debugging your project. Compiled procedures run more quickly, but it takes time to compile them.

**Automatically run this project when you press the 'Finish' button**

Deselect this option if you do not wish to run the project immediately, for instance, if you wanted to perform further editing on your project files.
About Your Update Assist Application

How to:

Run Your Update Assist Application
Use the Search Button

Reference:

Files Generated By Update Assist
Working With Empty or New Data Sources

Once you click Finish in the last Update Assist window, Update Assist generates the files needed for your application and deploys and runs the application, if specified. This section describes how to run your application from outside this window and about how Update Assist applications work.

Procedure: How to Run Your Update Assist Application

Once you have generated an Update Assist application, you can run it any time your WebFOCUS Server is running. To do this, navigate to the URL:

http://yourmachine/approot/appname/launchname.htm

where:

yourmachine
  Is the name of your machine on the network. If you are at the machine where the WebFOCUS Server is running, you can type localhost.

appname
  Is the name of your project.

launchname.htm
  Is the name of the HTML launch file for the Update Assist application. The default value for launchname is the name of your Master File.
**Reference: Files Generated By Update Assist**

Update Assist generates the following files:

<table>
<thead>
<tr>
<th>File name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MasterFileName.htm</td>
<td>Is the launching HTML file for your application and contains user interface information.</td>
</tr>
<tr>
<td>MasterFileName</td>
<td>Is the name of your Master File.</td>
</tr>
<tr>
<td>MasterFileName_navbar.htm</td>
<td>Contains more user interface information and determines the appearance of the Tree, Combo Box and Edit Box options. For more information on ways you can edit this file, see <em>Changing the Search Field for Tree and Combo Box Navigation</em> on page 305 and <em>Customizing the Tree Control</em> on page 307.</td>
</tr>
<tr>
<td>mntSignon.htm</td>
<td>Contains the window prompting you for your data source password (only if you selected <em>Prompt user to enter database security information (DBA)</em> in Step 3).</td>
</tr>
<tr>
<td>SegmentName.mnt</td>
<td>Contains the Maintain language code to display and modify the fields in that segment of the data source. You will see a Maintain procedure for each segment that you selected in Step 1.</td>
</tr>
<tr>
<td>SegmentName</td>
<td>Is the name of one of the segments in your Master File.</td>
</tr>
<tr>
<td>SegmentName_validationInit.js</td>
<td>Contains the JavaScript code to automatically validate user-entered values (only if you selected Automatic validation for any field). You will see a JavaScript file for each segment that you selected in Step 1.</td>
</tr>
</tbody>
</table>
### About Your Update Assist Application

<table>
<thead>
<tr>
<th>File name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>color_name.gif</td>
<td>(multiple) Are used in the user interface.</td>
</tr>
</tbody>
</table>

where:

- **color**
  - Is the color scheme you selected in Step 4.

- **name**
  - Is the user interface element.

- **listfill.fex**
  - Contains code to extract values for a dynamic list (for more information, see *How to Use a Dynamic List to Validate a Field* on page 296).

---

**Procedure:**  **How to Use the Search Button**

If you turn on Partial-Key for any segment in your Master File, a Search button is placed in your application.

The user can click this button and enter any key field value in the box. Use $ to represent any character; * to represent multiple characters.

For example, entering 1$* retrieves all records where the key field begins with a 1 and is at least 3 characters long.

**Reference:**  **Working With Empty or New Data Sources**

- **In tree navigation:** If you have selected the Add option for any data segments that do not contain data, the Tree will display a dot for the null segment. You can right-click on the dot to enter new data for that segment.

- **In combo box navigation:** For any data segments that do not contain data, the combo boxes display a [New] command. This option enables you to enter new records.
Editing Your Update Assist Application

When you click Finish in the Update Assist (Step 6 of 6) - Summary dialog box, the files that comprise your Update Assist project are generated (for a list of files, see *Files Generated By Update Assist* on page 303).

If you wish, you can perform further editing on these files.

Changing the Search Field for Tree and Combo Box Navigation

If you select either Tree or Combo Box navigation for your Update Assist project, by default the navigation displays the keys for each selected segment. This is a safe choice because keys are always unique. Tree and Combo Box navigation code automatically ensures that duplicates will be removed.

Not all users find keys friendly. In many cases, the record key is an integer or a unique alphanumeric code. When users are trying to locate a particular record, they seldom do so by scanning a list of key code values.

You can easily change the search field from the key field to a more friendly field by editing the segname navbar.htm file. However, if the new friendly field is not unique, you will only be able to select the first instance of each value. For example, suppose your database contains a list of movies, and there are two movies named *Airplane*.
There are two methods to solve this problem:

- Specify a folder for each friendly field that will contain duplicate titles, and show each unique key for the duplicate inside that folder.

- Create a DEFINE field that concatenates the friendlier field with the key field (which as you know is always unique). Use that as the display field for the tree or combo box.

**Procedure: How to Change the Search Field for Tree and Combo Box Navigation**

1. Open the file `segname_navbar.htm` in a text editor.
2. Find the `constructFields()` function at the bottom.
3. For the call to `fieldInfo.addField()`, change the sixth parameter (the displayfield) to the field name you want to use on display. Make sure to leave the quotes around the field name.

**Procedure: How to Generate Folders for Non-Unique Search Fields**

1. Open the file `segname_navbar.htm` in a text editor.
2. Find the `constructFields()` function at the bottom.
3. Above the existing call to `fieldinfo.addfield()`, add a call as follows:

   ```javascript
   fieldInfo.addField("friendlyfield",
   "friendlyfield",
   "segname",
   null,
   false,
   "friendlyfield",
   false,
   true,
   true,
   "friendlyfield",
   "segname",
   null);
   
   where:

   friendlyfield
   Is the name of your friendly field.

   segname
   Is the name of the segment in which the friendly field resides.
   ```
Procedure: How to Define a New Unique Field in a Master File

To use a DEFINE field as the displayed value:

1. Edit the Master File for the data source as text.
2. Create a DEFINE field by adding the following line to the end of the fields in the segment:

   ```
   DEFINE newfield/An=keyfield | friendlyfield
   ```

   where:

   - `newfield` is the name of your new field.
   - `An` is the format for your new field. \( n \) is the sum of the format for your key field and the format for your friendly field.
   - `keyfield` is the name of your unique key field
   - `friendlyfield` is the name of the friendly field

   **Tip:** You may wish to add some punctuation between the two fields, for example, a comma or dash. Do not forget to increase the size of your new field by the appropriate number of characters.

Example: Creating a Unique Field for the MOVIES Data Source

Adding the following code to the end of the MOVIES Master File defines a field that concatenates the TITLE field with the MOVIECODE field, which will appear in parentheses:

```
DEFINE TITLEKEY/A50 = TITLE | '(' || MOVIECODE || ')';$
```

Customizing the Tree Control

The Tree control option for Update Assist applications enables users to find the record they are looking for by navigating through a tree hierarchy. You can add the following JavaScript variables in the `MasterFileName_navbar.htm` file to control this hierarchy:

- `maxNodesPerFolder`: an integer value that controls the maximum number of nodes that can be displayed in each folder. If the number of nodes exceeds this number, the Tree control automatically splits the nodes up using subfolders labeled with a range of the node values contained within.
• **maxFolders**: an integer value that determines the maximum number of folders that can be rendered. The total number of nodes that will be rendered will be divided up amongst these folders.

These variables and the values you wish to set for them can be added to the area of code inside the *MasterFileName*_navbar.htm file that generates the tree, using the syntax

```javascript
maintainTree.maxNodesPerFolder =
```

and

```javascript
maintainTree.maxFolders =
```

prior to the `maintainTree.init()` code.

For example:

```javascript
var maintainTree = new maintainTreeInfo(constructFields(),mCallBack);
linestyle = "white";
maintainTree.maxNodesPerFolder= 8;
maintainTree.maxFolders = 20;
maintainTree.init();
```

**Note:** If the `maxNodesPerFolder` and `maxFolders` values are set in a way that would cause a conflict, the setting for `maxFolders` takes precedence over `maxNodesPerFolder`. 
Calendar Control for Date-Formatted Fields

A calendar icon appears next to changeable date-formatted fields. When a user clicks the calendar icon, a calendar appears; any date selected on this calendar is entered into the date field. Users can also enter dates into the date field manually.

Date-Stamping Fields

How to:
Date-Stamp a Field in an Update Assist Application

Many DBMSs allow you to create a "time stamp" field. This automatically fills the field with the current date and/or time and saves the user having to do it. There are many reasons at an application level for doing this; the most common is to give reporting applications some way to track when a record was first created or when each change was entered.

Note: If you are using an external DBMS that directly supports Date and Time Stamp field types, you won’t need to use this technique. Instead, make sure the field that contains the time stamp is set to Changeable = No to prevent Update Assist from even touching that field.
Procedure:  How to Date-Stamp a Field in an Update Assist Application

To date-stamp a field in an Update Assist application, so that when a user clicks New, your application can set the initial value of the field to today's date in the stack before it's displayed in the form:

1. Open the SegmentName.mnt file in the Maintain Development Environment.
2. Add this line of code to the top of the maintain, just above Case Top:

   ```java
   MODULE IMPORT(MNTUWS);
   ```

   This imports the library of functions shipped with WebFOCUS Maintain.

3. Scroll down to the newrecord case and add this code right below the first Stack Clear statement:

   ```java
   COMPUTE TheDate/MDY = Today();
   COMPUTE stack.datefield = TheDate;
   ```

   Where stack and datefield are the stack name and field name to which you want to assign the current date.

   **Note:** If you have multiple fields that need to be set to today's date, you only need to set the variable TheDate once and can re-use it as many times as you need.

Example:  Date-Stamping a Field in the MOVIES Data Source

If you wanted the Release Date field from the MOVIES data source to contain the current date, your code would look like this:

```java
COMPUTE TheDate/MDY = Today();
COMPUTE Movinfo_stack.RELDATE = TheDate;
```

Auto-numbering Fields in Update Assist Applications

**How to:**

Auto-Number a Field in an Update Assist Application

Some DBMSs allow you to create an "auto-number" field. This automatically fills the field with a sequence number that’s the last record’s index plus one. This saves the user having to make up an arbitrary key for the record, so it’s a popular feature.
Procedure: **How to Auto-Number a Field in an Update Assist Application**

To auto-number a field in an Update Assist application, so that when a user clicks New, your application can set the initial value of the field to the next sequence number in the stack before it’s displayed in the form:

1. Open the *SegmentName*.mnt file in the Maintain Development Environment.
2. Scroll down to the newrecord case and add this code right below the first Stack Clear statement:

   ```
   Stack clear *SegmentNameStk*;
   For all next MasterFileName.*SegmentName*.autonum into *SegmentNameStk*;
   NextVal/I5 = *SegmentNameStk*(*SegmentNameStk*.FOCCOUNT).val + 1;
   Stack clear *SegmentNameStk*;
   ```

**Note:** If you are using an external DBMS that directly supports Date Stamp field types, you won’t need to use this technique.

Continuing Displaying Currently Displayed Values After a New Action

**How to:**

Continue Displaying Currently Displayed Values After a New Action

By default, Update Assist clears all text boxes and controls in the form on a New action. You can have the values stay in the text boxes by editing the *SegmentName*.MNT file.

For example, users of some types of applications may be entering many similar records, one after another, and would like to display a record, then essentially have the New action display a copy of the record which they can tweak and edit before clicking Save.

Procedure: **How to Continue Displaying Currently Displayed Values After a New Action**

1. Open the *SegmentName*.MNT file and go to the newrecord case.
2. Comment out the line that clears the stack, using a double dollar sign.
Calling an Update Assist Procedure From a WebFOCUS Report

In this section:
- Calling an Update Assist Project From a WebFOCUS Report Example

How to:
- Call an Update Assist Maintain Procedure From a WebFOCUS Report

One way to use an Update Assist procedure is to call it from a WebFOCUS report. You can set up the WebFOCUS report so that a user can click on a row in the report and open the Update Assist procedure with the data from the row of the report.

Procedure: How to Call an Update Assist Maintain Procedure From a WebFOCUS Report

1. Create an Update Assist procedure that includes the following selections:
   a. In the Update Assist (Step 3 of 6) - Navigation Options window, select No key values required for your user interface.
   b. In the Update Assist (Step 5 of 6) - Output File Options window, note the names of your procedures. By default, Update Assist names them after the segments they are modifying in the data source.

2. In the Report Painter, create a report using the same data source you used for the Update Assist project. The report must contain the key fields in the segment you want to update (if you do not want to view them in the results, you can make them invisible).

3. Select the column you want to make "clickable" in the report, then select Options from the Properties menu.
   The Field Properties dialog box opens.

4. Click the Drill Down tab and select Column Data from the Active Object drop-down list. This means the users can't drill down from the title of the column in the report, only by clicking on the column's data.

5. In the Drilldown Definition drop-down list, select Maintain Procedure. This specifies that when the user clicks on one of the last names in the report, a Maintain procedure is executed.

6. Enter the name of the Maintain procedure you want to execute in the Procedure name field.
7. Specify the parameters that get passed to the Maintain procedure:
   a. Click Add in the With Parameters section.
      The Drill Down Parameter dialog box opens.
   b. For each key field in the segment, enter the following in the Parameter name field:
      \[ segname_{\text{fieldname}}_{\text{Edit}} \]
      where:
      * \textit{segname} is the name of the segment that contains the key field.
      * \textit{fieldname} is the name of the key field (a key field uniquely identifies a record in a segment).
      \_\textit{Edit} must be typed \textit{exactly} as shown. This is Update Assist's convention for naming variables.
   c. Click OK.
   d. Repeat for each parameter that is to be passed to the Maintain procedure.
      \textbf{Note:} You must pass all key fields needed to locate the record you want to update; otherwise the Maintain procedure will not have enough information to uniquely identify a record. For example, in the CAR file, if you wanted to update a particular car model, you would need to pass the country and car as well as the model type to locate the model field.

8. In the Drill Down tab, ensure the key field is selected in the list of parameters.
9. Click OK.
10. Close your procedure and save it.

When you run your report, you will see that all of the items in the selected column of the report are underlined and clickable. Clicking any item on the report opens the Update Assist form with the information for that item already filled in.

**Calling an Update Assist Project From a WebFOCUS Report Example**

This example describes how to create a report in the Report Painter and then create a link to a simple Update Assist application that will update information in the report.

This example is broken down into two steps:

1. Create an Update Assist project that updates a customer from the Videotrk data source.
   See \textit{Creating an Update Assist Project For the Videotrk Data Source} on page 314.

When you are done, you will have a WebFOCUS report that displays a list of customers in the Videotrk data source. Clicking on the last name of someone in this report will bring up a form where you can change information about a customer, or delete the customer from the data source. The result is shown in the following image:

![Image of a WebFOCUS report with customers listed and a pop-up form for updating customer information]

**Example:** Creating an Update Assist Project For the Videotrk Data Source

The following is an example of creating an Update Assist project for the Videotrk data source.

1. Create a project.
2. Add the IBISAMP directory to the viewable directories for the project.
3. Add the Videotrk Master File to your project.
4. Right-click the Videotrk Master File and click Update Assist in the pop-up menu.

   The Update Assist (Step 1 of 6) - Select Segment(s) for Auto Update window opens. For details, see Update Assist (Step 1 of 6): Selecting Segments to Update on page 290.

5. In the Select lowest segment or hierarchy you would like to auto update section, select the CUST segment.

   The CUST segment appears in the Segments section.
6. In the Segments section, select the CUST segment again. The window appears as follows:

7. Click Next.

The Update Assist (Step 2 or 6) - Segment & Field Options window opens.

8. In the Update Segment Options section, set Update and Delete to Yes.
9. In the Field View Options section, set Changeable to Yes for all fields except CUSTID, which is a key field. The window appears as follows:

![Update Assist (Step 2 of 6) - Segment & Field Options](image)

10. Click Next.

   The Update Assist (Step 3 of 6) - Navigation Options window opens.

11. Select No key values required, then click Next.

   The Update Assist (Step 4 of 6) - Template and Style Options window opens.

12. Select a color scheme from the supplied examples, and click Next.

   The Update Assist (Step 5 of 6) - Output File Options window opens.

13. Review the list of files that your application will update. You should only see one file here, cust.mnt. Update Assist derived the file name from the segment name. You can change it if you wish, but if you do, make a note of the name you use, since you will need it when you create your report. Click Next.

   The Update Assist (Step 6 of 6) - Summary window opens.

14. Review the summary of options you have chosen, and ensure Automatically run this project when you press the 'Finish' button is selected.
15. Click Finish.

WebFOCUS Maintain creates the cust.mnt file, based on the options you selected in Update Assist, and deploys it. During deployment, the Deploying Application window opens.

16. When the button at the bottom of the Deploying Application window turns to Close instead of Cancel, click Close.

WebFOCUS Maintain displays the first screen of the application you created and displays the message "Record not found." This is because the Maintain procedure that Update Assist created is expecting to be passed to a parameter containing a value for CUSTID (the key field).

---

**Example:** Creating a Report With the Videotrk Data Source

The following is an example of creating a report with the Videotrk data source.

1. Create a procedure that uses the Report Painter:
   a. Right-click the Procedures folder in the project that contains the Update Assist procedure, and select New from the pop-up window, then select Procedure.
   
   The Add Procedure dialog box opens.
   
   b. Enter update in the File Name field and click Open.

   The Open dialog box opens.
Calling an Update Assist Procedure From a WebFOCUS Report

c. Select Videotrk and click Open.
   The Report Painter opens.
   The Add Procedure dialog box opens.
   The Open dialog box opens.
   The Report Painter opens.

2. Place the LASTNAME, FIRSTNAME, CUSTID, and PHONE fields in the report. Optionally, select the LASTNAME field and click By to sort by the LASTNAME field.

3. Select the LASTNAME field on the report, and select Options from the Properties menu.
   The Field Properties dialog box opens.

4. Click the Drill Down tab.

5. In the Active Object drop-down list, select Column Data so that the users can’t drill down from the column title LASTNAME, only from the data in the report.

6. In the Drill down Definition drop-down menu, select Maintain Procedure. This specifies that when an user clicks on one of the last names in the report, a Maintain procedure is executed.

7. Enter cust in the Procedure name entry box (or, if you renamed your procedure when you created your Update Assist application, enter the new name).
The Field Properties window appears as follows:

8. Specify the parameter that gets passed to the Maintain procedure cust:

   a. Click Add in the With Parameters section.

      The Drill Down Parameter dialog box opens.

   b. Enter CUST_CUSTID_Edit in the Parameter name box.

   c. Select CUSTID from the drop-down list in the Parameter value section.
d. Click OK.

The Drill Down Parameter dialog box opens.

The window appears as follows:

9. Close your procedure and save it.
When you run your report, you will see all of the last names in the report are underlined and clickable, as in the following example:

Clicking any name on the report opens the Update Assist form with the information for that name already filled in.
Usage Notes

The following are known issues when using WebFOCUS Update Assist:

- Update Assist will not prevent the use of special characters or wildcard designations such as $* when entering data. Entering such character combinations can cause unexpected results.

- When renaming UA HTML files, the ampersand character (&) is not supported.

- Using the browser Refresh action while running an Update Assist application can cause unexpected results and is not recommended.

- Update Assist does not allow updates of cross-referenced segments. You must run the Update Assist on the individual Master Files and create separate update procedures.
Index

-HTMLFORM command 252
.GFA control file 31, 109
deleting 109
* character in Update Assist searches 304
$ character in Update Assist searches 304

A
Access Files 49
creating synonyms 49
ad hoc reports 25
adding comments 60
adding Master Files to projects 49
adding virtual folders 36
adding virtual subfolders 38
adding/removing comments 274
Advanced Allocation dialog box 155
aliases for APPROOT 265
allocating files 296
Dynamic List validation option in Update Assist 296
Allocation Wizard 138, 139, 153, 155
fields and options 153
application components 269, 270, 279
editing 269
viewing as text 279
APPROOT aliases 265
APPROOT variable 261, 262
Automatic validation in Update Assist 291
autonumbering fields in Update Assist applications 310

B
bookmarks 283, 284
text editor 283, 284

C
cache 203
clearing 203
Cascading Style Sheets (CSS) 204
case sensitivity 280, 282
cgivars.wfs file 262
color schemes for Update Assist applications 299
Combo box navigation option for Update Assist 297, 305
changing search field 305
Comment component 125, 273
comments 60, 120, 273
Comment component 120
creating 273
compiling Update Assist procedures 301
Component Connector toolbar 120, 126, 157
adding objects to procedures 120
components 40, 120
configuring deployment servers 261
controls 23
adding to forms 23
cookies 18, 21
Index

creating aliases 266
creating deployment scenarios 241
creating launch pages 24
creating project directories 31
CSS (Cascading Style Sheets) 204
customizing file types 36, 38, 59

date-stamping fields in Update Assist applications 309
ddnames 138
default partition environments 245, 253
defining fields for Update Assist applications 307
defining files 138
  Windows 138
Deploy Wizard 170, 171, 172
deploying applications to the Web 172
deploying files created with HTML Composer 252
deploying files with -HTMLFORM command 252
deploying Maintain procedures 243
deploying project files 240, 256
deployment guidelines 252
deployment scenario window 241
deployment scenarios 61, 240, 241
deployment servers 261
deployment status 257, 260
deployment steps 240
designing user interfaces 17
Developer Studio 170, 269
text editor 269
Developer Studio text editor 269
DEVELOPERS variable 261, 266
Development Tools 164
Dialogue Manager 165
directory paths 57
display formats 169, 173
documentation roadmap 13, 14
drill-down reports 25, 191, 193, 312
  using with Update Assist applications 312
Dynamic List validation option in Update Assist 296
dynamic reports 25

E
edeserve.cfg file 262
editing application components 269, 270
empty databases and Update Assist applications 304
ENGINE dialog box 165
Engine Tool 164
environments 28
  local development 28
  remote development 28
Excel 2000 172
Execute component 156
Execute Wizard 156, 157
execution search paths 255
EXPIRE_REPORTS variable 211

F
Field Validation - File dialog box 296
Field Validation - List dialog box 295
Field Validation - Range dialog box 294
fields
defining for Update Assist applications 307
selecting for Update Assist 291
file attributes 138
file types 36, 38, 59
customizing 36, 38, 59
FILEDEF command 138, 139
Windows 138
files
generated by Update Assist 300
filters 37, 59
flat files and Dynamic List validation in Update Assist 296
folders in Update Assist applications 305, 307
footings 282, 283
adding with text editor 282, 283
formatting reports 204
Cascading Style Sheets and 204
forms 23
adding controls to 23

G
generic procedures 115
graphical tools 286, 287

H
headings 282, 283
adding with text editor 282, 283
HTML Composer 17
HTML display pages 25, 230
creating 230
HTML files 17, 18, 25, 28
display pages 18, 25
HTML Files folder 29, 30
HTML forms 17, 231
creating 231
HTML pages 204
Cascading Style Sheets and 204
hyperlinks 180

I
IBFS (Information Builders File System) 32
ibiweb.cfg file 266
IBIWF_mframename variable 177
IBIWF_mprefix variable 177
IBIWF_mreports variable 177
Include component 156
Information Builders File System (IBFS) 32

J
JavaScript 191, 192
JavaScript functions 192

K
Key value navigation option for Update Assist 297

L
launch pages 18, 22, 24, 169
creating 24
customizing 24, 169
frameset 24
single-page 24
LINES parameter 200
List validation option in Update Assist 295, 296
listfill.fex file 296, 303
Local Deploy default value 61
Local Deploy scenario 241
logical names 138, 139
   Allocation Wizard 139
   FILEDEF command and 138
   Windows 138
logon pages 18, 21
   modifying 21

M

main pages 18
Maintain Files folder 29
managing projects 72
Master Files 28, 49, 50, 51
   adding to projects 49, 50
   creating synonyms 49
   deleting permanently 51
   removing from projects 50
Master Files folder 29, 30, 51
   upload data file 51
maxFolders JavaScript variable 307
maxNodesPerFolder JavaScript variable 307
menu options 212, 213, 216, 219, 220, 226, 227
   customizing 212, 213, 216, 219, 220, 226, 227
Microsoft Internet Explorer (IE) 170
MIME (Multipurpose Internet Mail Extensions) 170, 171
mntSignon.htm file 303
modifying directory paths 57
modifying project properties 56
multiple reports 176, 177
Multipurpose Internet Mail Extensions (MIME) 170, 171

N

calbar.htm file 303, 305, 307
   changing search field 305
   customizing tree control 307
Navigation options in Update Assist 297
new databases and Update Assist applications 304
New Filter dialog box 71
New Procedure dialog box 117
No key values navigation option for Update Assist 297, 312
   using with WebFOCUS reports 312

O

on-demand paging 198, 200, 203
opening graphical tools 286, 287
organizing projects 35
Other component 124, 269, 272
   text editor 272
Other folder 30
output files generated by Update Assist 300

P

parameters 156, 157
   passing to procedures 156, 157
Partial-Key option in Update Assist 294, 304
partition project files window 259
partitioning project files 240, 249
passwords 108, 297
   prompting for in Update Assist applications 297
PDF display format 169
procedure components 120, 121, 126, 127, 128, 129, 279, 285
   checking 128
procedure components (continued)
copying 129
creating 120, 121, 126
dragging and dropping 129
editing 128
running 127, 285
types 120
viewing as text 279
procedure creation tools 112, 113
activating OLAP 166
adding to applications 118
adding to projects 118
calling 156
calling procedures 156, 157
canceling 168
components 120
copying 120
creating 111, 183, 185, 188, 190, 227, 238, 276
creating with text editor 116
debugging 166
decrypting 166, 167
deleting 119
editing 166, 168, 270
embedding 157
encrypting 166, 167
including 157
passing parameters 156, 157
publishing 166
running 285
running locally 168
scheduling 166, 167
selecting tool for creating 113
text components 277
tools for creating 112, 113

Project files 249, 256
deploying 256
partitioning 249
project properties 56
Project Properties dialog box 57, 63, 248
Project Wizard 29
Project-based development environments 28
projects 29, 30, 31, 35, 57, 72, 108, 109, 248
applying read-only security 57
creating directories 31
decrypting 108
deleting 109
encrypting 108
file types 30
managing 72
organizing 35
removing 109
securing 108
tracking code 72
viewing properties 57, 248
Projects area 41, 42, 46, 47, 48
changing assigned projects to the workspace file 48
changing WebFOCUS environment for workspace files 47
closing workspace files 47
creating workspace files 42
inserting projects into the workspace file 48
loading workspace files 46
removing workspace files 48
renaming workspace files 46
saving workspace files 46
workspace files 41
workspace files properties dialog box 48

R

Range validation option in Update Assist 294
read-only security 57
registered file types 60
renaming files generated by Update Assist 300
renaming segments or fields in Update Assist 293
report formatting 204
  Cascading Style Sheets and 204
report types 25, 191, 193
  ad hoc 25
  drill-down 25, 191, 193
  dynamic 25
  static 25
ReportCaster 167
  scheduling procedures 167
reporting applications 27
reports 130, 169, 170, 171, 172, 173, 176, 177, 180, 181, 183, 198, 200, 204, 211, 270, 312
  calling Update Assist applications 312
  displaying 169, 170, 171, 172
  displaying in multiple formats 169, 176, 177, 180, 181, 183
  editing as text 270
  formatting 173, 204
  on-demand paging 198, 200
  SQL 130
  viewing 211
resorting fields in Update Assist 293
restricting deployment authorization 266
Results page for Update Assist applications 297
running paths 255
running procedures 168
  locally 168
running Update Assist applications 302
search paths 255
security 18, 108, 297
  in Update Assist applications 297
SegmentName_validationInit.js file 303
SegmentName.mnt file 303, 309, 310
  autonumbering fields 310
  date-stamping fields 309
segments 290
  selecting for Update Assist 290
selecting deployment scenarios 61
self-service reporting applications 27
SESSION project 29
SET command 200
SET parameters 200, 201
  LINES 200
  WEBVIEWHOME 201
setting deployment paths 261
setting deployment paths for AS/400 265
setting deployment paths for z/OS 264
setting running paths 255
setting Web server deployment path 262
setting WebFOCUS Reporting Server deployment path 263
skins for Update Assist applications 299
Smart Deploy 241
sorting fields in Update Assist 293
source control 72, 73, 74, 75, 78, 79, 80, 81, 84, 91, 96, 102, 103, 105, 108
  check files in and out 78
  ClearCase 81
  CVS 84
  enabling 73
  icons 80
  managed reporting 75
  Microsoft Visual SourceSafe 108

S

Scenario Editor 241
Search option in Update Assist applications 294, 304
source control (continued)
   options 74, 105
      accessing 74
   requirements 73
   SCC_Name Variable 103
   set source control options 96
   setting up 96
   Subversion (SVN) 91
   validating 102
   view options 79
   WebFOCUS Client 103
source file management 72
SQL 130
   executing code 130
SQL Report Wizard 130
Static List validation option in Update Assist 295
static reports 25
steps for deployment 240
style sheets 204
   Cascading Style Sheets (CSS) 204
subfolders for Web deployment 250
synonyms 49
syntax coloring 280, 281

T

   text
      finding 279
      replacing 279
   text colors 280, 281
      changing 280, 281
   text components 277
      creating 277
   text editor 116, 269, 270, 272, 276, 279, 280, 281, 282, 283, 286, 287
      adding footings 282, 283
      adding headings 282, 283
      case sensitivity 280, 282
      changing default settings 280, 281

   text editor (continued)
      components 270
      creating procedures 276
      creating text files 276
      opening files 279
      opening graphical tools 286, 287
      Other component 272
   text editor window 279
   text files 276, 277
      creating 276, 277
   tracking project code 72
   Tree navigation option for Update Assist 297, 305, 307
      changing search field 305
      customizing 307

U

   Update Assist 289, 302, 305, 312
      about the application 302
      calling applications from WebFOCUS reports 312
      editing applications 305
      generating application 289
   uploading data files 51, 55
   User interface for Update Assist applications 297
   user interfaces 17, 18
      designing 17

V

   validation in Update Assist 291
   validationInit.js files 303
   variables for setting deployment path 261
   version control 72
   viewing project properties 56
   virtual folders 29, 36
      adding 36
virtual subfolders 38

W

Web browsers 21
  cookies 21
Web deployment subfolders 250
WebFOCUS Client 24
WebFOCUS Client variables 176, 177, 183
  IBIWF_mframename 177
  IBIWF_mprefix 177
  IBIWF_mreports 177
WebFOCUS code 270
  editing 270
WebFOCUS files
  Windows 138
WebFOCUS Reporting Server 18
WebFOCUS StyleSheets 169
WebFOCUS Viewer 198, 199, 200, 201, 202, 203
  closing 202
  home pages 201, 202
  navigating 203
  opening 202
WEBVIEWER parameter 200
WEBVIEWHOME parameter 201, 202
WEBVIEWTARG parameter 202
wildcard characters in Update Assist searches 304
Windows
  Allocation Wizard 138, 139
  defining files 138
  FILEDEF command 138
  logical names 138
workspace file 41, 42, 46, 47, 48
  changing WebFOCUS environment 47
  changing assigned projects 48
  closing files 46
  creating files 41, 42
  inserting projects 48
  loading files 46
  properties dialog box 48
  removing files 48
  renaming files 46
  saving files 46
X

XML control file 31
Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

**Mail:** Documentation Services - Customer Support
Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898

**Fax:** (212) 967-0460

**E-mail:** books_info@ibi.com

**Web form:** http://www.informationbuilders.com/bookstore/derf.html

Name:__________________________________________________________________________

Company:_____________________________________________________________________

Address:______________________________________________________________________

Telephone:_________________________ Date:_____________________________________

Email:________________________________________________________________________

Comments:___________________________________________________________________