Information Builders.

# WebFOCUS

WebFOCUS Developing Reporting Applications
Version 5 Release 2

DN4500514.0103

Cactus, EDA/SQL, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, FOCUS Vision, Hospital-Trac, Information Builders, the Information Builders logo, Parlay, PC/FOCUS, SmartMart, SmartMode, SNAPpack, TableTalk, WALDO, Web390, WebFOCUS and WorldMART are registered trademarks and EDA, iWay, and iWay Software are trademarks of Information Builders, Inc.

Acrobat and Adobe are registered trademarks of Adobe Systems Incorporated.

HP MPE/iX is a registered trademark of Hewlett Packard Corporation.

Informix is a registered trademark of Informix Software, Inc.

Intel and Pentium are registered trademarks of Intel Corporation.

ACF/VTAM, AIX, AS/400, CICS, DB2, DRDA, Distributed Relational Database Architecture, IBM, MQSeries, MVS/ESA, OS/2, OS/390, OS/400, RACF, RS/6000, S/390, VM/ESA, VSE/ESA, VTAM, and WebSphere are registered trademarks and DB2/2, Hiperspace, IMS, iSeries, MVS, QMF, SQL/DS, z/OS and z/VM are trademarks of International Business Machines Corporation.

ActiveX, BizTalk, FrontPage, Microsoft, MS-DOS, PowerPoint, Visual Basic, Visual C++, Visio, Visual FoxPro, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Motorola is a trademark of Motorola, Inc.

Netscape, Netscape FastTrack Server, and Netscape Navigator are registered trademarks of Netscape Communications Corporation.

ServletExec is a trademark of New Atlanta Communications, LLC.

Oracle is a registered trademark and Rdb is a trademark of Oracle Corporation.

Reliant UNIX is a registered trademark of Fujitsu Siemens Computers.

Java and all Java-based marks, NetDynamics, Solaris, SunOS, and iPlanet are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

PowerBuilder, Powersoft, and Sybase are registered trademarks and SQL Server is a trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. 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 © 2003, by Information Builders, Inc. All rights reserved. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Printed in the U.S.A.

# Preface

This documentation describes how to use WebFOCUS to create and deploy report applications on the Internet or coporate intranets. It is intended for application developers. This documentation is part of the WebFOCUS documentation set.

## How This Manual Is Organized

This manual includes the following chapters:

| Chapter/Appendix | | Contents |
|---|---|---|
| **Application Development** | | |
| **1** | *Where Am I?* | Explains where to find the information you are looking for within the WebFOCUS documentation set. |
| **2** | *WebFOCUS Application Logic* | Describes the underlying logic of WebFOCUS that exists from the process of developing your application to the time a user launches a report requests in a Web browser and sees the results return to the browser. |
| **3** | *Creating a Reporting Application* | Describes how to create a reporting application that in the local development environment. |
| **4** | *Creating a Reporting Procedure* | Describes how to create a procedure, which is the core element of a reporting application. |
| **5** | *Understanding User Interface Basics* | Introduces user interface design in WebFOCUS and desribes the componenets of a typical user interface. |
| **6** | *Designing a User Interface for a Web Application With the Report Layout Painter* | Describes how to create a user interface in the local development environment using the Report Layout Painter (previously the Layout tool). |
| **7** | *Designing a User Interface With FrontPage and the Publish Utility* | Describes how to create a user interface in the local development environment using Microsoft FrontPage and the Information Builders-supplied Publish Utility. |
| **8** | *Coding a User Interface* | Describes how to code a user interface with HTML and JavaScript. |

| Chapter/Appendix | | Contents |
|---|---|---|
| **9** | *Enhancing a User Interface* | 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. |
| **10** | *Managing Flow of Control in an Application* | Describes how to control the processing of a procedure with Dialogue Manager. |
| **11** | *Testing and Debugging a Procedure* | Describes how to diagnose problems encountered when running a procedure. |
| **12** | *Partitioning and Deploying Project Files* | Describes the steps you will perform for successful deployment of your project files to the Web. |
| **Developer's Reference** | | |
| **13** | *Managing the Server Environment* | Describes how to manage your WebFOCUS Reporting Server environment in Developer Studio. |
| **14** | *Accessing a FOCUS Data Source* | Describes how to assign a logical name to a FOCUS data source. |
| **15** | *Customizing Your Environment* | Describes hot to control your WebFOCUS Environment with the SET command. |
| **16** | *Defining and Allocating WebFOCUS Files* | Provides file allocation requirements for WebFOCUS operating systems, and describes hot to assign a logical name to a file or device. |
| **17** | *Euro Currency Support* | Describes how to create and use a currency data source to convert to and from the new euro currency. |
| **18** | *Editing Application Components as Text in Developer Studio* | Describes how 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 with the Developer Studio text editor. |
| **19** | *Using the Command Console* | Explains how to use the Command Console to send commands directly to a server from Developer Studio. |

| Chapter/Appendix | | Contents |
|---|---|---|
| **20** | *Creating an Update Application With Update Assist* | Describes how to use the Update Assist to create an update application in Developer Studio. |

## Documentation Conventions

The following conventions apply throughout this manual:

| Convention | Description |
|---|---|
| `THIS TYPEFACE` or `this typeface` | Denotes syntax that you must enter exactly as shown. |
| *`this typeface`* | Represents a placeholder (or variable) in syntax for a value that you or the system must supply. |
| `underscore` | Indicates a default setting. |
| *this typeface* | Represents a placeholder (or variable) in a text paragraph, a cross-reference, or an important term. It may also Indicate a button, menu item, or dialog box option you can click or select. |
| **this typeface** | Highlights a file name or command in a text paragraph that must be lowercase. |
| Key + Key | Indicates keys that you must press simultaneously. |
| `{ }` | Indicates two or three choices; type one of them, not the braces. |
| `[ ]` | Indicates a group of optional parameters. None are required, but you may select one of them. Type only the parameter in the brackets, not the brackets. |
| `\|` | Separates mutually exclusive choices in syntax. Type one of them, not the symbol. |
| `...` | Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis points (…). |
| `.`<br>`.`<br>`.` | Indicates that there are (or could be) intervening or additional commands. |

## Related Publications

To view a current listing of our publications and to place an order, visit our World Wide Web site, http://www.informationbuilders.com. You can also contact the Publications Order Department at (800) 969-4636.

## Customer Support

Do you have questions about WebFOCUS?

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 WebFOCUS 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 (*xxxx.xx*) when you call.

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. 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.

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 most effectively, please provide the following information when you call:

- 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.
  - The server you are accessing, including release.
- The stored procedure (preferably with line numbers) or FOCUS commands being used in server access.

- The name of 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, WebFOCUS, 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 the code to access a single 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 relay suggestions for improving the publication or to alert us to corrections. You can also use the Documentation Feedback form on our Web site, http://www.informationbuilders.com.

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.

# Contents


**1. Where Am I? . . . . . 1-1**
WebFOCUS Core Documentation Roadmap . . . . . 1-2

**2. WebFOCUS Application Logic . . . . . 2-1**
Three-tier Application Logic . . . . . 2-2
Distributing Processing Among Platforms . . . . . 2-5
Deploying a Partitioned Project . . . . . 2-6
WebFOCUS Components . . . . . 2-6
Storing and Searching for Application Files . . . . . 2-12
Handicapped Accessibility in WebFOCUS . . . . . 2-14

**3. Creating a Reporting Application . . . . . 3-1**
What Is a Reporting Application? . . . . . 3-2
Supported Environments . . . . . 3-2
Creating a Project . . . . . 3-3
Organizing a Project . . . . . 3-9
Adding Master Files to a Project . . . . . 3-12
Viewing and Modifying Project Properties . . . . . 3-14
Tracking the Code for a Project . . . . . 3-27
Securing a Project . . . . . 3-27
Removing a Project . . . . . 3-27

**4. Creating a Reporting Procedure . . . . . 4-1**
Selecting a Creation Tool . . . . . 4-2
Incorporating a Procedure Into an Application . . . . . 4-10
Copying a Procedure . . . . . 4-13
Creating a Procedure Component . . . . . 4-13
Working With a Component . . . . . 4-18
Calling a Procedure From the Current One . . . . . 4-22
Managing Flow of Control . . . . . 4-24
Working With a Full Procedure . . . . . 4-25
Running a Procedure . . . . . 4-27
Canceling a Running Procedure . . . . . 4-29

**5. Understanding User Interface Basics . . . . . 5-1**
User Interface Basics . . . . . 5-2
About a Logon Page . . . . . 5-2
Setting a Browser . . . . . 5-5
About a Launch Page . . . . . 5-6
Adding a Control to a Form . . . . . 5-7
Types of Launch Pages . . . . . 5-8
Ways to Create a Launch Page . . . . . 5-8
About an HTML Display Page . . . . . 5-9

**6. Designing a User Interface for a Web Application With Report Layout Painter .6-1**
Uses for the Report Layout Painter .....6-2
Getting Started .....6-3
Creating an HTML Page Layout .....6-8
Adding a Report or Graph to the Layout .....6-8
Adding a Frame to the Layout ..... 6-14
Adding a Line to the Layout ..... 6-17
Adding an Image to the Layout ..... 6-19
Adding Text to the Layout ..... 6-21
Adding a Hyperlink to the Layout ..... 6-23
Adding a Group Box to the Layout ..... 6-24
Adding a Push Button to the Layout ..... 6-27
Adding a WebFOCUS Form ..... 6-30
Using WebFOCUS Forms to Supply Parameter Values ..... 6-31
Using a Text Box ..... 6-32
Using a Drop-Down List ..... 6-34
Using a List Box ..... 6-36
Using Check Boxes ..... 6-38
Using Radio Buttons ..... 6-42
Working With the Properties Window ..... 6-44
Working With the Parameters Window ..... 6-45
Styling Your Layout ..... 6-51
Adding an HTML Template to the Layout ..... 6-51
Adding a Theme to the Layout ..... 6-51
Setting HTML Page Properties ..... 6-52
Laying Out Objects With the Report Layout Painter ..... 6-53
Setting Relationships Between Objects ..... 6-53
Customizing the Layout ..... 6-56
Controlling the Report Layout Painter Environment ..... 6-56

**7. Designing a User Interface With FrontPage .....7-1**
Creating a Launch Page Overview .....7-2
Creating a Sample Launch Page for an Inventory Report .....7-9
Adding JavaScript to Automatically Run a Report ..... 7-28
Displaying a Report and Graph ..... 7-35
Creating a Main Page ..... 7-38

**8. Coding a User Interface .....8-1**
Which Tools Can You Use? .....8-2
Choosing Search Path Logic .....8-3
The WebFOCUS Client .....8-7
Using the CGI .....8-7
Using ISAPI ..... 8-14

Using the WebFOCUS Servlet . . . . . 8-16
Configuring the WebFOCUS Servlet . . . . . 8-16
Calling the WebFOCUS Servlet . . . . . 8-19
Using a Dynamic Multi-Select Drop-Down List . . . . . 8-19
Enabling Ad Hoc Reporting . . . . . 8-24
Validating a Form With JavaScript . . . . . 8-28
Amper Auto-Prompting in CGI/Servlet . . . . . 8-31
Amper Auto-Prompting Examples . . . . . 8-31
Displaying a Report on the Default WebFOCUS Page . . . . . 8-36
Designing an HTML Page for Report Display . . . . . 8-42

**9. Enhancing a User Interface . . . . . 9-1**
Displaying a Report in a Helper Application . . . . . 9-2
Controlling Multiple Reports . . . . . 9-8
Adding JavaScript for Drill-Down Reporting . . . . . 9-23
Facilitating Report Manipulation . . . . . 9-29
Using the WebFOCUS Viewer . . . . . 9-30
Opening and Closing the WebFOCUS Viewer . . . . . 9-31
Printing With On-Demand Paging . . . . . 9-32
WebFOCUS Viewer Navigation . . . . . 9-33
Using a Cascading Style Sheet to Standardize Display . . . . . 9-33
Displaying a Previously Run Report . . . . . 9-41
Passing a User ID From HTML for a Custom Menu . . . . . 9-42

**10. Managing Flow of Control in an Application . . . . . 10-1**
Uses for Dialogue Manager . . . . . 10-2
Dialogue Manager Processing . . . . . 10-4
Creating a Dialogue Manager Procedure . . . . . 10-7
Commenting a Procedure . . . . . 10-8
Customizing a Procedure With Variables . . . . . 10-9
Supplying a Default Variable Value . . . . . 10-12
Supplying Variable Values in an Expression . . . . . 10-13
Supplying Variable Values From Another Procedure . . . . . 10-14
Reading Variable Values From and Writing Variable Values to an External File . . . . . 10-16
Supplying Variable Values From Report Painter in Developer Studio . . . . . 10-21
Supplying Dates as Variable Values . . . . . 10-22
Screening a Variable Value . . . . . 10-23
Evaluating a Variable Immediately . . . . . 10-27
Counting With an Indexed Variable . . . . . 10-29
Creating and Working With Variables . . . . . 10-30
Creating and Working With Local and Global Variables . . . . . 10-31
Working With System and Statistical Variables . . . . . 10-36

Controlling the Execution of a Procedure . . . . . 10-41
Executing Stacked Commands and Continuing the Procedure . . . . . 10-41
Executing Stacked Commands and Exiting the Procedure . . . . . 10-42
Canceling the Execution of a Procedure . . . . . 10-43
Navigating a Procedure . . . . . 10-44
Performing Unconditional Branching . . . . . 10-45
Looping in a Procedure . . . . . 10-50
Calling Another Procedure With -INCLUDE . . . . . 10-53
Calling Another Procedure With EXEC . . . . . 10-57
Enhancing an HTML Web Page With a Procedure . . . . . 10-58
Referring to an External Web Page . . . . . 10-59
Embedding HTML Commands in a Procedure . . . . . 10-59
Embedding a Procedure in an HTML Web Page . . . . . 10-62
Including Variables in an HTML Web Page . . . . . 10-65
Issuing Operating System Commands . . . . . 10-66
Reviewing Command Output . . . . . 10-67
Controlling Passwords With Dialogue Manager . . . . . 10-67
Sending a Message to the Application . . . . . 10-68
Testing and Debugging a Dialogue Manager Procedure . . . . . 10-69
Dialogue Manager Syntax Reference . . . . . 10-71

**11. Testing and Debugging a Procedure . . . . . 11-1**
Debugging Your Application With Query Commands . . . . . 11-2
Displaying Combined Structures . . . . . 11-4
Displaying Virtual Fields . . . . . 11-5
Displaying the Currency Data Source in Effect . . . . . 11-6
Displaying Available Fields . . . . . 11-7
Displaying the File Directory Table . . . . . 11-8
Displaying Field Information for a Master File . . . . . 11-9
Displaying Data Source Statistics . . . . . 11-11
Displaying Current ddnames Assigned With FILEDEF . . . . . 11-13
Displaying Defined Functions . . . . . 11-14
Displaying HOLD Fields . . . . . 11-14
Displaying JOIN Structures . . . . . 11-15
Displaying National Language Support . . . . . 11-17
Displaying Explanations of Error Messages . . . . . 11-18
Displaying the Current Search Path . . . . . 11-19
Displaying the Release Number . . . . . 11-20
Displaying the Values of a Remote Server . . . . . 11-20
Displaying Parameter Settings . . . . . 11-21
Displaying Parameters That Cannot Be Set in an Area . . . . . 11-23
Displaying Graph Parameters . . . . . 11-24
Displaying Command Statistics . . . . . 11-25

Displaying StyleSheet Parameter Settings ..... 11-26
Displaying Information About the SU Machine ..... 11-29
Displaying Data Sources Specified With USE ..... 11-29
Displaying Global Variable Values ..... 11-30
Identifying the Files Being Used ..... 11-30

**12. Partitioning and Deploying Project Files ..... 12-1**
Deployment Basics ..... 12-2
Summary of Steps ..... 12-2
Step 1. Identify the Target Servers ..... 12-3
Step 2. Create a Deployment Scenario ..... 12-3
Step 3. Partition the Project Files ..... 12-5
Working in a Default Partition Environment ..... 12-8
Step 4. Deploy the Project Files ..... 12-9
Configuring the Target Servers (Optional) ..... 12-14

**13. Managing the WebFOCUS Environment ..... 13-1**
Adding an Environment ..... 13-2
Editing Environment Properties ..... 13-4
Working With WebFOCUS Environment Properties ..... 13-4
Web Server Properties ..... 13-7
WebFOCUS Client Properties ..... 13-8
Managed Reporting/ReportCaster Properties ..... 13-9
WebFOCUS Server Properties ..... 13-10
Project Development Properties ..... 13-11
Setting the Development Environment ..... 13-12
Refreshing or Removing an Environment ..... 13-13

**14. Accessing a FOCUS Data Source ..... 14-1**
The USE Command ..... 14-2
The Developer Studio Use Tool ..... 14-3
Identifying a FOCUS Data Source ..... 14-5
Using Alternative File Specifications ..... 14-8
Identifying a New Data Source ..... 14-11
Protecting a Data Source ..... 14-13
Concatenating Data Sources ..... 14-14
Displaying the Current USE Options ..... 14-20
Clearing the USE Options ..... 14-20

**15. Customizing Your Environment ..... 15-1**
When Do You Use the SET Command? ..... 15-2
Ways to Issue a SET Command ..... 15-2
Issuing a SET Command With the Set Tool ..... 15-3
Coding a SET Command ..... 15-5

Types of SET Parameters . . . . . 15-10
Calculations . . . . . 15-11
Data and Metadata . . . . . 15-11
Date Manipulation Tasks . . . . . 15-12
WebFOCUS Specific Tasks . . . . . 15-13
Developer Studio Specific Tasks . . . . . 15-13
Graph Tasks . . . . . 15-14
Memory Setup and Optimization Tasks . . . . . 15-15
Report Code and Content Tasks . . . . . 15-16
Report Layout and Display Tasks . . . . . 15-18
Security Tasks . . . . . 15-19
SET Parameter Syntax . . . . . 15-20

**16. Defining and Allocating WebFOCUS Files . . . . . 16-1**
WebFOCUS Files Under Windows NT/2000/XP . . . . . 16-2
Dynamically Defining a File Under Windows NT/2000/XP . . . . . 16-4
Assigning a Logical Name Under Windows NT/2000/XP . . . . . 16-5
Displaying Current ddnames Assigned With FILEDEF . . . . . 16-10
Clearing FILEDEF Assignments . . . . . 16-10
Application Files Under Windows NT/2000/XP . . . . . 16-11
Master Files Under Windows NT/2000/XP . . . . . 16-11
Locating a Master File Under Windows NT/2000/XP . . . . . 16-11
Access Files Under Windows NT/2000/XP . . . . . 16-12
Procedures Under Windows NT/2000/XP . . . . . 16-12
FOCUS Data Sources Under Windows NT/2000/XP . . . . . 16-13
External Indexes for FOCUS Data Sources Under Windows NT/2000/XP . . . . . 16-13
Other Supported Data Sources Under Windows NT/2000/XP . . . . . 16-13
Sequential Data Sources Under Windows NT/2000/XP . . . . . 16-14
WebFOCUS StyleSheet Files Under Windows NT/2000/XP . . . . . 16-14
Library of Functions Under Windows NT/2000/XP . . . . . 16-15
Profiles Under Windows NT/2000/XP . . . . . 16-15
Web Pages Under WebFOCUS Under Windows NT/2000/XP . . . . . 16-15
Extract Files Under Windows NT/2000/XP . . . . . 16-16
HOLD Files Under Windows NT/2000/XP . . . . . 16-16
SAVB Files Under Windows NT/2000/XP . . . . . 16-17
SAVE Files Under Windows NT/2000/XP . . . . . 16-17
HOLDMAST Files Under Windows NT/2000/XP . . . . . 16-18
Work Files Under Windows NT/2000/XP . . . . . 16-19
WebFOCUS Files Under MVS . . . . . 16-20
Referencing Files Under MVS . . . . . 16-20
Dynamically Defining Files Under MVS . . . . . 16-22

Application Files Under MVS . . . . . 16-22
Master Files Under MVS . . . . . 16-22
Access Files Under MVS . . . . . 16-23
Procedures Under MVS . . . . . 16-23
FOCUS Data Sources Under MVS . . . . . 16-24
Using FOCUS Data Sources Under MVS . . . . . 16-24
External Indexes for FOCUS Data Sources Under MVS . . . . . 16-25
Other Supported Data Sources Under MVS . . . . . 16-25
WebFOCUS StyleSheet Files Under MVS . . . . . 16-25
Profiles Under MVS . . . . . 16-26
Extract Files Under MVS . . . . . 16-26
HOLD Files Under MVS . . . . . 16-26
SAVB Files Under MVS . . . . . 16-27
SAVE Files Under MVS . . . . . 16-27
HOLDMAST Files: Temporary Master Files Under MVS . . . . . 16-28
HOLDSTAT Files Under MVS . . . . . 16-28
Work Files Under MVS . . . . . 16-29
Reviewing Attributes of Allocated Files Under MVS . . . . . 16-30
Displaying Allocated Files With ? TSO DDNAME Under MVS . . . . . 16-30
Displaying File Attributes With ? TSO DDNAME ddname Under MVS . . . . . 16-31
Displaying File Attributes With ? TSO DDNAME ddname . . . . . 16-31
Placing File Attributes in Dialogue Manager Commands . . . . . 16-32
Determining If a Data Source Exists Under MVS . . . . . 16-34
Estimating Data Set Sizes to Determine Available Space Under MVS . . . . . 16-35
WebFOCUS Files Under UNIX . . . . . 16-36
Dynamically Defining Files Under UNIX . . . . . 16-37
Assigning a Logical Name Under UNIX . . . . . 16-38
Displaying Current ddnames Assigned With FILEDEF . . . . . 16-39
Clearing FILEDEF Assignments . . . . . 16-39
Application Files Under UNIX . . . . . 16-40
Master Files Under UNIX . . . . . 16-40
Access Files Under UNIX . . . . . 16-41
Procedures Under UNIX . . . . . 16-41
FOCUS Data Sources Under UNIX . . . . . 16-42
External Indexes for FOCUS Data Sources Under Windows UNIX . . . . . 16-42
Sequential Data Sources Under UNIX . . . . . 16-42
WebFOCUS StyleSheets Under UNIX . . . . . 16-43
Profiles Under UNIX . . . . . 16-43

Extract Files Under UNIX ..... 16-44
HOLD Files Under UNIX ..... 16-44
SAVB Files Under UNIX ..... 16-44
SAVE Files Under UNIX ..... 16-45
HOLDMAST Files Under UNIX ..... 16-45
Work Files Under UNIX ..... 16-46

**17. Euro Currency Support ..... 17-1**
Integrating the Euro Currency ..... 17-2
Converting Currencies ..... 17-2
Creating the Currency Data Source ..... 17-4
Identifying Fields That Contain Currency Data ..... 17-7
Activating the Currency Data Source ..... 17-9
Processing Currency Data ..... 17-10
Querying the Currency Data Source in Effect ..... 17-15
Punctuating Numbers ..... 17-16
Using the Extended Currency Symbol Format ..... 17-17

**18. Editing Application Components as Text in Developer Studio ..... 18-1**
Text Editor ..... 18-2
The Other Component ..... 18-3
The Comment Component ..... 18-4
Creating a Text File or a Procedure Component as Text ..... 18-5
Opening Application Components as Text ..... 18-7
Finding and Replacing Text ..... 18-8
Changing Text Color and Case Size ..... 18-8
Adding Headings and Footings. ..... 18-11
Using Bookmarks to Move Within a File ..... 18-12
Running a Procedure From the Editor ..... 18-13
Opening a Graphical Tool From the Text Editor ..... 18-14

**19. Using the Command Console ..... 19-1**
Command Console Window ..... 19-2
Working in the Command Console ..... 19-4

**20. Creating an Update Application With Update Assist ..... 20-1**
Using Update Assist ..... 20-2
Calling an Update Assist Procedure From a WebFOCUS Report ..... 20-12
Calling an Update Assist Project From a WebFOCUS Report Example ..... 20-14

CHAPTER 1

# Where Am I?

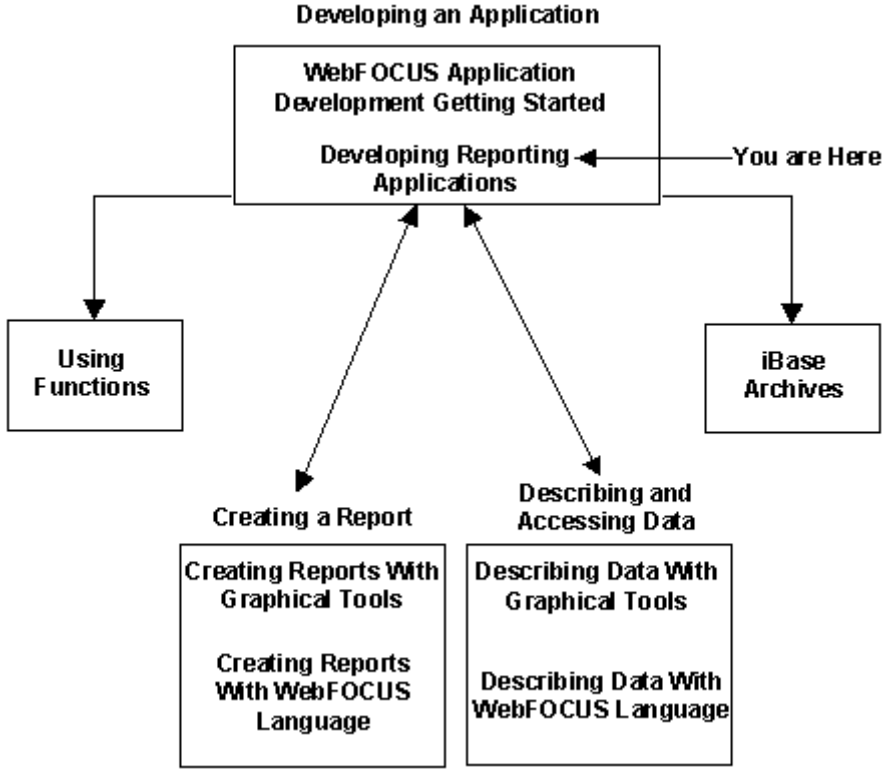
**Topic:**

- WebFOCUS Core Documentation Roadmap

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.

# 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...**

- *WebFOCUS Application Development Getting Started* provides an overview of the process of developing an application in WebFOCUS Developer Studio and deploying it to the Web. *Getting Started* also includes a description of the tools that are available for developing and deploying your applications as well as a tutorial to familiarize you with the development process.
- *WebFOCUS Developing Reporting Applications* 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. This includes detailed instructions on how to create the procedures and user interface that will comprise your application, and the process for deploying the application to the Web.

**If you are creating a report request...**

- *Creating Reports With WebFOCUS Language* describes how to create, style, and 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 (Windows version). It also includes information on Report Caster and mobile technology.

**If you are an administrator or developer setting up 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 (Windows version) 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 use Information Builders-supplied functions to perform complex calculations and manipulate data in your procedures.

**If you do not find information previously contained in Developing Reporting Applications...**

- *Functions and Subroutines* is a stand-alone manual titled *Using Functions*. This manual describes all reporting and Maintain functions, and explains how they can be used to perform calculations and manipulate data.
- *Connecting the Dots: Develop and Deploy Solution* is located in *WebFOCUS Application Development Getting Started*.
- The following topics previously found in *Developing Reporting Applications* are now located in the legacy repository:
  - *Working With Cross-Century Dates*
  - *Establishing String Substitutions*
  - *Designing User Interfaces For Windows Reporting Applications*

  The legacy repository 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.

### 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://www.documentation.ibi.com.

# Application Development

CHAPTER 2

# WebFOCUS Application Logic

**Topics:**

- Three-tier Application Logic
- Deploying a Partitioned Project
- WebFOCUS Components
- Storing and Searching for Application Files
- Handicapped Accessibility in WebFOCUS

To develop self-service reporting applications for the Web, you need software components that manage the flow from the time you begin to develop your project to the time a user launches a report request in a Web browser and sees the results return to the browser.

A project is a WebFOCUS application being developed on the local server. After building and testing the project, you can deploy it as a Web application.

During the development phase, you can use WebFOCUS Developer Studio, a Windows-based graphical environment, to build, test, and deploy your applications to the Web. Developer Studio adds the ease and speed of graphical development to the traditional power of WebFOCUS to access data, process requests, and deliver information to large communities of users.

Whether you are testing a project while building it in Developer Studio or deploying the project to end users who will access it from Web browsers, you can depend on WebFOCUS' underlying application logic to manage the process transparently, requiring only that the components have been properly installed and configured.

# Three-tier Application Logic

WebFOCUS's three-tier application logic is designed to exploit the capabilities of Web processing and enhance an application's performance, scalability, and maintenance by *partitioning* an application into logical components. Partitioning is a means of classifying the functionality of an application. Partitioned components can be deployed across many different platforms. See *Distributing Processing Among Platforms* on page 2-5.

The three tiers of application logic are:

- **Presentation logic**, which is commonly known as the user interface, or the application's front-end. In a WebFOCUS reporting application, presentation logic consists of Web pages (including HTML files, Cascading StyleSheets, JavaScript™ classes, and images).

  Presentation logic resides the Web server and is handled by the Web server and the end user's Web browser.

- **Program logic**, which is any application logic other than presentation or data access logic. Procedures provide the application program logic. They include the executable application objects that display reports or graphs, or perform more complicated processing based on values that the end user supplies.

  Program logic resides on the WebFOCUS Reporting Server and consists of procedures (FOCEXECs), WebFOCUS StyleSheets, certain customized HTML files, and embedded graphics.

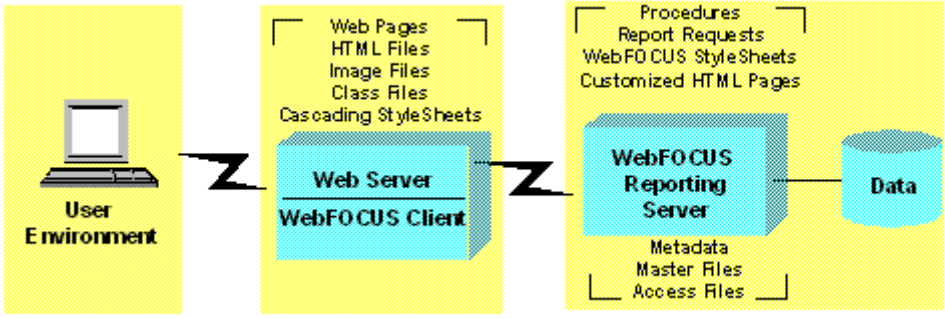
- **Data access logic**, which retrieves information from a data source. Data access logic contains the metadata that defines the data sources your applications access. It consists of Master Files and Access Files.

  Data access logic usually resides on the same server as the data itself (either the WebFOCUS Reporting Server or a sub-server). A WebFOCUS reporting application takes advantage of the data access logic of Information Builders middleware technology.

For an illustration, see *Partitioned Application Files and Servers* on page 2-3.

## *Reference* Partitioned Application Files and Servers

The following diagram illustrates the relationships among WebFOCUS components, including the locations of application files on their respective servers.




The *Web server* processes the Web pages that provide the presentation logic for a WebFOCUS reporting application. The Web server accesses HTML files, graphical image files, and Java class files, and Cascading Style Sheets. For more information about how to specify file locations on the Web server, see the WebFOCUS installation guide for your platform.

The *WebFOCUS Reporting Server* processes the procedures that supply the program logic for a reporting application, as well as the files that contain metadata for the data sources. Stored on the WebFOCUS Reporting Server are report requests, WebFOCUS StyleSheets, customized HTML pages called from WebFOCUS procedures, and Master and Access Files.

The WebFOCUS Reporting Server also accesses the data sources used by an application.

## *Reference* WebFOCUS File Types

A WebFOCUS reporting application can deploy and use all of the following types of files. The following chart includes file extensions used by the NT and UNIX operating systems. See Chapter 16, *Defining and Allocating WebFOCUS Files* for comparable information for OS/390® and other supported operating environments.

| File Type | Description | Location After Deployment | File Extension (NT/UNIX) |
|---|---|---|---|
| Web page | Includes files displayed for the end user through a Web browser, such as HTML files, graphical images, Java executable objects (class files), and Cascading Style Sheets. | In Web server home directory or in Web server alias. | .htm<br>.html<br>.jpg<br>.gif<br>.css<br>.js<br>.class<br>.jar |
| Procedure | Includes files that contain the executable functions of an application: report requests, WebFOCUS StyleSheets, and customized HTML called from WebFOCUS procedures using the Dialogue Manager command -HTMLFORM. For details about this command, see *Enhancing an HTML Web Page With a Procedure* on page 10-58. | On the path of the WebFOCUS Reporting server | .fex<br>.sty<br>.htm |
| Master File Access File | Includes all Master Files and Access Files. | On the path of the WebFOCUS Reporting server | .mas<br>.acx |

| File Type | Description | Location After Deployment | File Extension (NT/UNIX) |
|---|---|---|---|
| Data source | Includes all supported data sources types. | On the platform with the WebFOCUS Reporting Server or sub-servers | n/a |
| Temporary file | Includes data extracts, temporary files that your application creates during processing, and temporary work files that WebFOCUS uses internally. | In the EDA temporary directory (default) or a user-defined EDA location | .ftm or other requested extension |

## Distributing Processing Among Platforms

WebFOCUS applications are capable of distributing processing over many platforms, with the following advantages:

- **You can access data on multiple platforms**, thus forming relationships among disparate data sources.
- **Each component runs in the most suitable environment.** For example, a WebFOCUS HTML front-end may run on an NT or UNIX platform managed by local departmental administrators who authorize and group users, while the data access routines run on an OS/390 machine capable of securing the data and aggregating it quickly.
- **You can speed up your applications.** Procedures that access data can run on the platform where the data resides, thus ensuring that any aggregation or screening takes place immediately. This means that your application is not shipping large quantities of data across a network to be aggregated or screened somewhere else. Less network traffic means increased application speed.

For a fuller discussion of this topic see the WebFOCUS installation guide for your platform.

## Deploying a Partitioned Project

After you have developed a WebFOCUS reporting project that is logically partitioned into project components, you must deploy those components to the Web server and the WebFOCUS Reporting Server. Deployment is the process of copying or transferring the files to the appropriate server. All of your files must be placed in the proper paths so that the appropriate server can find them.

- If you are developing your project using the Developer Studio "develop and deploy" model (described in the *Application Development Getting Started* manual), the Deploy Wizard does the primary partitioning for you, locating components on their target servers. You can manually add types of files that the Deploy Wizard does not automatically partition. The wizard then copies the partitioned files to the Web server or the WebFOCUS Reporting Server. For details on the wizard, see *Step 2. Create a Deployment Scenario* in Chapter 12, *Partitioning and Deploying Project Files*.

  Developer Studio tools automatically include the path information required to locate and process the deployed files.

- If you have created your HTML files outside of the Developer Studio "develop and deploy" model, you can use an external tool such as FTP to transfer them to the appropriate server and platform. In this case, you will need to provide path information to ensure that these files can be located for processing. For details on choosing search path logic, see *Choosing Search Path Logic* on page 8-3.

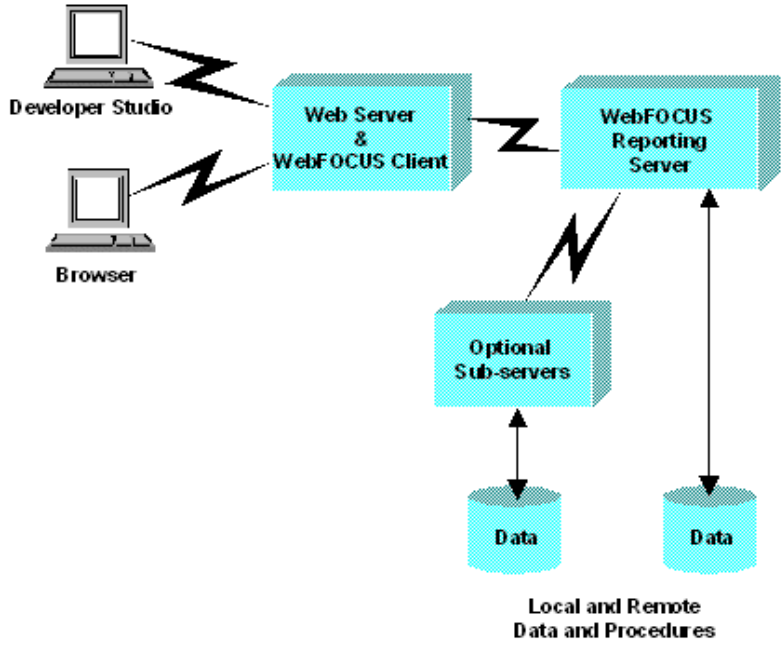
For details on search paths, see *Storing and Searching for Application Files* on page 2-12.

## WebFOCUS Components

To develop and deploy reporting projects to the Web, you need software components that manage the flow. A WebFOCUS system comprises several components, including:

- **Standard Web components.** A Web server and a Web browser that manage the display of Web pages.
- **WebFOCUS product components.** These include Developer Studio, and the WebFOCUS client and server components that manage processing of data and requests.
- **WebFOCUS application components.** These include user-created procedures and data.

More detailed descriptions of each component follow the diagram.




**Developer Studio**, the recommended WebFOCUS development environment, enables you to build, test, and deploy reporting projects using Windows®-based graphical tools, optionally supplemented by manual coding for maximum customization. Developer Studio tools handle a wide range of tasks, including report, graph, and form design.

The **Web browser** is responsible for displaying Web pages that may include reports, and graphs returned by the query after the project has been deployed. WebFOCUS works with Microsoft Internet Explorer.

The **Web server** handles requests by fetching HTML files from and returning them to the browser. When Web pages contain calls to the WebFOCUS client, the Web server launches the WebFOCUS client, which collects variables and sends the request to the WebFOCUS Reporting Server.

The **WebFOCUS client** resides on the Web server. It may be implemented as a Common Gateway Interface (CGI) program, an Internet Server API (ISAPI) program (for Microsoft IIS Web servers only), or as Java servlets; the choice is made at installation time. All options function identically to package the request for the WebFOCUS Reporting Server. WebFOCUS works with any standard Web server that supports the selected option. These include Netscape, Microsoft, IBM, and NCSA-compatible Web servers. Included with the WebFOCUS client is the iWay client, whose role is to locate the iWay Server on a specific host and port.

The **WebFOCUS Reporting Server** is responsible for accessing data, processing business logic, and generating fully styled output. It stores report procedures, WebFOCUS StyleSheets, and metadata (data source descriptions). You need one or more WebFOCUS Reporting Servers to deploy and run procedures and to access data sources.

**Optional WebFOCUS sub-servers** access data on remote platforms sources.

**Data**. A single server can access multiple types of data. For each type of DBMS, a data adapter must be installed.

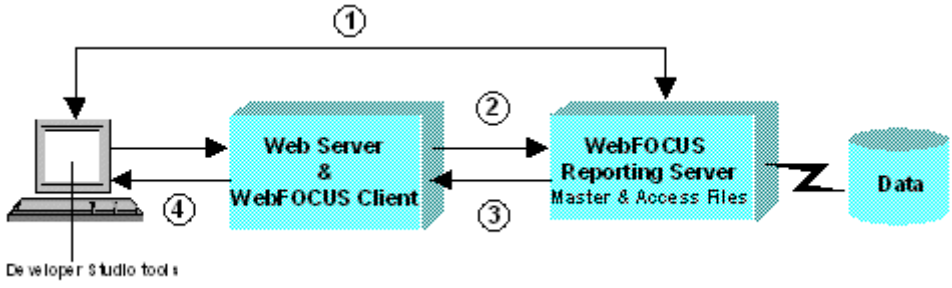
**Procedures**. Procedures can reside on a WebFOCUS Reporting Server or a sub-server. At runtime, the procedure requested by the WebFOCUS client is accessed by the WebFOCUS Reporting Server and executed. Procedures may call other procedures. Called procedures also reside on a WebFOCUS Reporting Server or sub-server.

Three additional diagrams will help you visualize how these components work together, in three integrated phases, to form a total development to deployment solution.

- *Phase 1: Develop and Test a Project* on page 2-9.
- *Phase 2: Partition and Deploy Project Components* on page 2-10.
- *Phase 3: Run Deployed Application From the Web* on page 2-11.

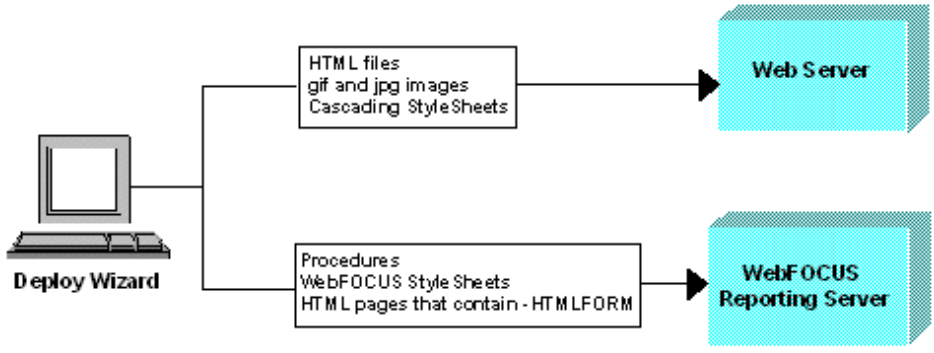
For a step-by-step overview of this process, see the *Application Development Getting Started* manual.

## *Reference* Phase 1: Develop and Test a Project




1. The application developer creates metadata (synonyms) for data sources on the server.
2. The application developer creates a project in Developer Studio, using graphical tools (supplemented by code, if desired). The project consists primarily of *reporting procedures* and *HTML forms* from which the procedures can be launched. The procedures and HTML files are stored on a server.
3. The developer runs and tests procedures and HTML forms. With each test, the request passes to a Web server, through the WebFOCUS client, to a WebFOCUS Reporting Server, which retrieves the data and processes the request. During the testing process, WebFOCUS uses the Master File and Access File on the server to interpret and access the data.
4. The WebFOCUS Reporting Server then sends the results back to the WebFOCUS client, which returns it to the Web server for display in Developer Studio's built-in browser.

## *Reference* Phase 2: Partition and Deploy Project Components



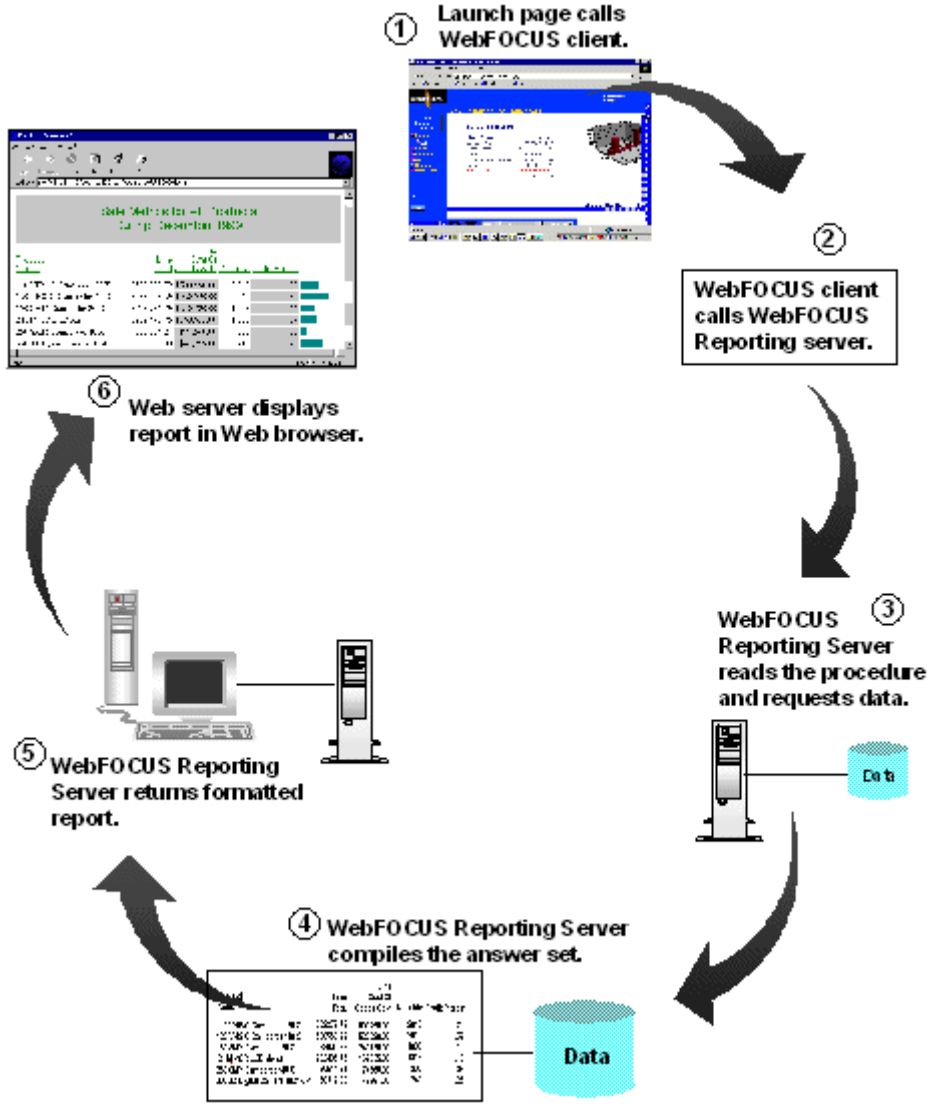

Once testing has been completed (in phase 1), the developer runs the Deploy Wizard, identifying the WebFOCUS Reporting Server that will receive the deployed application.

The wizard partitions the project's components between the Web server and the WebFOCUS Reporting server: it places the presentation logic on the Web server and the program logic on the WebFOCUS Reporting Server. The developer can manually add other project components, such as image files, class files, and Cascading Style Sheets to the appropriate server in the Deploy Wizard.

The wizard then copies, or deploys, the partitioned files to the designated server.

## *Reference* Phase 3: Run Deployed Application From the Web

If you used WebFOCUS before it was integrated with Developer Studio, you will recognize this as the same processing flow that has always applied when self-service applications are launched from a Web browser. What has been added is the ease of development and deployment described in phases 1 and 2.



1. Once the self-service application is deployed (in phase 2), a user opens a launch page in a Web browser and selects a report. The launch page, through the Web server, calls the WebFOCUS client. The request specifies the report procedure to be executed, as well as any parameters and values required by the procedure. Optionally, HTTP environment parameters set by the user's connection to the Web server may be passed to the procedure. The procedure name and all parameters are passed to the WebFOCUS client, which resides on your company's Web server.
2. The WebFOCUS client instructs the WebFOCUS Reporting Server to open an agent process for the request. The agent locates and executes the procedure.
3. The report procedure requests data from the data source (for example, Oracle). The data source may be local to or remote from the WebFOCUS Reporting Server. If it is remote, it is accessed through an intermediate sub-server, not shown in this simplified diagram.
4. The WebFOCUS Reporting Server compiles the answer set from the data source.
5. The WebFOCUS Reporting Server formats the answer set in the requested format (e.g., HTML, PDF, Excel 2000), and returns the report output to the Web server.
6. The Web server passes the report output to the user's browser. The browser displays the report in an HTML page or calls the appropriate desktop product, based on the file format, to display the output.

## Storing and Searching for Application Files

When you build your applications using Developer Studio tools, each project's files are stored in dedicated folders on the development server. When procedures are run against the WebFOCUS Reporting Server, the procedure code is sent across the wire to the Reporting Server for execution, but the code is not stored there.

The Reporting Server uses one of two methods to locate the target procedure: APP PATH or EDAPATH.

- APP PATH is a list of subdirectories beneath a common parent directory called APPROOT. APP PATH is the default search criteria for the WebFOCUS Reporting Server. The WebFOCUS Reporting Server searches each of the directories listed in APP PATH sequentially to find the target procedure. If APP PATH logic is enabled, it disables EDAPATH. APP PATH logic can be issued from a procedure, a profile, or from an HTML launch page.
- EDAPATH is a list of unrelated directories that the WebFOCUS Reporting Server searches sequentially to find the target procedure. EDAPATH logic can be issued from a reporting procedure or a profile.

You can use one of the following strategies when you build your applications:

- You can still use EDAPATH for your applications.
- Since the same server can simultaneously support both APP PATH and EDAPATH, you may choose to develop new applications using APP PATH, and keep your old application using EDAPATH.
- You can migrate all your applications from EDAPATH to APP PATH and use APP PATH as your new default.

For HTML files that are built with external graphical tools or coded manually in a text editor, you will need to decide whether to run the application with APP PATH or EDAPATH logic. You must make sure that application files are deployed to a directory included in EDAPATH or APP PATH, depending on which search path you plan to use.

- If applications are run with APP PATH logic, you must either add the appropriate variables (IBIAPP_*app*) to the HTML pages or add APP PATH logic commands to WebFOCUS procedures (.fex files). For details and illustrations, see *Choosing Search Path Logic* on page 8-3.
- If these applications run with EDAPATH, their directory must be included in the WebFOCUS Reporting Server's EDAPATH. The following sample syntax includes directories for various user added and WebFOCUS demo applications in EDAPATH:

```
    SET EDPATH=+D:\ibi\apps\ggdemo;-
1.  D:\ibi\apps\ncp;-
2.  D:\ibi\apps\centcorp\centdata;-
3.  D:\ibi\apps\centcorp\centfex;-
4.  D:\ibi\apps\template;-
5.  D:\ibi\apps\myapp1;-
6.  D:\user_directory\corpapp\myapp2;
```

where:

`lines 1-4`

Specify paths for directories containing sample data sources and applications for use within IB products.

`lines 5-6`

Are directories added to the search path to identify files associated with user-created applications.

### Deploy Wizard and APP PATH Logic

The Deploy Wizard in the Developer's Studio toolset uses APP PATH logic to create the necessary directories under the Web server and the WebFOCUS Reporting Server. Before you can deploy projects in Developer Studio, you must set APPROOT variables in two files:

- The APPROOT setting in the server's *edaserve.cfg* file controls the deployment of files to the WebFOCUS Reporting Server. This variable defines the path for deployed procedures. The WebFOCUS Reporting Server creates a subdirectory by appending the deployed application name to the directory path in the APPROOT setting.
- The APPROOT setting in the *CGIVARS.wfs* file controls the deployment of files to the Web server. This variable defines the path for deployed HTML files, graphic files, and other user-interface files. The WebFOCUS client creates a subdirectory by appending the deployed application name to the APPROOT setting.

  **Note:** You must also specify this APPROOT path as a virtual directory on the Web server. We suggest setting APPROOT to *install_drive/ibi/WebFOCUS52/ibi_html* because this is the default path defined as the alias on the Web server. Alternatively, you can set APPROOT to any other directory designated to hold the user interface files for a deployed application.

For complete information on configuring WebFOCUS to receive deployed applications, see *Summary of Steps* on page 12-2.

## Handicapped Accessibility in WebFOCUS

WebFOCUS Releases 4.3.6 and 5.2 provide handicapped accessibility that meets the requirements set by Section 508. WebFOCUS complies with accessibility requirements as a result of the following features:

- The ACCESSIBLE SET parameter generates report output in HTML code which supports products for handicapped accessibility. This HTML code automatically turns off pagination and column titles, page headings display only at the beginning of the report, and page footings are displayed only at the end of the report. For more information on this parameter, see Chapter 15, *Customizing Your Environment*.
- The SUMMARY WebFOCUS StyleSheet attribute maps reports and graphs to the HTML <TABLE SUMMARY> tag and places a description of the report or graph into a summary object inside the HTML table. For details on this attribute, see *Laying Out the Report Page* in the *Creating Reports With WebFOCUS Language* manual.
- The ALT WebFOCUS StyleSheet attribute is for use with a drill-down report you specify in a WebFOCUS StyleSheet. It maps to the HTML ALT tag on a hyperlink in a report. It can also adds descriptive text to an embedded image in an HTML report. For details on this attribute, see *Laying Out the Report Page* in the *Creating Reports With WebFOCUS Language* manual.

CHAPTER 3

# Creating a Reporting Application

**Topics:**

- What Is a Reporting Application?
- Supported Environments
- Creating a Project
- Organizing a Project
- Adding Master Files to a Project
- Viewing and Modifying Project Properties
- Tracking the Code for a Project
- Securing a Project
- Removing a Project

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.

## 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.

## Supported Environments

Developer Studio supports two environments for local development:

- In the first environment, Developer Studio, the WebFOCUS Reporting Server, and the Web server are installed on the same machine. This WebFOCUS environment is referred to as the *localhost*. You can choose to install the WebFOCUS Reporting Server during installation of Developer Studio. In this environment, the project files that you create reside in a subdirectory on the WebFOCUS Reporting Server under APPROOT.
- In the second environment, you can use a remote WebFOCUS installation for local development. Developer Studio is installed on a local machine, and the WebFOCUS components (which include a Reporting Server, WebFOCUS Client, and Web server) are installed on remote machines.

Use of a remote environment for local development requires the following:

- The WebFOCUS Reporting Server must be at a supported release level.
- The WebFOCUS Reporting Server must be installed on a supported Windows platform.
- The WebFOCUS Reporting Server and the Web server must be properly configured. For example, the APPROOT variable must be set as required on both servers. APPROOT on both servers must point to the same path.

  For more information on APPROOT, see Chapter 12, *Partitioning and Deploying Project Files*.
- You must have read and write access to the WebFOCUS Reporting Server and the Web server.

## Creating a Project

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 will contain the associated files. You create this directory with 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, and Procedures. (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 3-9.

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 *WebFOCUS 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.

## *Reference* File Types

The three 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 Chapter 5, *Understanding User Interface Basics*. 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 Chapter 4, *Creating a Reporting Procedure*.

**Tip:** By default, certain file types are not associated with a virtual folder. An example is a .gif file. In the Explorer, these file types are displayed in the right pane as separate entities. However, you can organize your project in such a way that they are included in folders. For details, see *Organizing a Project* on page 3-9.

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

| File Type | Description |
|---|---|
| Procedure | Includes files that contain the executable functions of a project, such as procedures (.fex). |
| File description | Includes Master Files (.mas) and Access Files (.acx). |
| Data source | Includes files that contain data values. |
| HTML resource | 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). |
| Temporary data file | Includes temporary hold files (.ftm) that Developer Studio creates during processing, and a number of temporary work files used internally. |

## *Procedure* How to Create a Project Directory

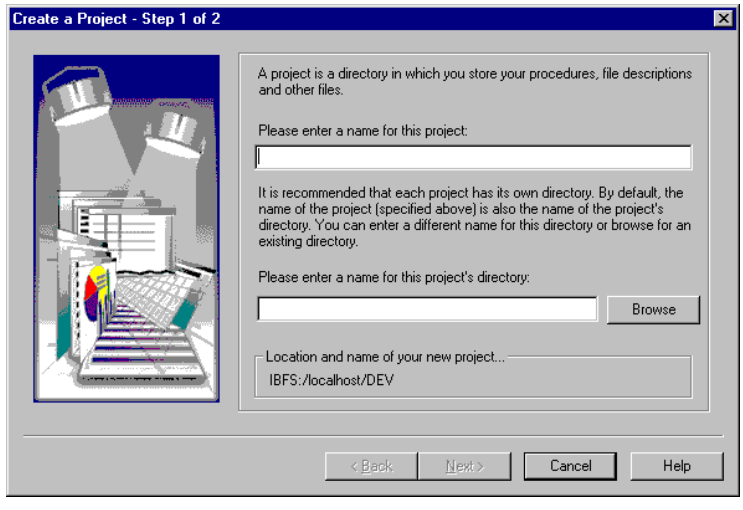
1. Do one of the following:

   - Right-click the *Projects on localhost* node and choose *New Project*.

     or

   - 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).

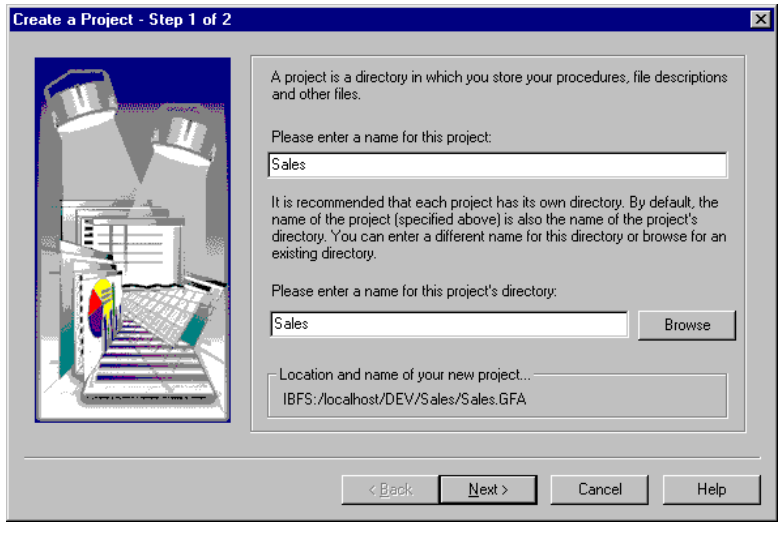
**3.** The Project Wizard displays the project name as the default directory in the second input field.

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.

**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.

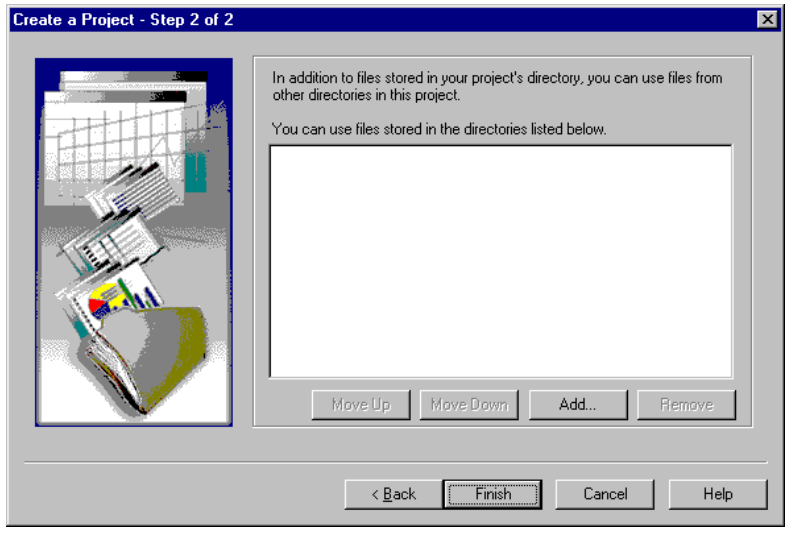
The following is a completed dialog box:




**5.** Click *Next* to continue.

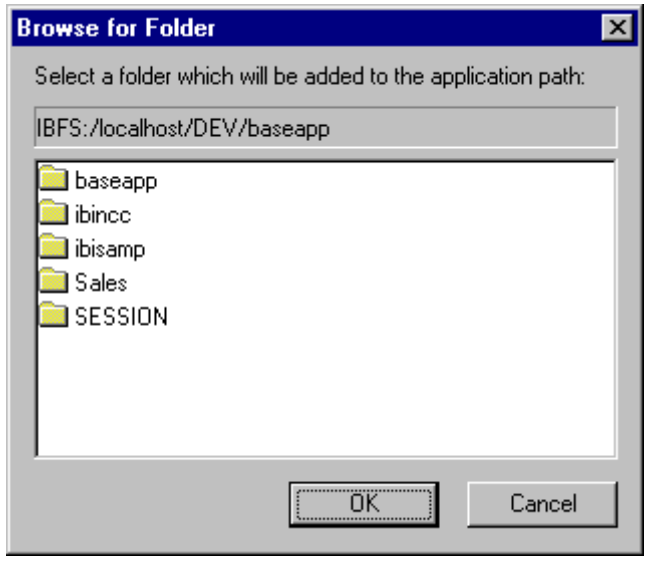
**6.** A message is displayed if the directory does not exist. Click *Yes* to create the new directory. Click *No* to change the name.

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.




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 3-14.

# Organizing a Project

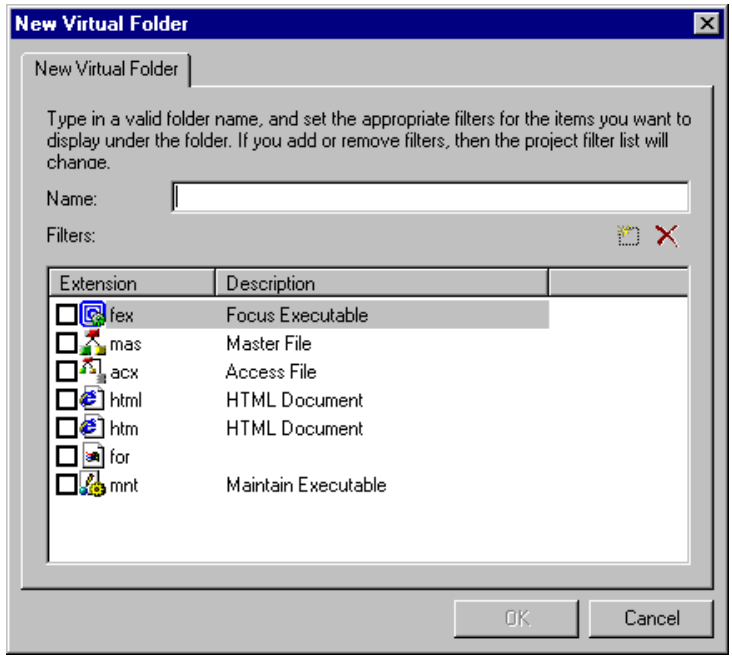
Developer Studio has many features that help you organize a project:

- 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.
- You can customize the file types that display in any folder using the Project Properties dialog box. For details, see *How to Customize Display of File Types* on page 3-19.
- 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.

## *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 shortcut menu, and click *Virtual Folder*.

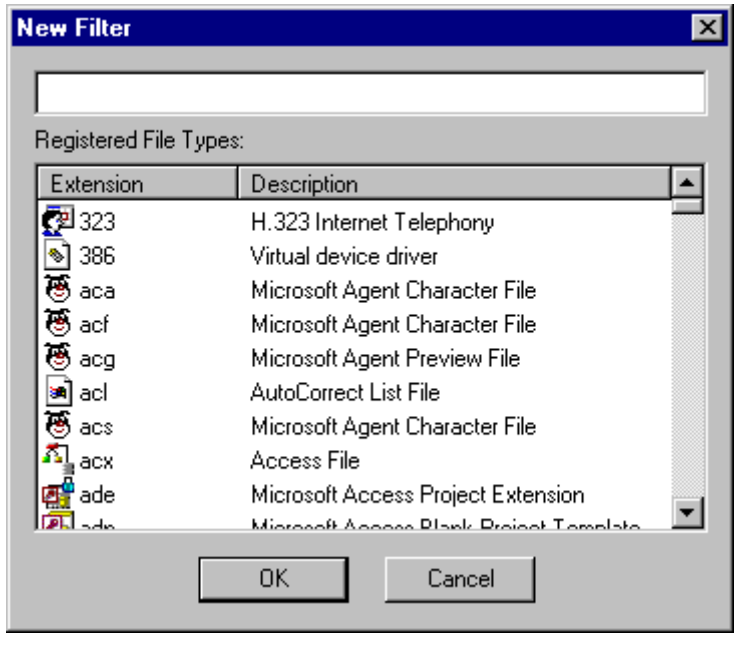
   The New Virtual Folder dialog box opens:



**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:



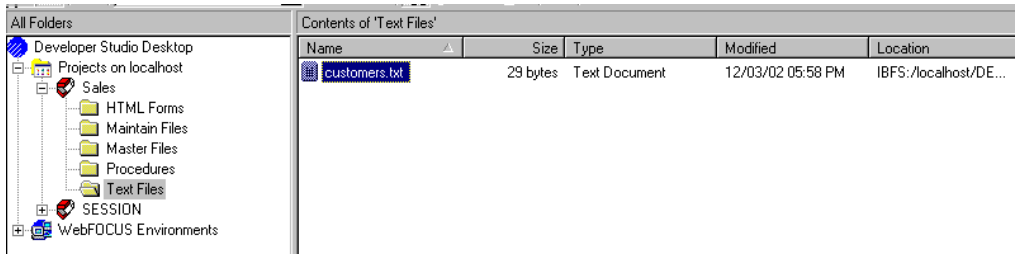
3. Scroll through the list and select one or more file types. Click *OK* when you are done.

   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 .txt files used by the project named Sales:




## *Procedure* How to Add a Virtual Subfolder to a Folder and Customize File Types

1. Select and right-click the folder. Select *New* from the shortcut menu, and click *Virtual Folder*.

   The New Virtual Folder dialog box opens.

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

   Customize the list of file types that will be displayed in the subfolder. 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 subfolder.

   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.

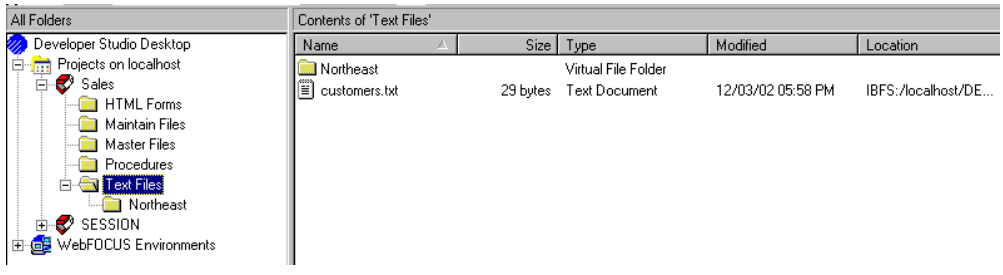
3. Scroll through the list and select one or more file types. Click *OK* when you are done.

   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 subfolder named Northeast:




### *Procedure* How to Rename a Component

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

## Adding Master Files to a Project

Before you create a procedure that reports against a data source, your project must understand how the data is organized. To obtain this information, a project reads a data source description. Developer Studio refers to these descriptions as Master Files and Access Files:

- A Master File describes the structure of the data.
- An Access File supplements the Master File for some data sources. An Access File includes additional information that completes the description of a data source, for example, the full data source name and location. If your data source is local and is either FOCUS or text format, an Access File is not necessary.

There are three 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 3-13 for details.

- **Create synonyms.** Using the Server Console or the Create Synonym Wizard, 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.

- **Create a Master File from scratch.** Intended primarily for use by a database administrator, the Master File Editor can be used to create a data source description from scratch. An application developer can use this tool to view the structure of a data source to make minor, authorized modifications to the Master File. 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 (Displays all files in the project path) 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 (Displays all files in the project path) 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 button (Removes the selected items from the project) 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.

## Viewing and Modifying Project Properties

You can view and modify project properties to:

- **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.
- **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.
- **Run a project on a remote server.** You can run a project on a server other than the localhost.
- **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.

You can also view project properties such as file type, location, size, and date of last modification.

*Procedure* **How to View Project Properties**

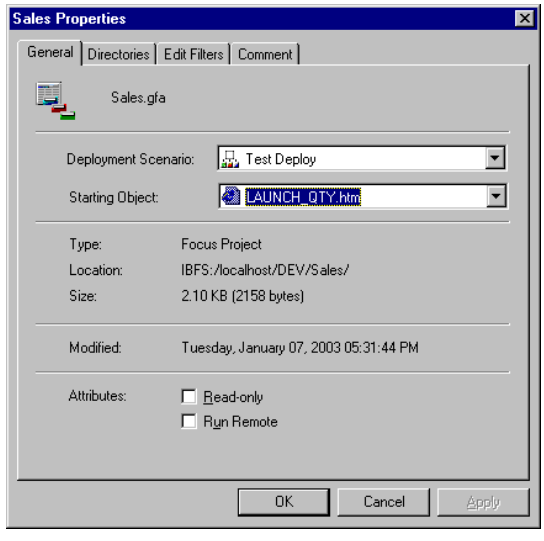
1. Right-click a project and choose *Properties*.
2. The Project Properties dialog box opens, displaying the General, Directories, Edit Filters, and Comment tabs.
3. The General tab is selected by default. It displays the project type, location, size, and date of last modification.

   You can modify certain project properties using this dialog box.

## *Procedure* How to Select a Deployment Scenario

1. Right-click a project and select *Properties*.

   The Project Properties dialog box opens.




2. On the General tab, click the Deployment 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).

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 Select a Starting Object

1. Right-click a project and choose *Properties*.

   The Project Properties dialog box opens.

2. On the General tab, 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.

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.

   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 *Starting*, or by highlighting it and selecting *Starting* from the File menu.

## *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 Run a Project on a Remote Server

1. Right-click a project and choose *Properties*.

   The Project Properties dialog box opens.

2. On the General tab, click the *Run Remote* check box.

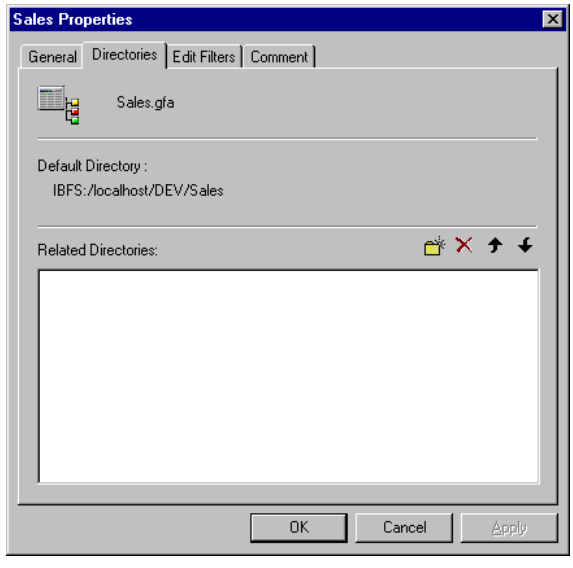
3. Complete the dialog box as described in this topic. Click *OK* to apply changes and close the dialog box, or click *Apply* to temporarily apply changes and keep the dialog box open. Click *Close* to apply the temporary changes.

## *Procedure* How to Modify Directory Paths

1. Right-click a project and select *Properties*.

   The Project Properties dialog box opens.

2. Click the *Directories* tab:




   Make the following modifications to the directory paths:

   - To add a new directory path, click (Add new directory). 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 (Remove directory). Click *Yes* to confirm the deletion.
   - To reposition a directory path, highlight it and click (Move Directory Up) or (Move Directory Down). These buttons are inactive for a single directory.

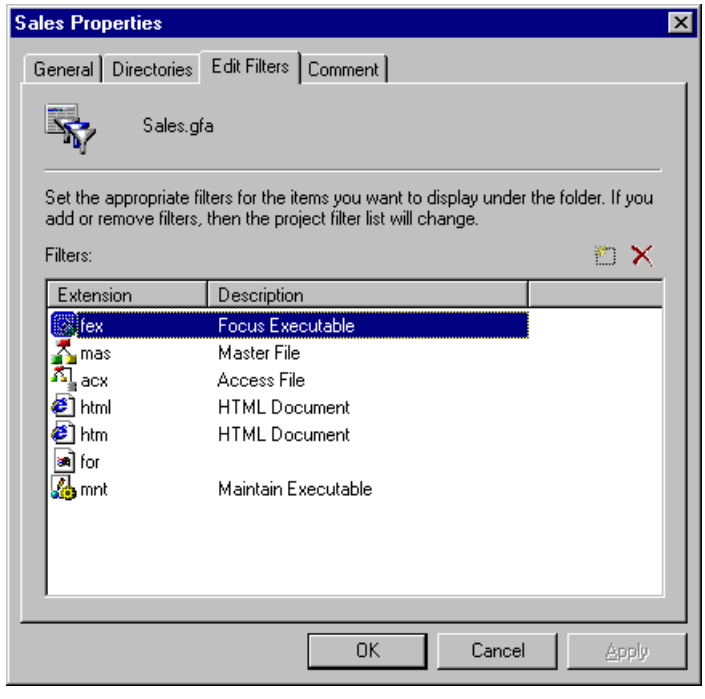
3. Complete the dialog box as described in this topic. Click *OK* to apply changes and close the dialog box, or click *Apply* to temporarily apply changes and keep the dialog box open. Click *Close* to apply the temporary changes.

## *Procedure* How to Customize Display of File Types

1. Right-click a project and choose *Properties*.

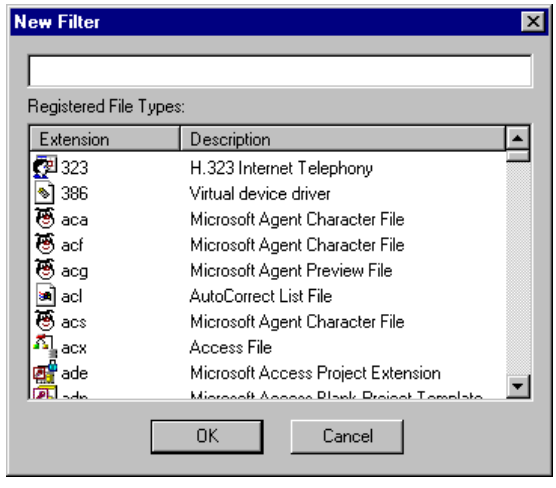
   The Project Properties dialog box opens.

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 (Add new file type filters). 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 (Remove file type filters). 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.

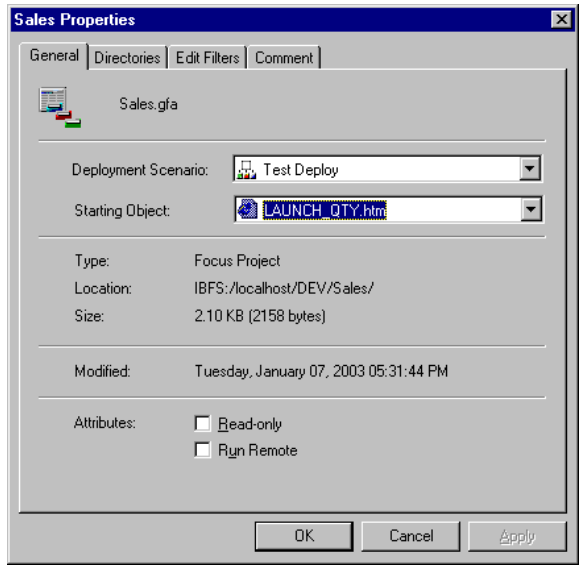
2. Click the *Comment* tab:

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.

## *Reference* Project Properties Dialog Box

### General Tab




**Deployment Scenario**

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

**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.

**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.

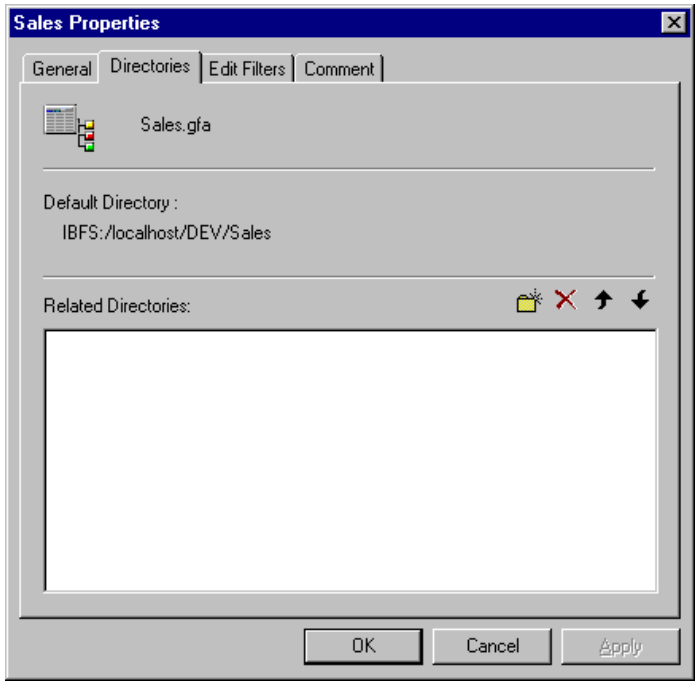
**Cancel**

Aborts changes and returns to the Explorer.

**Apply**

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

**Directories Tab**




**Related Directories**

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 Button**

Adds a directory to the project search path.

**Remove Directory Button**

Deletes a directory from the project search path.

**Move Directory Up Button**

Moves a directory up in the project search path.

**Move Directory Down Button**

Moves a directory down in the project search path.

**OK**

Confirms changes and returns to the Explorer.

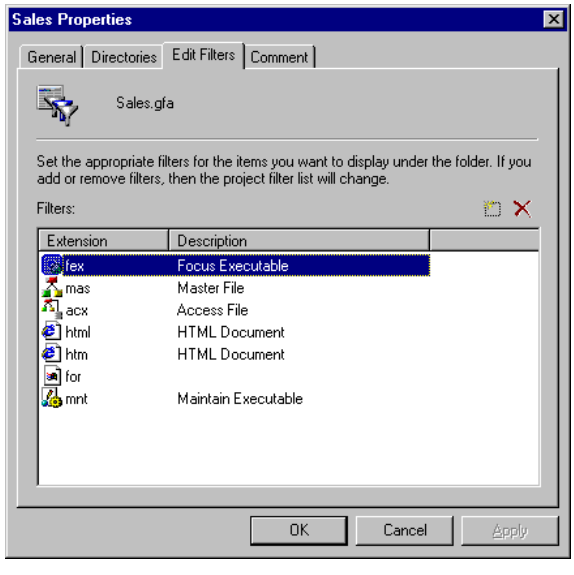
**Cancel**

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) Button**

Adds a filter.

**Remove File Type Filter(s) Button**

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.

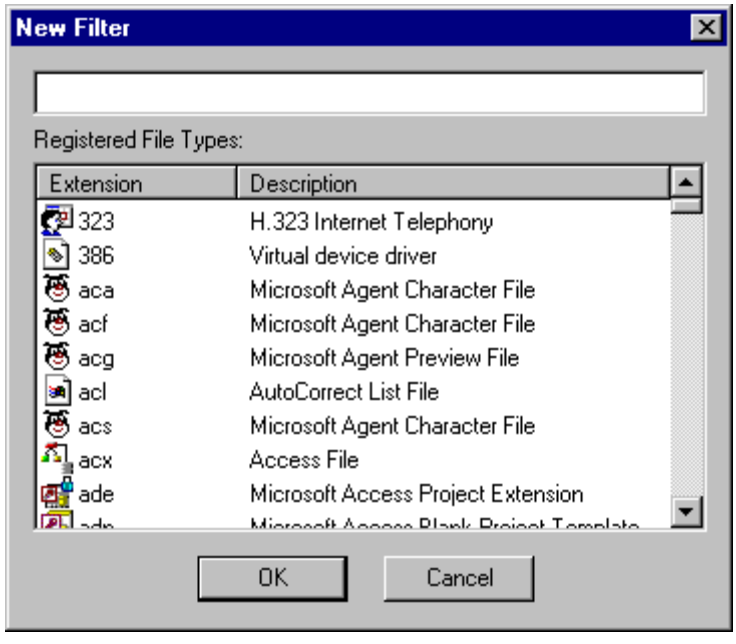
**Cancel**

Aborts changes and returns to the Explorer.

**Apply**

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

## *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.

## Tracking the Code for a Project

Developer Studio supports third-party source file management products such as Microsoft Visual SourceSafe, PVCS Version Manager from Merant, and others that use an industry standard API. Source file management provides version control for development teams, allowing multiple users of Developer Studio to work on the same project without overwriting each other's modifications. A source file management product typically controls access to source code and documents all code changes.

If your site has installed a source file management product, you can track the code for a project from the Explorer through an audit trail of changes.

## Securing 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

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 shortcut 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.

CHAPTER 4

# Creating a Reporting Procedure

**Topics:**

- Selecting a Creation Tool
- Incorporating a Procedure Into an Application
- Copying a Procedure
- Creating a Procedure Component
- Working With a Component
- Calling a Procedure From the Current One
- Managing Flow of Control
- Working With a Full Procedure
- Running a Procedure
- Canceling a Running 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.

## Selecting a Creation Tool

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

- **Report Painter.** Provides a graphical environment for the creation of a report procedure.
- **Report Layout Painter.** Provides a graphical environment for the creation of a procedure and HTML form in one integrated process.
- **Procedure Viewer.** Provides a graphical environment for the creation of a generic procedure.
- **Text editor.** Provides a standard text editor, allowing you to create a procedure by entering commands.
- **Compound Document Tool.** Provides a graphical environment, called the Compound Report Constructor, which allows you to combine reports with styled hold formats (for example, PDF, HTML, PS, EXL2K, and so on) or regular character-based reports into one PDF file.

## *Procedure* How to Select a Creation Tool

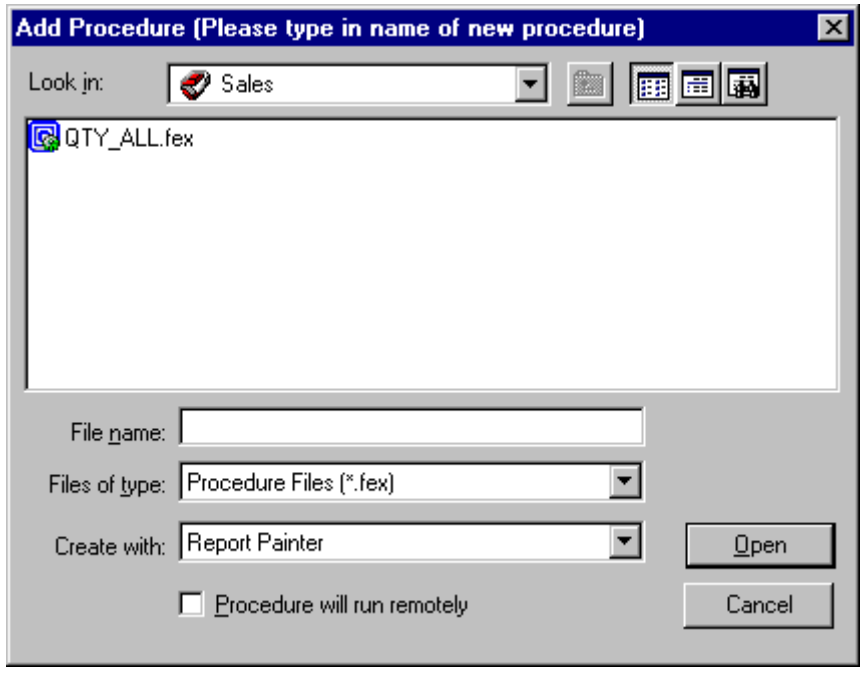
1. Open a project. Then do one of the following:

   - 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:




   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.

2. In the File name field, type a name for the new procedure; do not include an extension. Developer Studio automatically appends the extension .fex.

   The name can be up to 66 characters in length.

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

**4.** In the Create with drop-down list, select the creation tool:

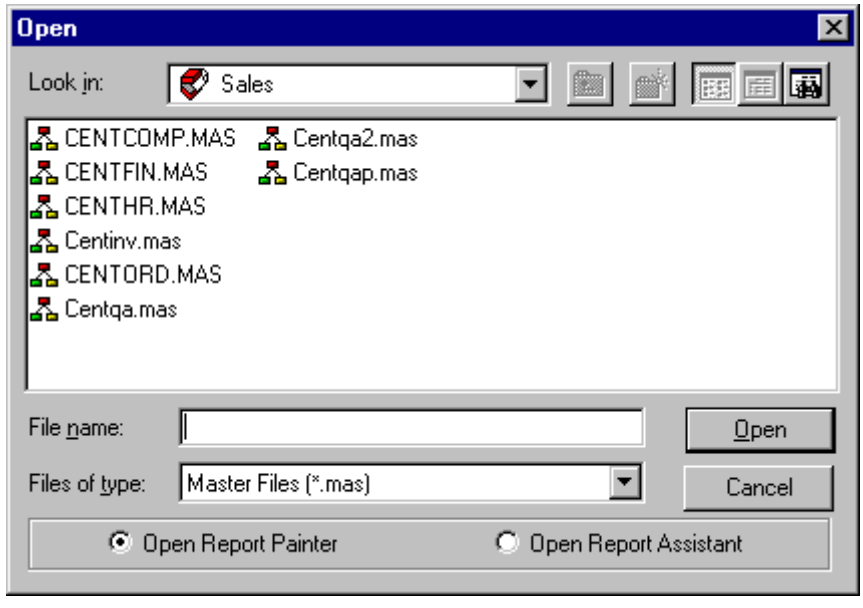
- *Report Painter* provides a graphical environment for the creation of a report procedure. For information, see *How to Create a Report Procedure* on page 4-5.
- *Report Layout Painter* provides a graphical environment for the creation of a procedure and HTML form in one integrated process. When you save the layout, both the procedure file (.FEX) and the HTML file (.HTM) are stored in their respective project folders. You can deploy both files with the project; they will retain their relationship.
- For information see Chapter 6, *Designing a User Interface for a Web Application With the Report Layout Painter*.
- *Procedure Viewer* provides a graphical environment for the creation of a generic procedure. For information, see *How to Create a Reporting Procedure* on page 4-7.
- *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 4-8.
- *Compound Document Tool* provides a graphical environment, called the Compound Report Constructor, which allows you to combine reports with styled hold formats (for example, PDF, HTML, PS, EXL2K, and so on) or regular character-based reports into one PDF file.

  For information, see the *Creating Reports With Graphical Tools* manual.

**5.** Optionally, click *Procedure will run remotely* to run the procedure on a remote server. You can also designate a procedure to run remotely after you have created it. See *How to Run a Procedure on a Remote Server* on page 4-28.

## *Procedure* How to Create a Report Procedure

1. In the Add Procedure dialog box, enter a name for the new procedure, select *Report Painter,* and click *Open*. The Open dialog box prompts you for the name of the Master File to use:



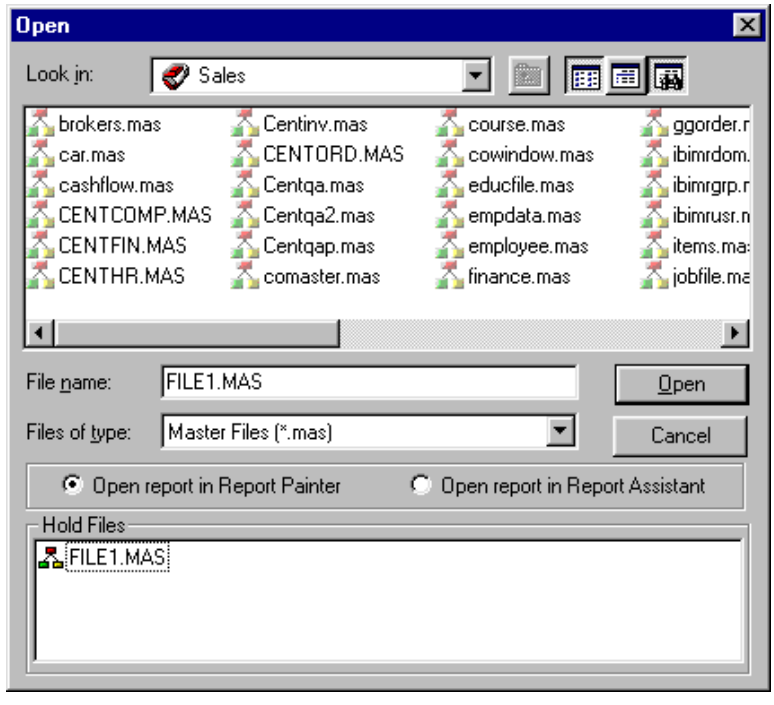

   The display box lists all the Master Files currently visible in the project path, or all the Master Files that have been added to the project path. Click the *Displays all files in the project path* button to toggle between the two displays.

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 either *Open Report Painter* or *Open Report Assistant,* and click *Open* to start the selected tool.

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

   **Note:** If you create a Hold file for a report procedure, you can select this file from the Open dialog box. To select a Hold file, click and hold a component connector from the Procedure window and select the Report tool. The Open dialog box displays:




5. Select the Hold file and click *Open*. Report Painter opens with the Hold file that you selected.

## *Procedure* How to Create a Reporting Procedure

1. On the Add Procedure dialog box, enter a name for the new procedure, select *Procedure Viewer*, and click *Open*.

   The Procedure window opens:

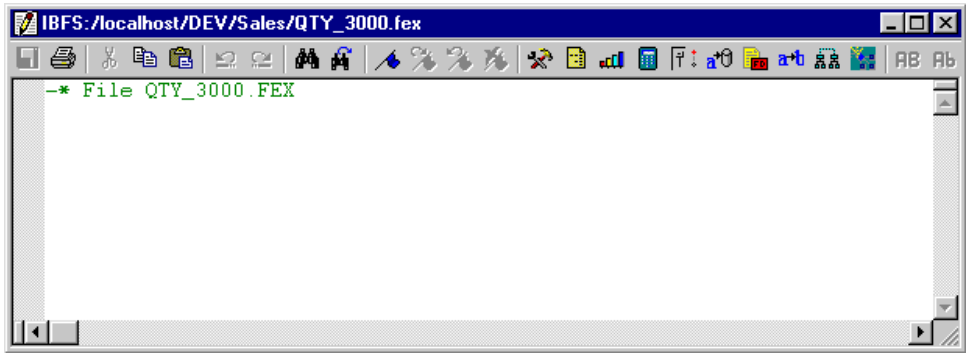
   This window contains a default Comment component with two diamond objects on either side called component connectors. The Comment component does not contain any executable commands. At this point, you have created a procedure shell. You can now make this shell executable by adding components, such as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

   Components are stored in a file with the extension .FEX, including any information on running the procedure remotely.

2. Left-click 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 4-13.

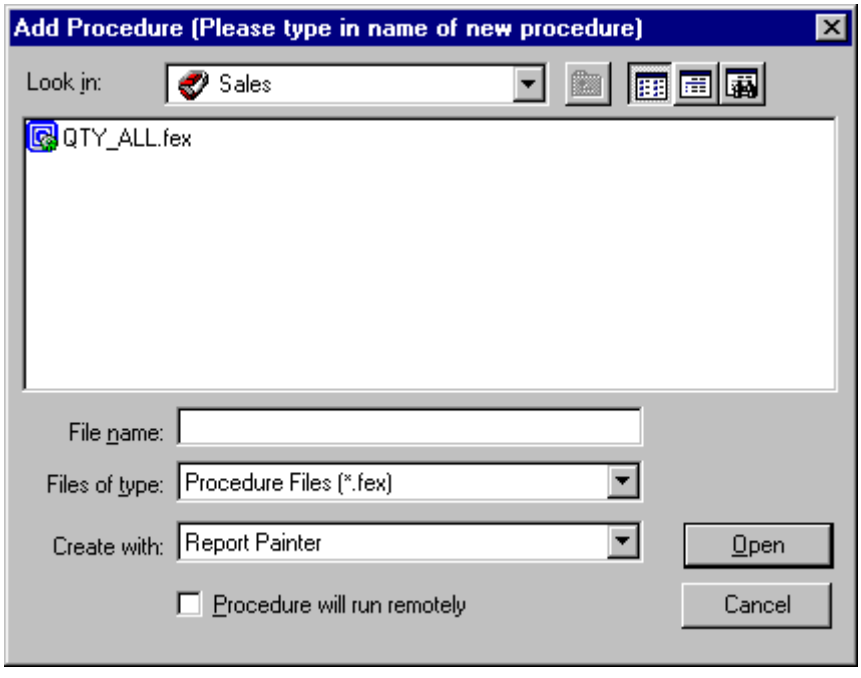
## *Procedure* How to Create a Procedure With the Text Editor

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




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 Chapter 18, *Editing Application Components as Text in Developer Studio* for details on the text editor.

## *Reference* Add Procedure Dialog Box




**Look in**

Is the name of the project. Select the name from the drop-down list.

**Procedure List Box**

Lists the procedures associated with the selected project.

**File name**

Is the name of the procedure. Enter a name in this field.

**Files of type**

Is the type of file. Accept the default—Procedure Files (*.fex).

**Create with**

Select the procedure creation tool from the drop-down list. The options are:

**Report Painter** provides a graphical environment for the creation of a report procedure.

**Report Layout Painter** provides a graphical environment for the creation of a procedure and HTML form in one integrated process.

**Procedure Viewer** provides a graphical environment for the creation of a generic procedure.

**Text editor** provides a standard text editor, allowing you to create a procedure by entering commands.

**Compound Document Tool** provides a graphical environment, called the Compound Report Constructor, which allows you to combine reports with styled hold formats (for example, PDF, HTML, PS, EXL2K, and so on) or regular character-based reports into one PDF file.

**Procedure will run remotely**

Click this check box to run the procedure on a remote server.

## Incorporating a Procedure Into an Application

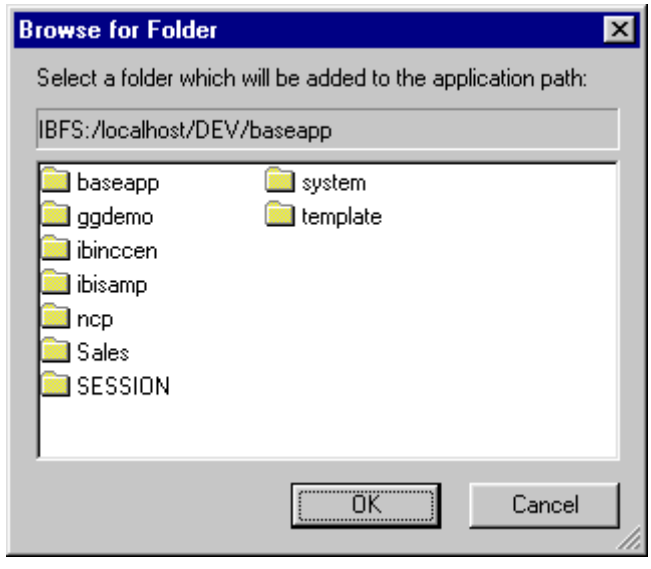
When you create a procedure in the current project, it is automatically incorporated into that project. However, you are not limited to the use of procedures created in the project. You can make procedures from another project available.

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

- **Add the procedure to the project path.** When you create a project with the Project Wizard, you define the project path. Any procedure in a project path is visible to the project.
- **Add a procedure to the project.** Once a procedure is visible to a project, add it to the project so it will be available for reporting and deployment.

## *Procedure* How to Add a Procedure to the Project Path

1. Right-click the project and choose *Properties*, then select the *Directories* tab.
2. Click the *New* button to display the Browse for Folder dialog box.
3. From the Browse for Folder dialog box, select a folder to add to the project path. The dialog box displays only those folders stored on the APPROOT.




4. 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 Directories tab, 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 on the Directories tab, 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 *Apply* on the Directories tab.

## *Procedure* How to Add a Procedure to a Project

1. Open the Procedures folder in the project.

2. 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 grey icon next to them, and the file type description is Focus Executable Visible within Project Path.

3. 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 description 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. 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. In the right window pane, right-click a procedure and choose *Remove from Project* from the shortcut menu.

   or

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

   Click *Yes* to confirm permanent removal of the procedure.

## Copying a Procedure

You can drag and drop any component in a project, including a procedure, between projects or Explorer environments.

You can also copy and paste a procedure within a single project. The copied procedure 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 Single Project

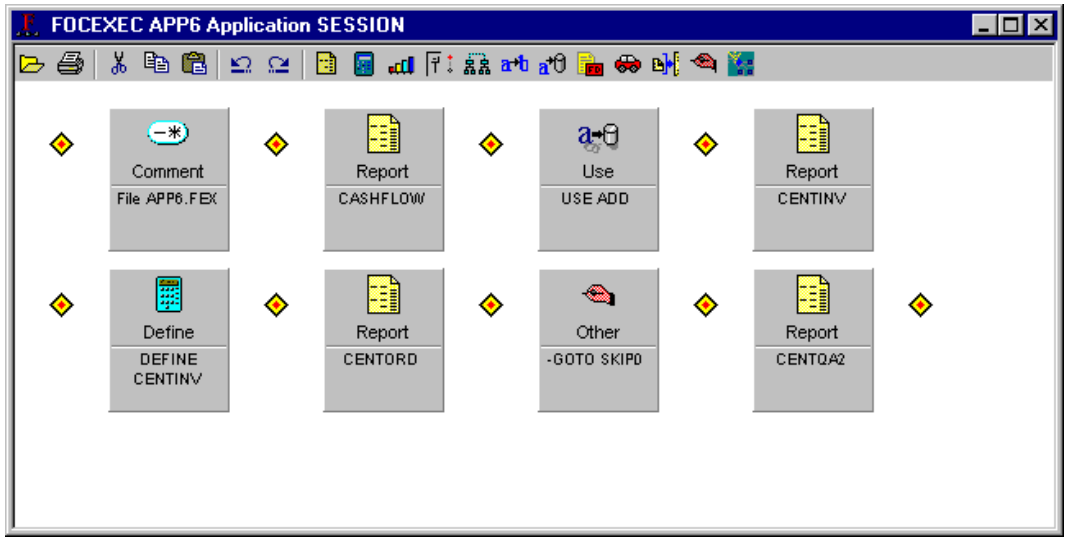
1. Open the Procedures folder for a project.
2. Right-click the procedure to be copied, and select *Copy* from the shortcut menu.
3. Right-click the Procedures folder, and select *Paste* from the shortcut menu.

## Creating a Procedure Component

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—Let, Set, Filedef, 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, a combination, or all of these components in a procedure. The following example includes a Comment, four instances of a Report, a Use, Define, and Other. Together these components make up the procedure named APP6.




## *Reference* Component Types

You can include the following components in a procedure.

| Component | Description |
|---|---|
| **Report** | Creates a tabular representation of the data retrieved from a data source.<br><br>A Report is placed after a Join and a Define. For details, see the *Creating Reports With Graphical Tools* manual. |
| **Define** | Creates a temporary (virtual) field that is evaluated before a report is run.<br><br>A Define is placed before a Report. For details, see the *Creating Reports With Graphical Tools* manual. |
| **Graph** | Creates a graphical representation of the data retrieved from a data source.<br><br>A Graph is placed after a Join and a Define. For details, see the *Creating Reports With Graphical Tools* manual. |

| Component | Description |
|---|---|
| **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.<br><br>For details on the Set tool and parameters, see Chapter 15, *Customizing Your Environment*. |
| **Join** | Defines a relationship between two or more data sources so that a report can use data from all of them at once.<br><br>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. |
| **Let** | Defines a word to represent another string of text. This feature saves time when you are writing code for local use. However, you must delete any Let components before you deploy a project to the Web.<br><br>For details on the Let tool, see the archive information on the Documentation CD. |
| **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).<br><br>For details on accessing FOCUS data sources, including techniques for maintaining Use commands, see Chapter 14, *Accessing a FOCUS Data Source*. |
| **Filedef** | Assigns a temporary name to a file created and used by WebFOCUS.<br><br>To preserve an extract file (created by the HOLD command) on the server's disk, you must issue a FILEDEF 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.<br><br>For details on the Filedef tool, including techniques for maintaining Filedefs, see Chapter 16, *Defining and Allocating WebFOCUS Files*. |

| Component | Description |
|---|---|
| **Execute** | Enables one procedure to call another. You can select a procedure to be called by the one you are editing.<br><br>An Execute component can be placed anywhere in a procedure.<br><br>For details on the Execute component, see *Calling a Procedure From the Current One* on page 4-22. |
| **Include** | 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.<br><br>An Include component can be placed anywhere in a procedure.<br><br>For details on the Include component, see *Calling a Procedure From the Current One* on page 4-22. |
| **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.<br><br>An Other component can be placed anywhere in a procedure.<br><br>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. |

| Component | Description |
|---|---|
| **Dimension** | 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.<br><br>A Dimension component applies to FOCUS data sources and relational tables. It must be placed before the Report component.<br><br>For information on defining dimensions, see the *Describing Data With Graphical Tools* manual and the *Describing Data With WebFOCUS Language* manual.<br><br>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. |
| **Modify** | 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.<br><br>For details on Modify, see the archive information on the Documentation CD. |
| **Comment** | 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.<br><br>For information on the Comment component, see the *Developer Studio Application Development Getting Started* manual. |

### *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. Left-click a component connector and hold it down.

   The Component Connector toolbar opens:

   You can also select a tool from the static Component Connector toolbar that displays in the Procedure window.
3. Select the tool for the component you wish to create.

   The corresponding tool opens. For details, see *Component Types* on page 4-14.

   Once you have constructed a component using a tool, exit the tool. You are asked if you want to update the procedure. 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

Tasks that you can perform on a component are:

- **Running.** Running some types of components, for example, a report or graph, shows you how the output appears from a visual point of view. For other components, such as a Define, running produces no visible output, but validates the structure against the server and allows the use of the result in the procedure.
- **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 4-20.
- **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 a project folder.
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.

**Tip:** If you are running a report flagged for remote execution from a reporting tool (Report Painter, Report Assistant, Graph Assistant), the report will automatically run remotely. To run a report remotely when it has not been flagged for remote execution, select *Run Remote* from the File menu.

## *Procedure* How to Edit a Component

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

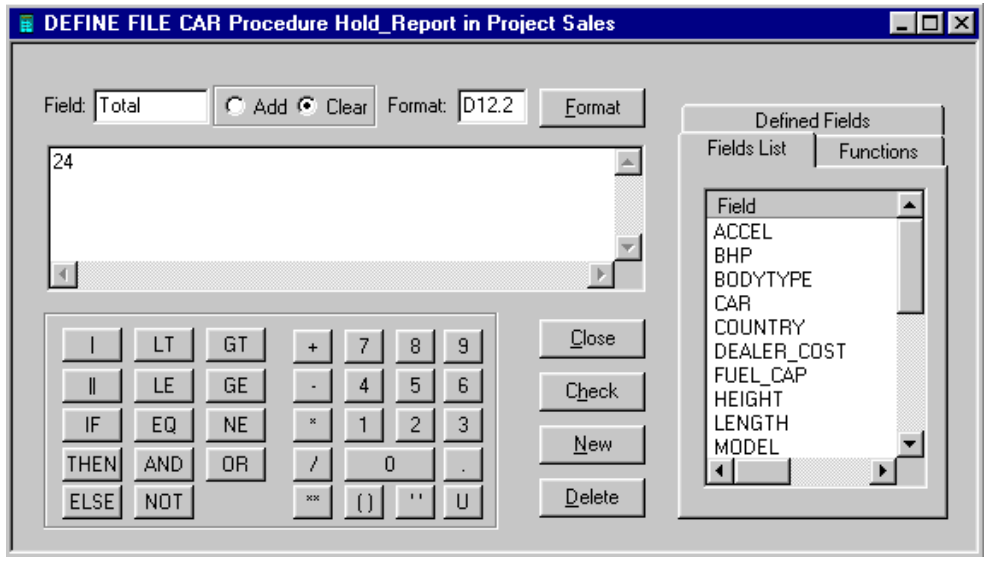
     or
   - Right-click the component and choose *Open*.
2. Make your changes to the selected component.
3. Exit the tool by clicking the control menu box. When asked if you want to update the procedure, click *Yes*. Your changes are saved.

**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 Chapter 18, *Editing Application Components as Text in Developer Studio*.

## *Procedure* How to Check a Component

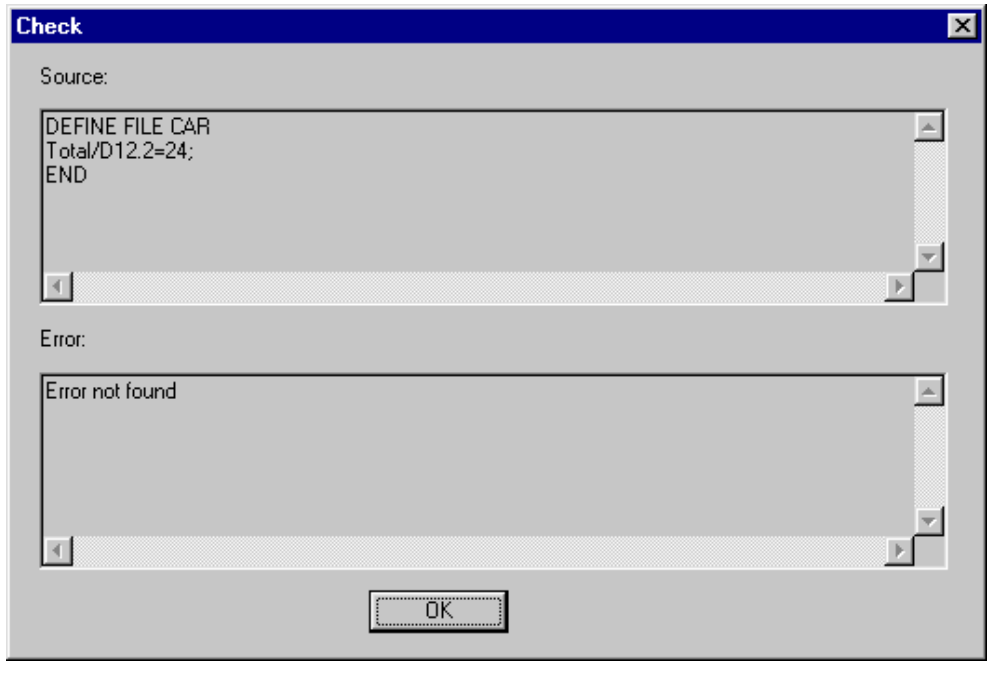
1. From the Procedure window, choose one of the following component tools:
   - Define
   - Set
   - Let
   - Use
   - FileDef.

   For example, if you choose Define, the following window opens:



**2.** Enter your Define criteria and click the *Check* button.

The Check window opens:




The source code for your Define and any errors that may exist are displayed.

**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 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, drag the mouse pointer to the second Procedure window.
5. Release the mouse button when the mouse pointer is over the second Procedure window.

   The component is added 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 using the same method, except that you must press and hold *Ctrl* before clicking and dragging the component you want to move.

## Calling a Procedure From the Current One

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

The Execute component allows one procedure to execute or call another procedure. In this case, the called procedure behaves as a completely separate procedure, with its own context. It cannot use any local variables (&variables) defined by the calling procedure unless they are explicitly passed to the called procedure on the command line. However, the executed (called) procedure can use any global variables (&&variables) defined in the calling procedure.

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. The Include object can be embedded up to four levels deep.

## *Procedure* How to Execute a Procedure From the Current One

1. Open the host procedure.
2. Click and hold the component connector at the point at which to include the called procedure.

   The Component Connector toolbar opens.
3. Click the *Execute* button.

   The Open dialog box opens.
4. In the display box, select the name of the called procedure. The name appears in the File name field.

   Instead of selecting an existing procedure, you can type the name of a new procedure in the File name field. Developer Studio opens a new Procedure window, from which you can create the called procedure from scratch.
5. Click *Open*.

   The Procedure window for the called procedure opens.
6. After viewing, editing, or creating the called procedure, close it.

   The Execute object is added to the host procedure to represent the called procedure.
7. To run the called procedure, right-click the *Execute* object and choose *Run*.

**Tip:** The Execute component inserts the EX command in the procedure's code. Right-click the *Execute* object and choose *Quick View* to see the EX command.

## *Procedure* How to Include Another Procedure in the Current One

1. Open the host procedure.
2. Click and hold the component connector at the point at which to include the existing procedure.

   The Component Connector toolbar opens.
3. Click the *Include* button on the toolbox.

   The Open dialog box is displayed.
4. In the list box, select the name of the called procedure. The name appears in the File name field.

   Instead of selecting an existing procedure, you can type the name of a new procedure in the File name field. Developer Studio opens a new Procedure window, from which you can create the called procedure from scratch.

5. Click *Open*.

   The Procedure window for the called procedure opens.

6. After viewing, editing, or creating the called procedure, close it.

The Include object is added to the host procedure to represent the called procedure.

You can embed procedures up to four levels deep.

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

## 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. Stored 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 Chapter 10, *Managing Flow of Control in an Application*.

## Working With a Full Procedure

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

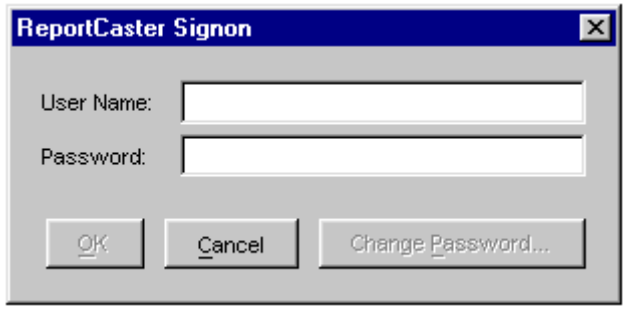
- **Activating OLAP.** If a procedure is set to run in HTML format and it uses an OLAP-enabled Master File, you can activate OLAP for that procedure and use the OLAP interface to manipulate data in a report or graph. For details on OLAP, see the *Creating Reports With Graphical Tools* manual.
- **Scheduling 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 4-26.
- **Create a Web-based user interface.** You can use the Report Layout Painter or Microsoft® FrontPage 2000 to create a launch page (Web-based interface) that makes a report available to browser users. For more information on your options, see Chapter 5, *Understanding User Interface Basics*.
- **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 4-26.
- **Editing a procedure.** You can edit a procedure as text. See *How to Edit a Procedure as Text* on page 4-26.
- **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. See *How to Debug a Procedure* on page 4-27.

## *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:




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

From an open project:

- To encrypt, right-click a procedure in the Procedures folder, then select *Properties*. On the Project Properties dialog box, click the *Encrypted* check box and click *Apply*.
- To decrypt, right-click a procedure in the Procedures folder, then select *Properties*. On the Project Properties dialog box, click the *Encrypted* check box to remove the check mark and click *Apply*.

## *Procedure* How to Edit a Procedure as Text

1. 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.
2. The text editor opens, displaying the procedure's syntax.

For details on using the text editor, see Chapter 18, *Editing Application Components as Text in Developer Studio*.

### *Procedure* How to Debug a Procedure

You can debug a procedure if you encounter any errors. To debug a procedure at any time, you must use the Web Console. For more information, see Chapter 11, *Testing and Debugging a Procedure*.

## Running a Procedure

You can run a procedure on the local host or on a remote server at any point during the development process:

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

1. Select the procedure.
2. Choose one of the following methods:
   - Right-click the procedure, then choose *Run* from the shortcut menu.

     or

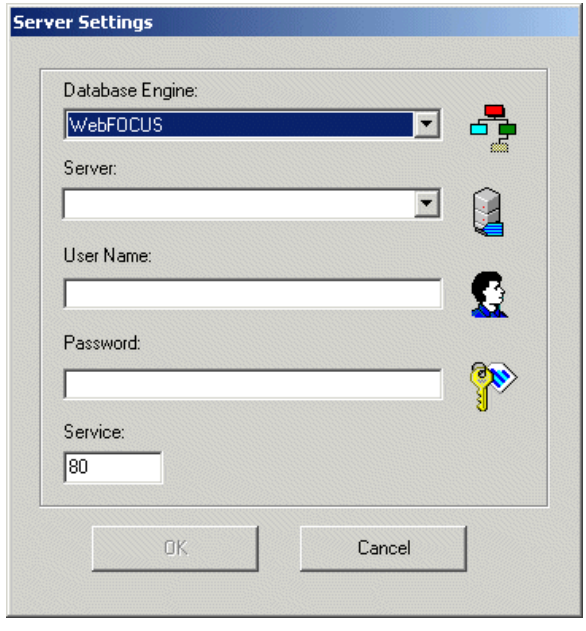
   - Click (Run) on the Main toolbar.

## *Procedure* How to Run a Procedure on a Remote Server

1. From an open project, right-click the procedure and select *Flag as Remote* from the shortcut menu or the File menu.
2. Right-click the procedure, then choose *Run* from the shortcut menu.

   or

   Select the procedure, then click the *Run* button on the Main toolbar. The Server Settings dialog box opens:




3. Select the remote server from the Server drop-down list and enter your user name and password to access that server.
4. Enter a port number where the remote server is listening in the Service field.
5. Click *OK*.
6. The selected procedure is run, and its output appears in the output facility associated with the tool that created it. For example, an HTML report or graph is displayed in the browser, and the result of a query command is displayed in the Command Console.

**Tip:** You can also set a procedure to run remotely from the New Procedure dialog box. See *Selecting a Creation Tool* on page 4-2.

## Canceling a Running Procedure

You may run a procedure and find that it takes too long to process or that it is the wrong one. You can cancel any procedure while it is running. To cancel a procedure at any time, you must use the Web Console. For more information, see the *iWay Server Administration for UNIX, Windows, OpenVMS, OS/400, OS/390 and z/OS* manual.

CHAPTER 5

# Understanding User Interface Basics

**Topics:**

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

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 in a variety of ways:

- Use Developer Studio's Report Layout Painter to create HTML forms and procedures in one integrated process. For details, see Chapter 6, *Designing a User Interface for a Web Application With the Report Layout Painter*.
- Use Microsoft® FrontPage® 2000, a graphical development tool integrated with WebFOCUS. For details, see Chapter 7, *Designing a User Interface With FrontPage*.

Regardless of the design method that you choose, read the following topics: *User Interface Basics* on page 5-2, *About a Logon Page* on page 5-2, *About a Launch Page* on page 5-6, and *About an HTML Display Page* on page 5-9.

In addition to the above techniques for creating an interface, you can code a user interface or modify a supplied template. For details, see Chapter 8, *Coding a User Interface* and Chapter 9, *Enhancing a User Interface*.

# 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. See *Creating a Main Page* on page 7-38 for a basic example.

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

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 NT/2000/XP:** *install_drive*:\ibi\WebFOCUS435\ibi_html

**UNIX:** /ibi/WebFOCUS52/ibi_html

**OS/390:** /ibi/WebFOCUS52/ibi_html

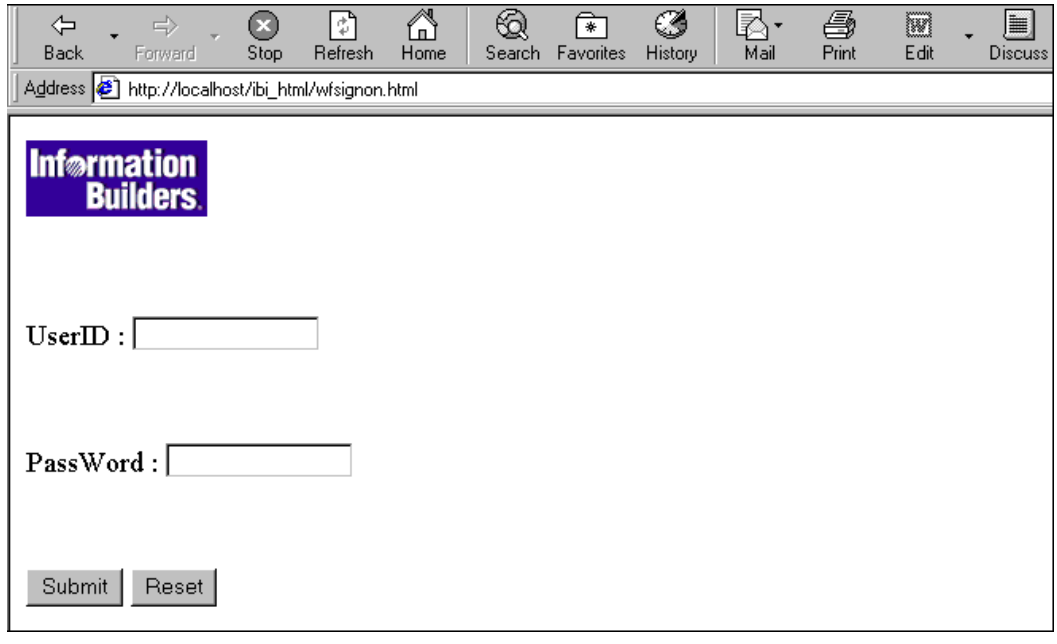
You can copy the file and modify the code for your application, or create a custom page of your own.

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. See Chapter 9, *Enhancing a User Interface*, for an example.

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




## *Procedure* How to Modify WFSIGNON.HTML

**1.** Modify WFSIGNON.HTML in one of two ways:

- Use the Developer Studio text editor (TED). For details on this editor, see Chapter 18, *Editing Application Components as Text in Developer Studio*.

  or

- Use the FrontPage HTML source editor. To use this editor, copy the file to your project directory, rename it with the three-letter extension .HTM, add it to your project, and open it in FrontPage. To open the file in FrontPage:

  In Developer Studio, open the project for which you are coding the logon page.

  Open the HTML Forms folder for the project. Right-click the name of the logon page, and click *Open* on the pop-up menu.

  The logon page opens in FrontPage.

  At the bottom left of the FrontPage window, click the *HTML* tab. The HTML source editor opens, displaying the HTML code.

**2.** Modify the lines with letters on the left, as explained in the notes following the code.

```
    <!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="/cgi-bin/ibi_cgi/ibiweb.exe" method="POST">
    <a href="http://www.informationbuilders.com"><img border="0"
    src="ib_logo_home.gif" width="114" height="48"></a>
    <p> </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,

```
<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

```
<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. See Chapter 12, *Partitioning and Deploying Project Files*, for details.

### *Example* Specifying an Absolute Address

```
<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">
```

### *Example* Specifying a Relative Address

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

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

## Setting a Browser

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 from WebFOCUS.

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

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

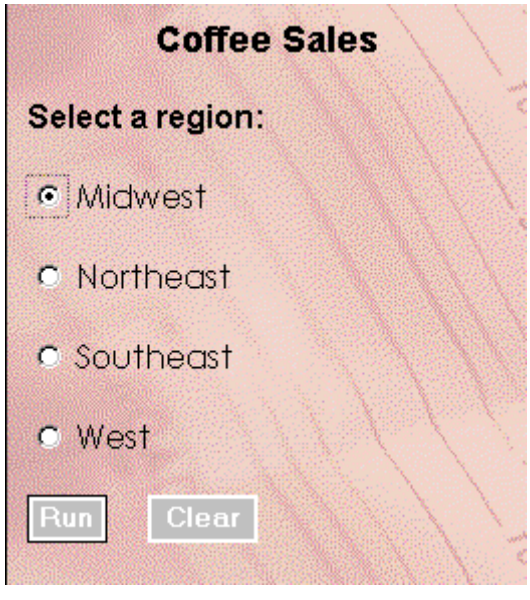
1. Launch Microsoft Internet Explorer.
2. From the Tools menu, select *Internet Options*.
3. Click the *Security* tab.
4. Click *Custom Level*.
5. Under the Cookies option, select *Enable* for both items.
6. Click *OK*.

# About a Launch Page

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 sample launch page was created in FrontPage. If a manager selects *Midwest* and clicks *Run*,




the report is displayed:

Coffee Sales for Midwest Region

| Unit Sales | Dollar Sales |
|---|---|
| 332777 | 4178513 |

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 Chapter 12, *Partitioning and Deploying Project Files*.

## Adding a Control to a Form

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.

| Control | Enables a user to... |
|---|---|
| One-Line Text Box | Enter a single value for a variable, such as user name or password on a custom logon page. |
| Radio Button | Choose one value from a group. |
| Check Box | Choose multiple values. |
| Drop-Down List | Choose a value from a drop-down list.<br>You can add text to describe the values. |
| Multi-Select Drop-Down List | Choose multiple values from a drop-down list. Allows a user to report on several field values at one time. |
| Dynamic Multi-Select Drop-Down List | Choose multiple values from a drop-down list populated by field values retrieved live from a data source. |
| Text Area | Enter an ad hoc report request. |
| User-Customized Menu | See Chapter 9, *Enhancing a User Interface,* for information on this control. |

| Control | Enables a user to... |
|---|---|
| Submit Button | Run a report.<br>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. |
| Push Button | Trigger an action. |
| Hyperlink | Run a procedure using fixed values. |

## 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.

FrontPage provides predefined templates for a single page and frameset, and WebFOCUS adds others to the available set. They simplify the creation of a launch page by presenting basic, standard page layouts. Optionally, you can develop your own layout by choosing the default blank report page or a simple frameset to start, and then enhancing it as desired.

## 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 Report Layout Painter. The Report Layout Painter allows you to create a launch and display page in an integrated process within Developer Studio. See Chapter 6, *Designing a User Interface for a Web Application With the Report Layout Painter*.
- Use FrontPage. FrontPage automatically generates the required code for the CGI. See Chapter 7, *Designing a User Interface With FrontPage*.
- Code the call to the WebFOCUS Client yourself. For details, see Chapter 8, *Coding a User Interface*.
- Customize templates supplied with your software. They include the required code for the CGI, but need modification for your application. For details, see Chapter 8, *Coding a User Interface*.

## 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 include:

- **Dynamic.** Created and formatted at the time a user requests it. Displays the most current information from live data retrieval. Called by the WebFOCUS Client. Two kinds of dynamic reports are drill-down and ad hoc.
- **Drill-Down.** Dynamic, multilevel report consisting of a parent (summary) component and a child (detailed) component. For an example of a drill-down report, see Chapter 9, *Enhancing a User Interface*.
- **Ad Hoc.** Dynamic report, coded by a user directly on a launch page. For instructions on enabling ad hoc reporting, see Chapter 8, *Coding a User Interface*.
- **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 Chapter 8, *Coding a User Interface*.

You can create a report using the Report Painter, Report Assistant, or TED.

CHAPTER 6

# Designing a User Interface for a Web Application With the Report Layout Painter

**Topics:**

- Uses for the Report Layout Painter
- Getting Started
- Creating an HTML Page Layout
- Using WebFOCUS Forms to Supply Parameter Values
- Styling Your Layout
- Laying Out Objects With the Report Layout Painter
- Customizing the Layout
- Controlling the Report Layout Painter Environment

The Report Layout Painter (previously the Layout tool) allows you to graphically create an HTML page that incorporates WebFOCUS forms, reports, graphs, and Web objects. The Report Layout Painter enables you to add these components to your HTML page in an integrated process within Developer Studio with the same precision as the -HTMLFORM command. You can also develop report and graph procedures by launching the Report Painter or the Graph Assistant from the Report Layout Painter.

This topic describes how to create a user interface in the local development environment. For information on creating a user interface in Managed Reporting, see your Managed Reporting documentation.

For local development and deployment, you must have a WebFOCUS Reporting Server installed on the same machine as Developer Studio. This server is referred to as the *localhost*. The files that you create locally reside in a repository on the localhost and are located by APP PATH logic.

After creating a project in the Report Layout Painter, you can deploy the project. For details, see Chapter 12, *Partitioning and Deploying Project Files*.

# Uses for the Report Layout Painter

You can do the following when creating a layout:

- Add reports and graphs to your HTML page. The Report Layout Painter allows you to control the display of your reports and graphs as well as supply parameter values, also known as variable values, at run time. For details, see *Adding a Report or Graph to the Layout* on page 6-8.
- Add Web objects such as images and frames to an HTML page. Adding objects allows you to customize your HTML page. For details, see *Creating an HTML Page Layout* on page 6-8.
- Use WebFOCUS form controls to execute requests and supply parameter values to your report or graph. For details, see *Using WebFOCUS Forms to Supply Parameter Values* on page 6-31.
- Lay out the page. You can change the location, size, and properties of all objects on your HTML page. For details, see *Laying Out Objects With the Report Layout Painter* on page 6-53.

**Note:** The Report Layout Painter does not support a report with an embedded Table of Contents (TOC) or an OLAP-enabled report. If you execute a report from the Report Layout Painter with one or both of these options, the output window will not display the TOC or the OLAP controls and you will receive a scripting error. In order to execute these types of reports, you must use a frame. For more information, see *Adding a Frame to the Layout* on page 6-14.

# Getting Started

You can create procedures and HTML forms in one integrated process with the Report Layout Painter. When you save your layout, both the procedure file (.fex) and the HTML file are saved in their respective Project folders. You can deploy both files with your project and they will retain their relationship.

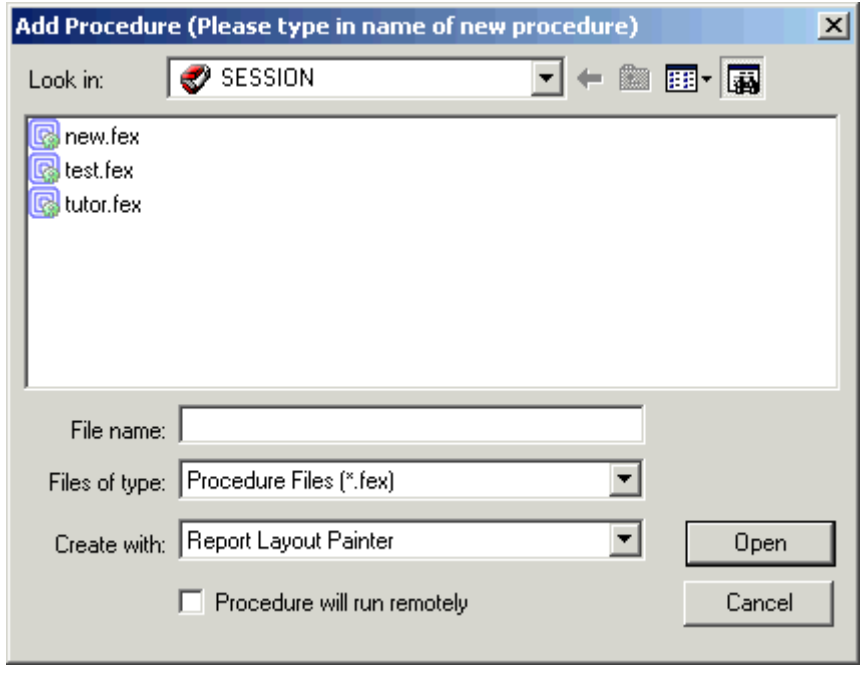
You access the Report Layout Painter when you select the Report Layout button in the Open dialog box.

## *Procedure* How to Access the Report Layout Painter

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:



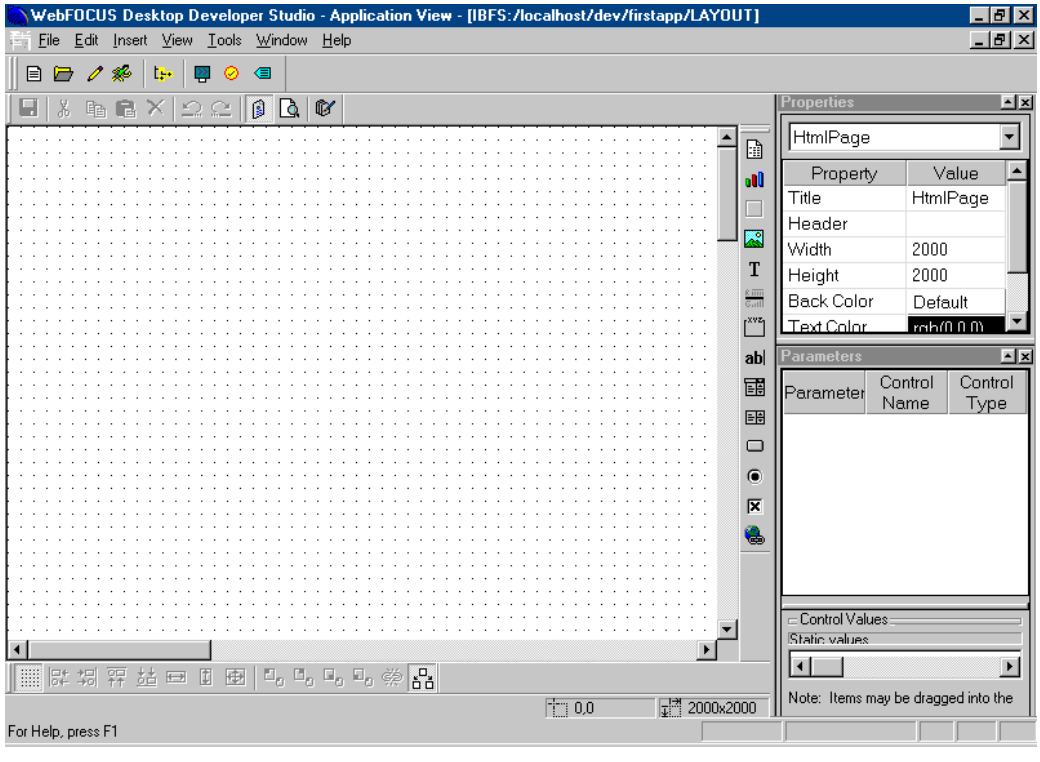

2. Enter a name for the new procedure in the File name field.
3. Select *Report Layout Painter* from the Create with drop-down list.

4. Optionally, select *Procedure will run remotely* to flag the procedure to run on a remote server.
5. Click *Open*.

The Report Layout Painter opens.

## *Reference* Report Layout Painter Windows and Toolbars

When you open the Report Layout Painter, it appears as the following:



The main elements of the Report Layout Painter are:

**Menu Bar**

Displays pull-down menus for the Report Layout Painter.

**Developer Studio Toolbar**

Displays tool buttons such as Open and Run. For details, see *Creating Reports With WebFOCUS Tools*.

**Standard Toolbar**

Displays buttons such as Cut and Paste that allow you edit the layout.

**Components Toolbar**

Contains buttons that add reporting objects and WebFOCUS form controls to the layout.

**Positioning Toolbar**

Contains buttons that control the appearance of the layout. For details, see *Layout Toolbar* on page 6-55.

**Properties Window**

Displays the properties of objects in the layout. For details, see *Working With the Properties Window* on page 6-44.

**Parameters Window**

Displays information about the parameters in your report or graph. For details, see *Working With the Parameters Window* on page 6-45.

*Reference*

## Standard Toolbar

The Standard toolbar contains the following buttons:

| Button | Description |
|---|---|
| | Saves the existing procedure (.fex) and HTML files to the current project. This button is grayed out once the layout is saved and no new changes have been made. When a change has been made to the layout, the Save button is active until the report is saved again. |
| | Removes the highlighted object(s) and saves it to the clipboard. |
| | Copies the highlighted object(s) to the clipboard. |

| Button | Description |
|---|---|
| | Pastes the object(s) to the specified location. |
| | Deletes the highlighted object(s). |
| | Resets the layout by reversing the last action performed. |
| | Repeats the last action performed. |
| | Displays the layout in design mode. In design mode, you can perform all operations. This button is recessed when you are in design mode. |
| | Displays the layout in preview mode. Preview mode executes the procedure and displays the result in Developer Studio.This allows you to preview the finished report. This button is recessed when in preview mode. |
| | Launches FrontPage and enables you to use one of the existing themes (templates). |

## *Reference* Components Toolbar

The Components toolbar contains the following buttons:

| Button | Description |
|---|---|
| | Inserts a report object to the layout. |
| | Inserts a graph object to the layout. |
| | Inserts a placeholder for an IFRAME. An IFRAME can have its own URL and contain HTML content, and it can be a placeholder for a drill-down report. |
| | Inserts a placeholder for an image. |
| | Inserts text. |
| | Inserts a line. |
| | Adds a group box. |
| | Adds a text box. |
| | Adds a drop-down list. |
| | Adds a list box (a drop-down list that allows multiple selections). |
| | Adds a push button. |
| | Adds a radio button. |
| | Adds a check box list. |
| | Inserts a hyperlink. |

# Creating an HTML Page Layout

You can use the Report Layout Painter to create an HTML page that launches and displays your report or graph. You can add elements to the HTML page including a report or graph, text, and form controls. The following is a complete list of objects you can add to your HTML page with the Report Layout Painter:

- A **report or graph**. For details, see *Adding a Report or Graph to the Layout* on page 6-8.
- A **frame**. For details, see *Adding a Frame to the Layout* on page 6-14.
- A **line**. For details, see *Adding a Line to the Layout* on page 6-17.
- An **image**. For details, see *Adding an Image to the Layout* on page 6-19.
- **Text**. For details, see *Adding Text to the Layout* on page 6-21.
- A **hyperlink**. For details, see *Adding a Hyperlink to the Layout* on page 6-23.
- A **group box**. For details, see *Adding a Group Box to the Layout* on page 6-24.
- A **push button**. For details, see *Adding a Push Button to the Layout* on page 6-27.
- A **WebFOCUS form** control. For details, see *Adding a Push Button to the Layout* on page 6-27 and *Using WebFOCUS Forms to Supply Parameter Values* on page 6-31.
- A **theme** (template). For details, see *Customizing the Layout* on page 6-56.
- An **HTML template**. For details, see *Customizing the Layout* on page 6-56.

You can also set properties for the HTML page in the Report Layout Painter. For details, see *Setting HTML Page Properties* on page 6-52.

## Adding a Report or Graph to the Layout

You can add a report or graph to the Layout that will display when you run the procedure. You can add a new report which you create in the Report Painter or Report Assistant, a new graph which you create in the Graph Assistant, or an existing report or graph that resides on an available server.

You can also include parameters in a report whose values can be assigned with WebFOCUS forms that are added with the Report Layout Painter. For information, see *Using WebFOCUS Forms to Supply Parameter Values* on page 6-31.

You can set the graphic used as a placeholder for a report or graph in the layout using the Settings dialog box. For details, see *How to Set Page Properties* on page 6-56.

## *Procedure* How to Add a New Report or Graph to a Layout

1. Insert a report or graph object by doing one of the following:
   - Click the *Insert report* or *Insert graph* button.
   - From the Insert menu, select *New Report* or *New Graph*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Report* or *Graph*.

   The cursor changes into a crosshair.
2. Click and drag the crosshair to create a report or graph object and adjust it to the size you want.

   A report or graph object is created in the layout and assigned the name visiblefex*n* where *n* is a number. The object will appear in gray and white to indicate that the placeholder does not have a report or graph associated with it. Once a report or graph is associated with the object, the object appears in color.
3. Create a report or graph by doing one of the following:
   - Double-click the placeholder.

     or
   - Right-click the placeholder, select *Report properties* for a report or *Graph properties* for a graph, and then select *Open*.

   The Open dialog box opens.
4. Select the Master File you want to report against from the Master File list, select *Open Report Painter* or *Open Report Assistant*, and click *Open*.

   The selected tool opens. For details on using these tools, see *Creating Reports With the Report Assistant*, *Creating Reports With the Report Painter*, and *Creating Graphs With the Graph Assistant* in the *Creating Reports With Graphical Tools* manual.
5. Optionally, after creating the report or graph, you can change its properties by selecting the placeholder's name from the Properties window drop-down list and making your changes. For details, see *Report and Graph Properties Window* on page 6-11.

## *Procedure* How to Add an Existing Report or Graph to a Layout

1. Insert a report object by doing one of the following:

   - Click the *Insert report* or *Insert graph* button.

     The cursor changes into a crosshair. Click and drag the crosshair to create a report or graph object and adjust it to the size you want.

     A report object is created in the layout and assigned the name visiblefex*n* where *n* is a number. The object will appear in black and white to indicate that the placeholder does not have a report or graph associated with it.

   - From the Insert menu, select *Existing Report* or *Existing Graph*.

   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Report* or *Graph*.

2. Do one of the following:

   **For a report:**

   - Right-click the report object and select *Report Properties* and then *Get existing report*.

     or

   - From the Insert menu, select *Import Existing Procedure*.

   **For a graph:**

   - Right-click the graph object and select *Graph properties* from the pop-up menu, then select *Get existing graph*.

   The Get existing focexec dialog box opens.

   Enter the name of the procedure, or click the browse (...) button to the right of the field and select the procedure you want to add to the layout.

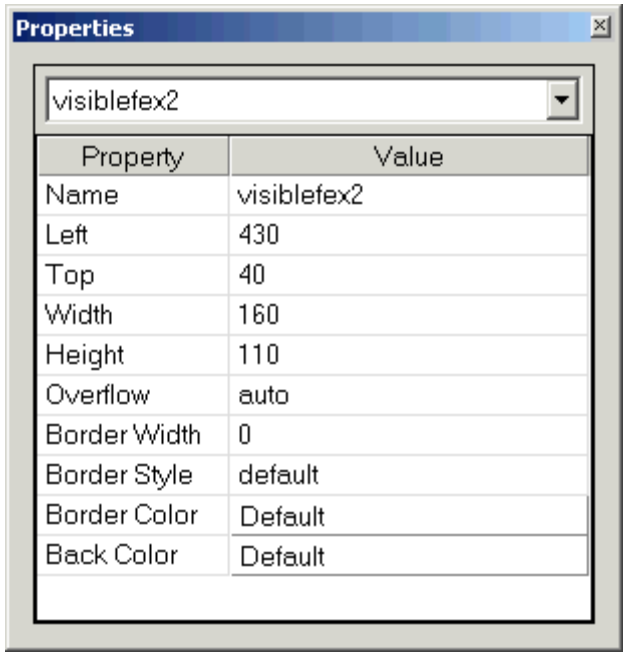
3. Click *OK*.

   The Manage Layout dialog box opens. For details, see *Manage Layout Dialog Box* on page 6-13.

   Make your changes, and click *OK*.

4. Optionally, change the procedure's properties by selecting the report or graph object's name from the Properties window drop-down list and making your changes. For details, see *Report and Graph Properties Window* on page 6-11.

## *Reference* Report and Graph Properties Window




The Report Properties window contains the following fields/options:

**Name**

Is the name of the report or graph object.

**Left**

Is the distance in pixels from the left of the layout to the report or graph's left edge.

**Top**

Is the distance in pixels from the top of the layout to the report or graph's top edge.

**Width**

Is the report or graph object's width in pixels.

**Height**

Is the report or graph object's height in pixels.

**Overflow**

Determines the behavior of the report or graph when its output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report or graph is too large. This is the default value.

**visible** always displays the entire report whether or not the report or graph is too large. This may cause objects to overlap.

**hidden** crops the report or graph if it is too large for the report object.

**scroll** displays scrollbars even if the report or graph fits within the report object.

**Border Width**

Is the width of the border. Border Style must be set to a value other than none to set the Border Width value.

**Border Style**

Determines the style of the report or graph's border. The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's color. The default value is black.

**Back Color**

Is the report's background color. The default value is white.

*Procedure*

## How to Set the Report or Graph Icon

1. Select *Settings...* from the Tools menu.

   The Settings dialog box opens.

2. Click the ... button next to the field for the icon you want to change, and navigate to the directory that contains the report and graph icons.

3. Select a new icon and click *Open*.

   You return to the Settings dialog box.

4. Click *OK*.

## Modifying Report and Graph Components

You can select to include or exclude components within your procedure with the Manage Layout dialog box. The Manage Layout dialog box allows you to select from components of your request such as TABLE, SET, and GRAPH.

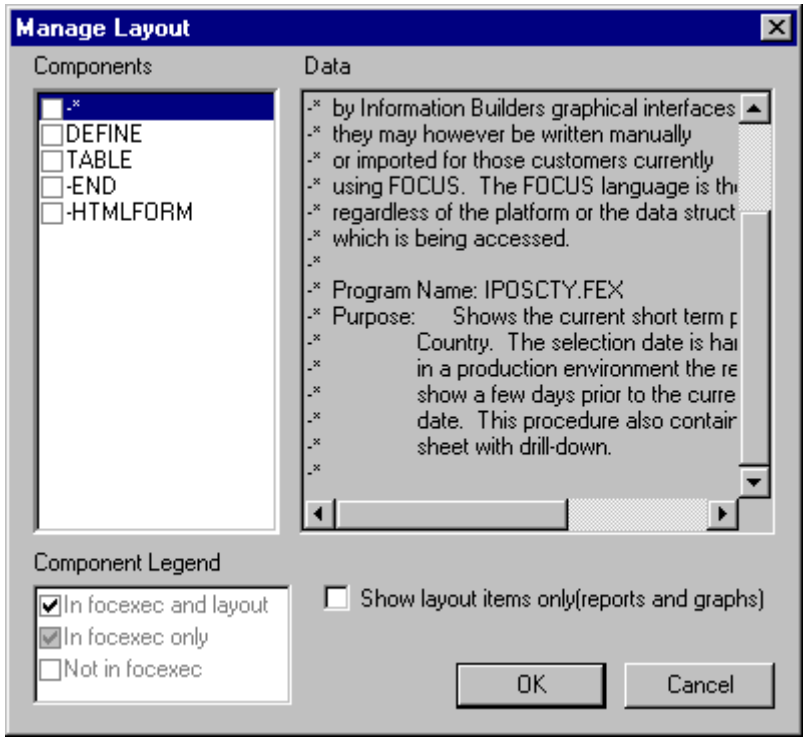
If you have several requests in one procedure, you can use the Manage Layout dialog box to prevent the execution of some of these requests.

## *Procedure* How to Modify Report or Graph Components

Select *Manage Layout...* from the Tools menu.

The *Manage Layout* dialog box opens. For details, see *Manage Layout Dialog Box* on page 6-13.

## *Reference* Manage Layout Dialog Box




The Manage Layout dialog box contains the following fields/options:

**Components**

Displays all the components (SET, DEFINE, etc.) in the request.

**Data**

Displays the contents of the selected component.

**Component Legend**

Determines whether the request is included in the procedure, outside of the procedure, or in the procedure and layout.

**In focexec and layout**

Activates the component and displays the component in the layout.

**In focexec only**

Deactivates the component and does not display the component in the layout. This is a good method for hiding report requests that produce temporary files.

**Not in focexec**

Removes the component from the procedure.

**Show layout items only (reports and graphs)**

Displays the layout's reports and graphs in the Data field.

This check box only displays when you access the Manage Layout dialog box from the Insert menu. When accessing it from the pop-up menu, only the report or graph component displays, and the check box does not apply.

## Adding a Frame to the Layout

You can use a frame to embed additional Web sources or run reports. You can also use a frame as the output location or target for a drill-down report. You can also use a frame to run a table of contents report or an OLAP report. By default, frames that are used as targets are hidden until they are populated with output.

You can also make a frame invisible until you display a report or Web page in it.

### *Procedure* How to Add a Frame to the Layout

1. Insert a frame by doing one of the following:
   - Click the *Frame* button.
   - From the Insert menu, select *Frame*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Frame*.

   The cursor changes into a crosshair.
2. Click and drag the crosshair to create a frame and adjust it to the size you want.

   A frame is created in the layout and assigned the name iframe*n* where *n* is a number.
3. Optionally, change the frame's properties by selecting the frame's name from the Properties window drop-down list and making your changes. For details, see *Frame Properties Window* on page 6-16.

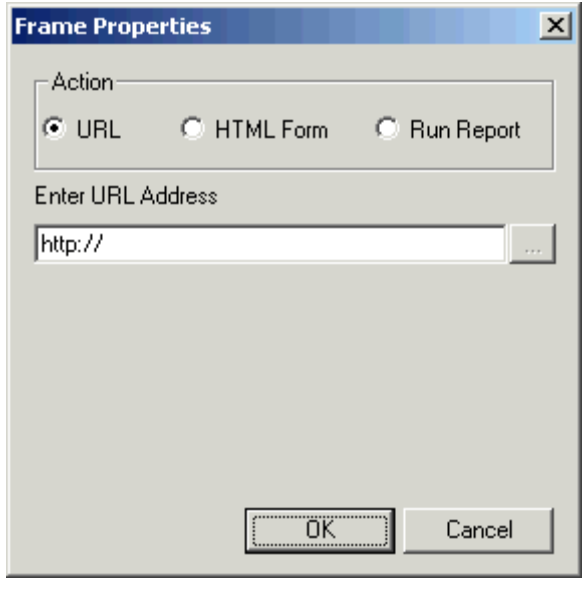
## *Procedure* How to Assign a URL, HTML Form, or Report to a Frame

1. Insert a frame by doing one of the following:
   - Click the *Frame* button.
   - From the Insert menu, select *Frame*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Frame*.

   The cursor changes into a crosshair.
2. Click and drag the crosshair to create a frame and adjust it to the size you want.

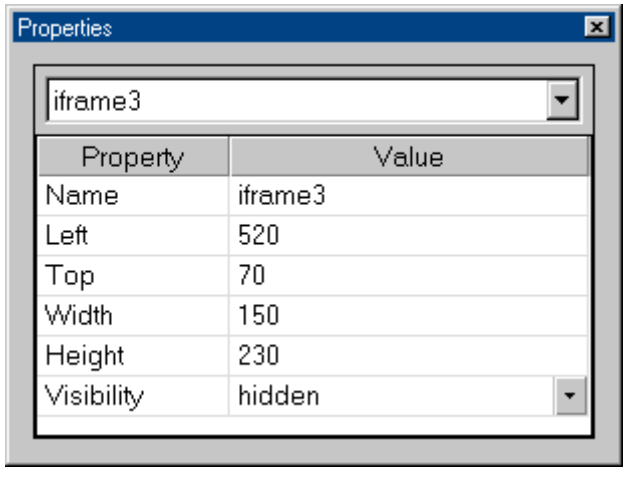
   A frame is created in the layout and assigned the name iframe*n* where *n* is a number.
3. Right-click the frame, and select *Frame Properties* from the pop-up menu.

   The Frame Properties dialog box opens:



4. Do one of the following:
   - To assign a **URL** to the frame, select the *URL* radio button and enter the URL address in the Enter URL Address field.
   - To assign an **HTML form** to the frame, select the *HTML Form* radio button and enter the form's location in the Select HTML Form field, or click the ... button to browse to the location.
   - To assign **report output** to the frame, select the *Run Report* radio button and enter the report's location in the Select Report field, or click the ... button to browse to the location.
5. Click *OK*.
6. Optionally, change the frame's properties by selecting the frame's name from the Properties window drop-down list and making your changes. For details, see *Frame Properties Window* on page 6-16.

## *Reference* Frame Properties Window




The Frame Properties window contains the following fields/options:

**Name**

Is the name of the frame.

**Left**

Is the distance in pixels from the left of the layout to the frame's left edge.

**Top**

Is the distance in pixels from the top of the layout to the frame's top edge.

**Width**

Is the frame's width in pixels.

**Height**

Is the frame's height in pixels.

**Visibility**

Determines whether the empty frame is visible before a request is executed. The options are:

**visible** displays an empty frame in the layout.

**hidden** hides an empty frame in the layout until it is populated with a drill-down report, hyperlink, or URL. This is the default value.

## Adding a Line to the Layout

You can add a horizontal or vertical line to the layout. This is useful for distinguishing between sections of your launch or display page.

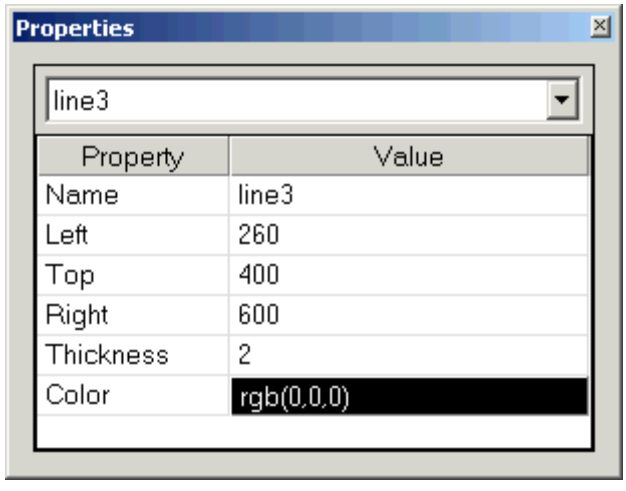
### *Procedure* How to Add a Line to the Layout

1. Do one of the following:
   - Click the *Insert line* button.
   - From the Insert menu, select *Line*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Line*.

   The cursor changes into a crosshair.
2. Click and drag the crosshair to create a horizontal or vertical line.

   A line is created in the layout.
3. Optionally, change the line's properties by selecting the line's name from the Properties window drop-down list and making your changes. For details, see *Line Properties Window* on page 6-18.

## *Reference* Line Properties Window




The Line Properties window contains the following fields/options:

**Name**

Is the name of the line.

**Left**

Is the distance in pixels from the left of the layout to the line's left edge.

**Top**

Is the distance in pixels from the top of the layout to the line's top edge.

**Right**

Is the distance in pixels from the right of the layout to the line's right edge.

**Thickness**

Is the line's thickness in pixels.

**Color**

Is the line's color.

## Adding an Image to the Layout

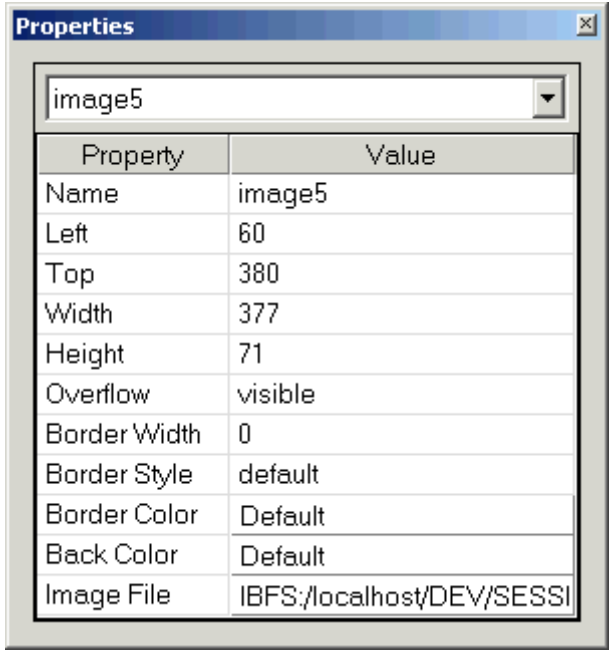
You can add an image to the layout. This is useful for including graphics such as a company logo.

### *Procedure* How to Add an Image to a Layout

1. Do one of the following:
   - Click the *Insert image* button.
   - From the Insert menu, select *Image*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Image*.

   The Get source file dialog box opens.
2. Click the browse button (...) to the right of the blank field and select the image you want to add to the layout.
3. Click *OK*.
4. Optionally, change the image's properties by selecting the image placeholder's name from the Properties window drop-down list and making your changes. For details, see *Image Properties Window* on page 6-19.

### *Reference* Image Properties Window



The Image Properties window contains the following fields/options:

**Name**

Is the name of the image.

**Left**

Is the distance in pixels from the left of the layout to the image's left edge.

**Top**

Is the distance in pixels from the top of the layout to the image's top edge.

**Width**

Is the image's width in pixels.

**Height**

Is the image's height in pixels.

**Overflow**

Determines the behavior of the report when the report output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report is too large.

**visible** always displays scrollbars whether or not the report is too large. This may cause objects to overlap. This is the default value.

**hidden** crops the report if it is too large for the report object.

**scroll** displays scrollbars even if the report fits within the report object.

**Border Width**

Is the width of the border. This value cannot be changed when Border Style is set to none.

**Border Style**

Determines the style of the image's border.The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's RGB color. The default value is black.

**Back Color**

Is the image's background color. The default value is black.

**Image File**

Is the location of the image file.

## Adding Text to the Layout

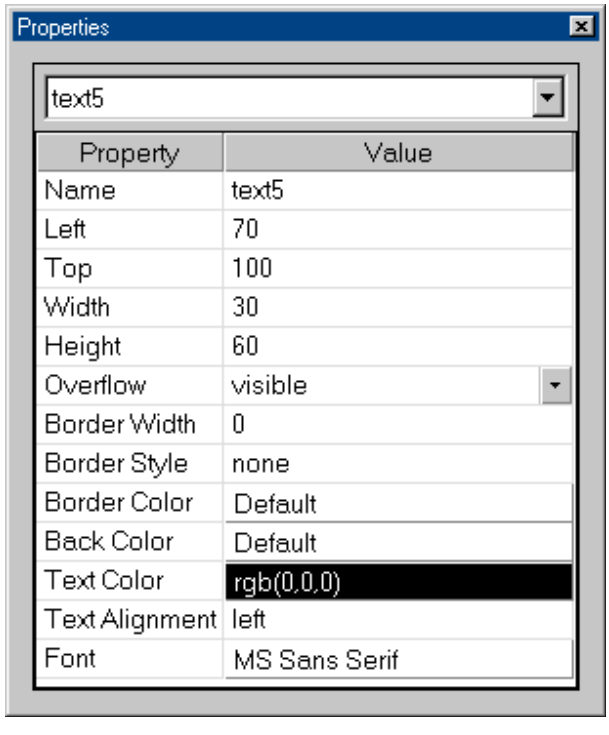
You can add text to the layout. This is useful for including headings for your Web page, or adding directions or explanation for your report or graph.

### *Procedure* How to Add Text to a Layout

1. Insert text to the layout by doing one of the following:
   - Click the *Insert Text* button.
   - From the Insert menu, select *Text*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Text*.

   A text object appears.
2. Replace the text with the text you want to appear in the layout.
3. Optionally, change the text's properties by selecting the text object's name from the Properties window drop-down list and making your changes. For details, see *Text Properties Window* on page 6-21.

### *Reference* Text Properties Window



The Text Properties window contains the following fields/options:

**Name**

Is the name of the text object.

**Left**

Is the distance in pixels from the left of the layout to the text object's left edge.

**Top**

Is the distance in pixels from the top of the layout to the text object's top edge.

**Width**

Is the width of the text object in pixels.

**Height**

Is the height of the text object in pixels.

**Overflow**

Determines the behavior of the report when the report output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report is too large.

**visible** always displays scrollbars whether or not the report is too large. This may cause objects to overlap. This is the default value.

**hidden** crops the report if it is too large for the report object.

**scroll** displays scrollbars even if the report fits within the report object.

**Border Width**

Is the width of the border. This value cannot be changed when Border Style is set to none.

**Border Style**

Determines the style of the text object's border. The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's color.

**Back Color**

Is the text's background color.

**Text Color**

Is the color of the text.

**Text Alignment**

Is the text alignment within the text object. The options are **left**, **right**, or **center**.

**Font**

Is the text's font. The default value is default font for your browser unless you set it to a different value.

## Adding a Hyperlink to the Layout

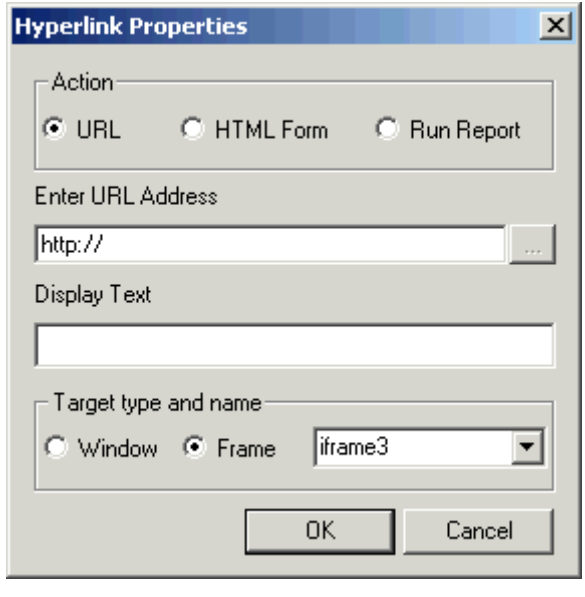
You can create a hyperlink for your HTML page. A hyperlink can execute a report, link to a URL, or open an HTML page. You can create a hyperlink in two ways:

- Insert a hyperlink. For details, see *How to Create a Hyperlink* on page 6-23.
- Add a hyperlink to a button. For details, see *Adding a Push Button to the Layout* on page 6-27.

### *Procedure* How to Create a Hyperlink

1. Do one of the following:
   - Click the *Insert Hyperlink* button.
   - Select *Hyperlink* from the Insert menu.
   - Add the text to use as the hyperlink, right-click the text, and select *Create hyperlink...*

   The Hyperlink Properties dialog box opens:



2. Set the action of the hyperlink:
   - To link to a URL, select *URL* in the Action section, and enter the URL in the Enter URL Address field.
   - To open an HTML page, select *HTML Form* in the Action section, and enter the HTML page in the Select HTML Form field.
   - To execute a report, select *Run Report* and enter the report name in the Select Report field.
3. Enter the text you want to display as the hyperlink in the Display Text field.
4. Optionally, direct the output to a specific location by selecting *Window* or *Frame* in the Target type and name section.
5. Select the target for the window or from from the drop-down list.
6. Click *OK*.

## Adding a Group Box to the Layout

A group box can be used to create a border around a group of objects, for example, WebFOCUS forms or reports and graphs.

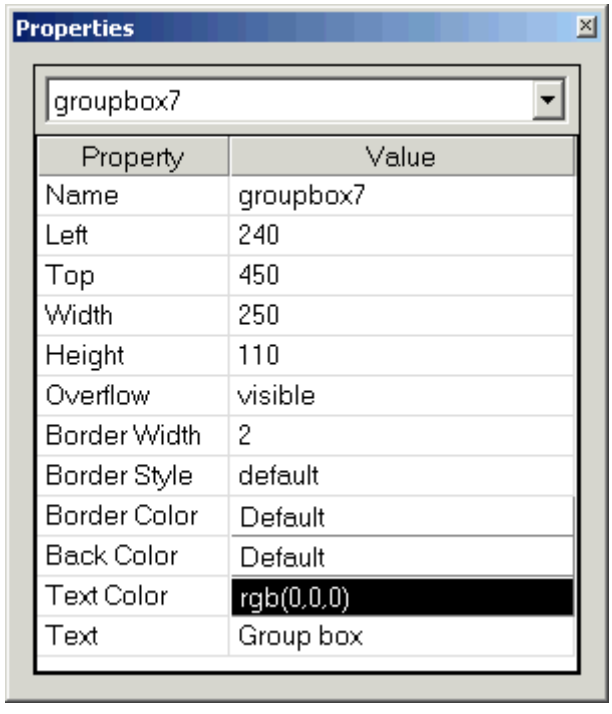
### *Procedure* How to Add a Group Box

1. Insert a group box by doing one of the following:
   - Click the *Group box* button.
   - From the Insert menu, select *Group box*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Group box*.

   The cursor changes into a crosshair.
2. Click and drag the crosshair to create a group box and adjust it to the size you want.

   A group box is created in the layout and assigned the name groupbox*n* where *n* is a number.
3. Optionally, change the group box's properties by selecting the group box's name from the Properties window drop-down list and making your changes. For details, see *Group Box Properties Window* on page 6-25.

## *Reference* Group Box Properties Window




The Group Box Properties window contains the following fields/options:

**Name**

Is the name of the group box placeholder.

**Left**

Is the distance in pixels from the left of the layout to the group box's left edge.

**Top**

Is the distance in pixels from the top of the layout to the group box's top edge.

**Width**

Is the group box's width in pixels.

**Height**

Is the group box's height in pixels.

**Overflow**

Determines the behavior of the report when the report output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report is too large.

**visible** always displays scrollbars whether or not the report is too large. This may cause objects to overlap. This is the default value.

**hidden** crops the report if it is too large for the report object.

**scroll** displays scrollbars even if the report fits within the report object.

**Border Width**

Is the width of the border. This value cannot be changed when Border Style is set to none.

**Border Style**

Determines the style of the group box's border. The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's RGB color. The default value is purple.

**Back Color**

Is the group box's background color. The default value is black.

**Text**

Is the text that displays at the top of the group box.

## Adding a Push Button to the Layout

You can add a push button to the layout. A push button allows you to submit a report, or link to a URL or HTML form.

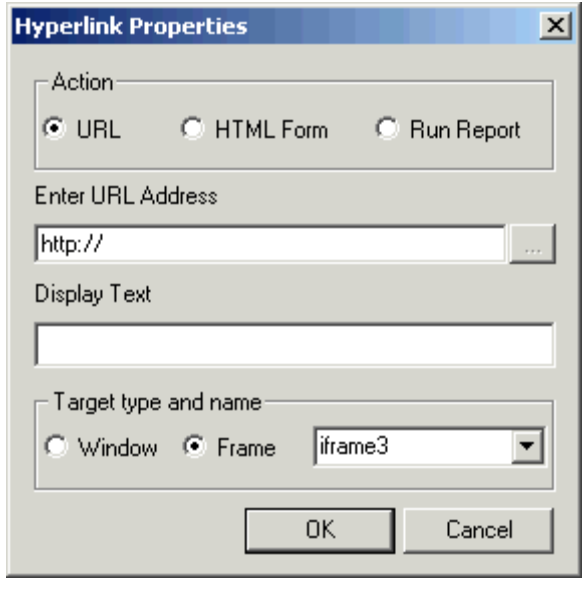
### *Procedure* How to Add a Hyperlink to a Push Button

1. Insert a push button by doing one of the following:
   - Click the *Push button* button.
   - From the Insert menu, select *Push Button*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Push Button*.

   The cursor changes into a crosshair.
2. Click and drag the crosshair to create a push button and adjust it to the size you want.

   A push button is created in the layout and assigned the name button*n* where *n* is a number.
3. Right-click the button, and select *Button properties*, then *Create hyperlink*....

   The Hyperlink Properties dialog box opens:



4. Set the action of the hyperlink:
   - To link to a URL, select *URL* in the Action section, and enter the URL in the Enter URL Address field.
   - To open an HTML page, select *HTML Form* in the Action section, and enter the HTML page in the Select HTML Form field.
   - To execute a report, select *Run Report* and enter the report name in the Select Report field.
5. Enter the text you want to display as the hyperlink in the Display Text field.
6. Optionally, direct the output to a specific location by selecting *Window* or *Frame* in the Target type and name section.
7. Select the target for the window or from from the drop-down list.
8. Click *OK*.
9. Optionally, change the push button's properties by selecting the push button's name from the Properties window drop-down list and making your changes. For details, see *Push Button Properties Window* on page 6-29.

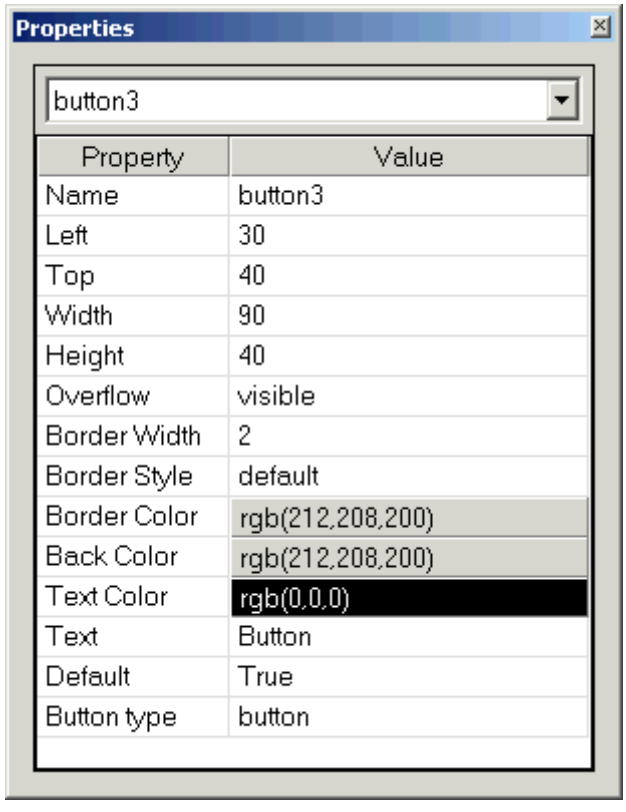
## *Procedure* How to Create a Submit or Reset Button

1. Insert a push button by doing one of the following:
   - Click the *Push button* button.
   - From the Insert menu, select *Push Button*.
   - Right-click in the layout, select *Insert* from the pop-up menu, then select *Push Button*.

   The cursor changes into a crosshair.
2. Click and drag the crosshair to create a push button and adjust it to the size you want.

   A push button is created in the layout and assigned the name button*n* where *n* is a number.
3. In the Push Button Properties window, select *submit* or *reset* from the Button type field.

## *Reference* Push Button Properties Window




The Push Button Properties window contains the following fields/options.

**Name**

Is the name of the push button placeholder.

**Left**

Is the distance in pixels from the left of the layout to the push button's left edge.

**Top**

Is the distance in pixels from the top of the layout to the push button's top edge.

**Width**

Is the push button's width in pixels.

**Height**

Is the push button's height in pixels.

**Overflow**

Determines the behavior of the report when the report output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report is too large.

**visible** always displays scrollbars whether or not the report is too large. This may cause objects to overlap. This is the default value.

**hidden** crops the report if it is too large for the report object.

**scroll** displays scrollbars even if the report fits within the report object.

**Border Width**

Is the width of the border. This value cannot be changed when Border Style is set to none.

**Border Style**

Determines the style of the group box's border. The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's RGB color. The default value is black.

**Back Color**

Is the group box's background color. The default value is gray.

**Text**

Is the text that displays on the push button.

**Button type**

Determines whether the push button is a hyperlink button, a submit button, or a reset button. The options are:

**button** specifies that the button can link to a URL or HTML form, or run a report.

**submit** specifies that the button submits a request.

**reset** specifies that the button resets the form.

## Adding a WebFOCUS Form

A WebFOCUS form is added automatically to the Report Layout Painter when a parameter is specified in your report or graph. A WebFOCUS form can also be added and associated with a parameter. For details on WebFOCUS forms and how to use them, see *Using WebFOCUS Forms to Supply Parameter Values* on page 6-31.

# Using WebFOCUS Forms to Supply Parameter Values

WebFOCUS forms allow you to prompt users for a parameter value. When you create a parameter as part of a report or graph, the Report Layout Painter automatically adds a WebFOCUS form, Submit button, and Reset button for the parameter to your layout, and the parameter appears in the parameters box. You can also add a WebFOCUS form and then associate it with a parameter.

When you delete a parameter in the Report Painter that was assigned a form in the Report Layout Painter, you are prompted to choose whether or not to delete the form control automatically. You can also change this setting in the Settings dialog box. For details, see *How to Change the Type of WebFOCUS Form Control Associated With a Parameter* on page 6-46.

WebFOCUS forms, with the exception of a text box which does not supply a list of possible values, can supply values to the user with a dynamic list or static list of values:

- A *dynamic* list retrieves values from a specified data source when the request is run.
- A *static* list consists of a list of values you supply. These values do not change unless you change them.

The properties of a WebFOCUS form as well as the parameters associated with each form can be controlled with the Properties window and Parameters window. For details, see *Working With the Properties Window* on page 6-44 and *Working With the Parameters Window* on page 6-45, respectively.

*Reference*

## Types of WebFOCUS Forms

WebFOCUS forms can be single or multi-select forms.

Single-select forms are:

- A **text box**. For details, see *Using a Text Box* on page 6-32.
- A **drop-down list**. For details, see *Using a Drop-Down List* on page 6-34.
- **Radio buttons**. For details, see *Using Radio Buttons* on page 6-42.

Multi-select forms are:

- A **list box**. For details, see *Using a List Box* on page 6-36.
- **Check boxes**. For details, see *How to Add a List Box* on page 6-36.

By default, a text box is created for a single parameter, and a drop-down list is created for a multi-select parameter. You can change the type of WebFOCUS form using the Parameters window. For details, see *How to Change the WebFOCUS Form Control Associated With a Parameter* on page 6-46.

## Using a Text Box

A text box allows the user to enter a parameter's value in a text entry field.

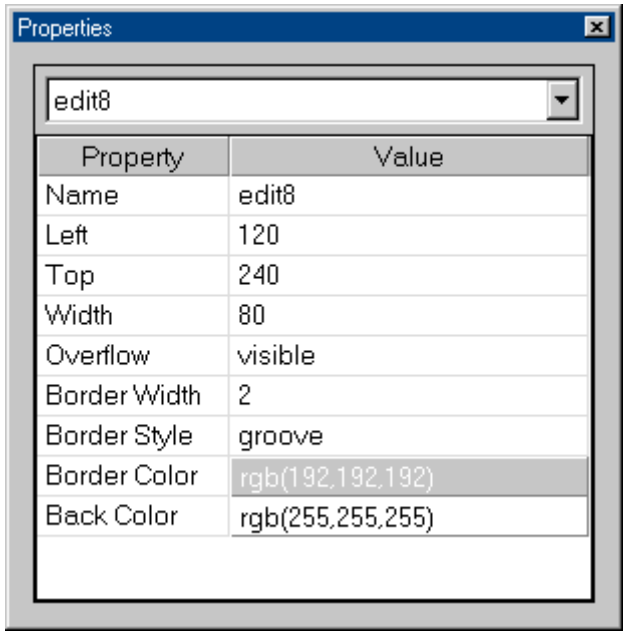
### *Procedure* How to Create a Text Box

1. Add a text box by doing one of the following:
   - Click the *Text Box* button
   - From the Insert menu, select *Text box*.

   The cursor changes to a crosshair.
2. Click and drag the crosshair to create a text box and adjust it to the size you want.

   A text box is created in the layout and assigned the name edit*n* where *n* is a number.
3. In the Control field for the parameter you want to associate the text box with, select the text box's name from the drop-down list.
4. Optionally, enter a value in the Initial Value field.
5. Optionally, change the text box's properties by selecting the text box's name from the Properties window drop-down list and making your changes. For details, see *Text Box Properties Window* on page 6-33.

## *Reference* Text Box Properties Window




The Text Box Properties window contains the following fields/options:

**Name**

Is the name of the text box placeholder.

**Left**

Is the distance in pixels from the left of the layout to the text box's left edge.

**Top**

Is the distance in pixels from the top of the layout to the text box's top edge.

**Width**

Is the text box's width in pixels.

**Overflow**

Determines the behavior of the text box when the text entered is too large for the box.

**Border Width**

Is the width of the border. This value cannot be changed when Border Style is set to none.

**Border Style**

Determines the style of the text box's border. The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's color.

**Back Color**

Is the text box's background color.

## Using a Drop-Down List

A drop-down list allows a user to select a single value from a list of supplied values. You can use either a dynamic drop-down list, or a static drop-down list.

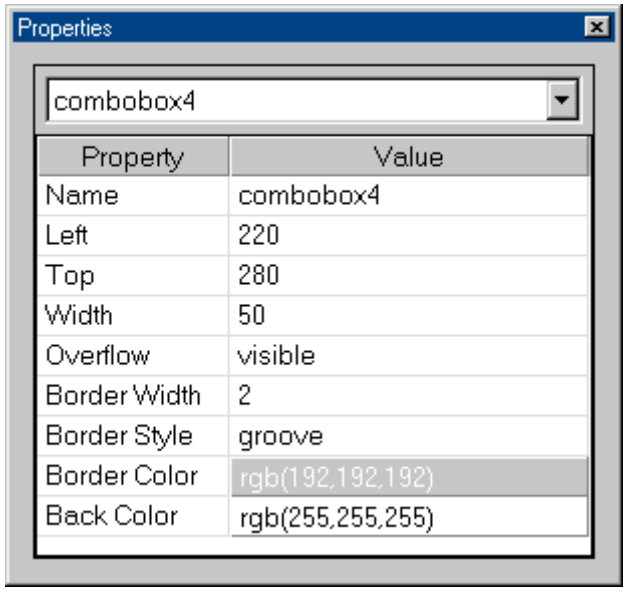
### *Procedure* How to Add a Drop-Down List

1. Add a drop-down list by doing one of the following:
   - Click the *drop-down list* button.
   - From the Insert menu, select *drop-down list*.

   The cursor changes to a crosshair.
2. Click and drag the crosshair to create a drop-down list, and adjust it to the size you want.

   A drop-down is created in the layout and assigned the name combobox*n* where *n* is a number.
3. In the Control field for the parameter you want to associate the drop-down list with, select the drop-down list's name from the drop-down list.
4. In the Create list items section of the Parameters window, select static or dynamic values for the drop-down list. For details, see *Working With the Parameters Window* on page 6-45.
5. Optionally, change the drop down list's properties by selecting the drop-down list's name from the Properties window drop-down list, and making your changes. For details, see *Drop-Down List Properties Window* on page 6-35.

## *Reference* Drop-Down List Properties Window




The Drop-Down List Properties window contains the following fields/options:

**Name**

Is the name of the drop-down list placeholder.

**Left**

Is the distance in pixels from the left of the layout to the drop-down list's left edge.

**Top**

Is the distance in pixels from the top of the layout to the drop-down list's top edge.

**Width**

Is the report object's width in pixels.

**Overflow**

Determines the behavior of the report when the report output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report is too large.

**visible** always displays scrollbars whether or not the report is too large. This may cause objects to overlap. This is the default value.

**hidden** crops the report if it is too large for the report object.

**scroll** displays scrollbars even if the report fits within the report object.

**Border Width**

Is the width of the border. Border Style must be set to a value other than none to set the Border Width value.

**Border Style**

Determines the style of the report's border. The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's color.

**Back Color**

Is the drop-down list's background color.

## Using a List Box

A list box allows a user to select single or multiple values at one time:

- A *single* select list allows the user to select only one value for each time a request is run.
- A *multi-select* list allows the user to select multiple values by using the Ctrl key while selecting values. In order to provide multiple values, the procedure must be set up to accept multiple values.

List box values can be dynamic or static.

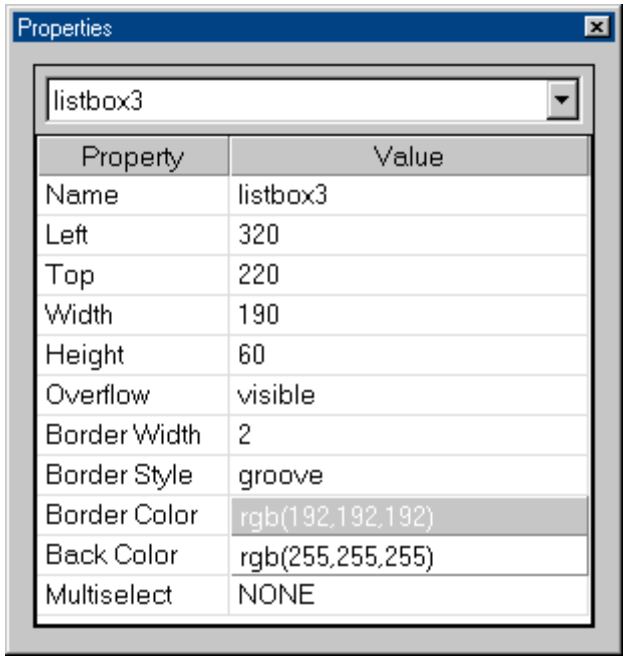
### *Procedure* How to Add a List Box

1. Add a list box by doing one of the following:
   - Click the *list box* button.
   - From the Insert menu, select *List box*.

   The cursor changes to a crosshair.
2. Click and drag the crosshair to create a list box, and adjust it to the size you want.

   A list box is created in the layout and assigned the name listbox*n* where *n* is a number.
3. In the Control field for the parameter you want to associate the drop-down list with, select the list box's name from the drop-down list.
4. In the Create list items section of the Parameters window, select static or dynamic values for the list box. For details, see *Working With the Parameters Window* on page 6-45.
5. Optionally, change the list box's properties by selecting the list box's name from the Properties window drop-down list, and making your changes. For details, see *Drop-Down List Properties Window* on page 6-35.

## *Reference* List Box Properties Window




The List Box Properties window contains the following fields/options:

**Name**

Is the name of the list box placeholder.

**Left**

Is the distance in pixels from the left of the layout to the list box's left edge.

**Top**

Is the distance in pixels from the top of the layout to the list box's top edge.

**Width**

Is the list box's width in pixels.

**Height**

Is the list box's height in pixels.

**Overflow**

Determines the behavior of the list box's when the report output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report is too large.

**visible** always displays scrollbars whether or not the report is too large. This may cause objects to overlap. This is the default value.

**hidden** crops the report if it is too large for the report object.

**scroll** displays scrollbars even if the report fits within the report object.

**Border Width**

Is the width of the border. Border Style must be set to a value other than none to set the Border Width value.

**Border Style**

Determines the style of the border. The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's color.

**Back Color**

Is the list box's background color.

**Multiselect**

Determines whether more than one value can be selected. The options are:

**None** does not allow more than one value to be selected. This value is the default.

**And** specifies that all selected parameter values must meet the selection criteria.

**Or** specifies that only one of the selected parameter values must meet the selection criteria.

## Using Check Boxes

Check boxes allow a user to select values from a list of values. Check box lists can allow the user to select single or multiple values at one time:

- A *single select* list allows the user to select only one value each time a request is run.
- A *multi-select* list allows the user to select multiple values by using the Ctrl key while selecting values. In order to provide multiple values, the procedure must be set up to accept multiple values.

Check box list values can be dynamic or static.

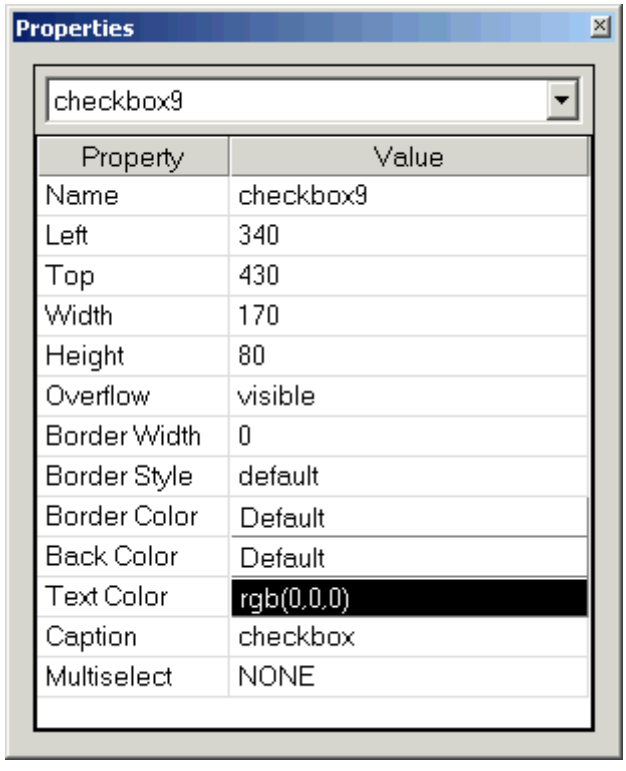
## *Procedure* How to Add Check Boxes

1. Add check boxes by doing one of the following:
   - Click the *Check box* button.
   - From the Insert menu, select *Check box*.

   The cursor changes to a crosshair.
2. Click and drag the crosshair to create a check box list and adjust it to the size you want.

   A check box is created in the layout and assigned the name checkbox*n* where *n* is a number.
3. In the Control field for the parameter you want to associate the check boxes with, select the check box's name from the drop-down list.
4. In the Create list items section of the Parameters window, select static or dynamic values for the check boxes. For details, see *Working With the Parameters Window* on page 6-45.
5. Optionally, check box list's properties by selecting the check box list's name from the Properties window drop-down list, and making your changes. For details, see *Check Box Properties Window* on page 6-40.

## *Reference* Check Box Properties Window




The Check box Properties window contains the following fields/options:

**Name**

Is the name of the check box's placeholder.

**Left**

Is the distance in pixels from the left of the layout to the check box's left edge.

**Top**

Is the distance in pixels from the top of the layout to the check box's top edge.

**Width**

Is the check box's width in pixels.

**Height**

Is the check box's height in pixels.

**Overflow**

Determines the behavior of the check box when the report output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report is too large.

**visible** always displays scrollbars whether or not the report is too large. This may cause objects to overlap. This is the default value.

**hidden** crops the report if it is too large for the report object.

**scroll** displays scrollbars even if the report fits within the report object.

**Border Width**

Is the width of the border. Border Style must be set to a value other than none to set the Border Width value.

**Border Style**

Determines the style of the border.The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's color.

**Back Color**

Is the list box's background color.

**Text Color**

Is the color of the text of the check box options.

**Caption**

Is an descriptive title for the group of check boxes.

**Multiselect**

Determines whether more than one check box can be selected. The options are:

**False** does not allow more than one check box to be selected. This value is the default.

**True** allows more than one check box to be selected.

## Using Radio Buttons

Radio buttons allow a user to select a single value from a list of supplied values. Radio button values can be static or dynamic.

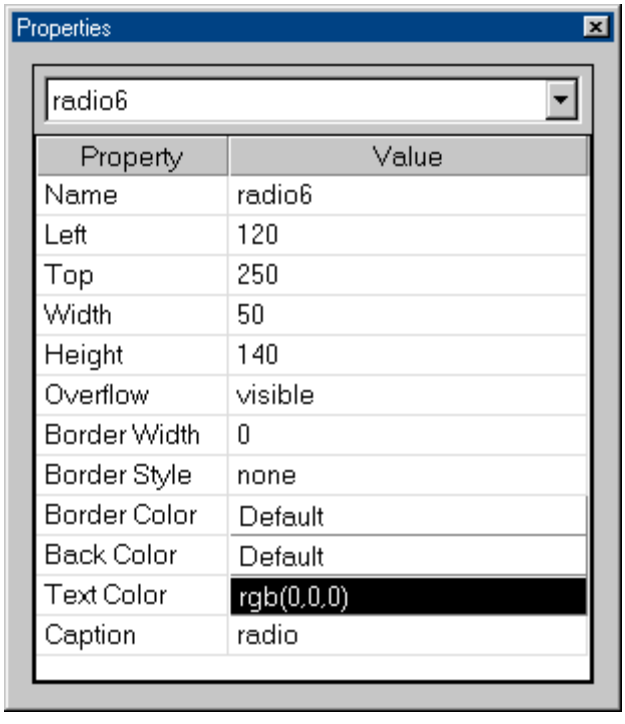
### *Procedure* How to Add Radio Buttons

1. Add radio buttons by doing one of the following:
   - Click the *Radio button*.
   - From the Insert menu, select *Radio button*.

   The cursor changes to a crosshair.
2. Click and drag the crosshair to create a radio buttons list and adjust it to the size you want.

   A radio button placeholder is created in the layout and assigned the name edit*n* where *n* is a number.
3. In the Control field for the parameter you want to associate the radio buttons with, select the radio button's name from the drop-down list.
4. In the Create list items section of the Parameters window, select static or dynamic values for the radio buttons. For details, see *Working With the Parameters Window* on page 6-45.
5. Optionally, change the radio buttons' properties by selecting the radio buttons' name from the Properties window drop-down list, and making your changes. For details, see *Radio Buttons Properties Window* on page 6-43.

## *Reference* Radio Buttons Properties Window




The Radio Buttons Properties window contains the following fields/options:

**Name**

Is the name of the radio buttons placeholder.

**Left**

Is the distance in pixels from the left of the layout to the radio buttons' left edge.

**Width**

Is the radio buttons list's width in pixels.

**Height**

Is the radio buttons list's height in pixels.

**Overflow**

Determines the behavior of the radio buttons list's when the report output is too large for the placeholder. The options are:

**auto** automatically displays vertical and horizontal scrollbars if the report is too large.

**visible** always displays scrollbars whether or not the report is too large. This may cause objects to overlap. This is the default value.

**hidden** crops the report if it is too large for the report object.

**scroll** displays scrollbars even if the report fits within the report object.

**Border Width**

Is the width of the border. Border Style must be set to a value other than none to set the Border Width value.

**Border Style**

Determines the style of the border.The options are: **none**, **solid**, **groove**, **inset**, and **outset**. **none** is the default value.

**Border Color**

Is the border's color.

**Back Color**

Is the radio buttons' background color.

**Text Color**

Is the color of the text of the radio buttons options.

**Caption**

Is an descriptive title for the group of check boxes.

## Working With the Properties Window

The Properties window is a dockable window that consists of the following:

- A drop-down list that contains a list of all objects currently in the layout.
- A Name column that contains the name of each property.
- A Value column that contains the value of each property designated in the Name column.

For details on the properties than can be set for an object, see the topic on that object.

### *Procedure* How to Move the Properties Window

To **undock** the Properties window, click and hold the bar at the top of the window, then move the window to the desired location in the layout.

To **dock** the Properties window, click and hold the Properties title bar, and release in the docking area.

### *Procedure* How to Change a Form's Name

1. Select the form, or select the form's name from the Properties window drop-down list.
2. Single-click the Name's Value field, and enter a new name.

### *Procedure* How to Change a Form's Properties With the Properties Window

1. Select the form whose properties you want to change, or select the form's name from the Properties window drop-down list.
2. Double-click in the Value field next to the property you want to change.
   - If the field contains options for you to choose from, a drop-down list appears.
   - If the field can be any value, the current value becomes highlighted.
   - If you click on a color property, the Color dialog box opens.
   - If you select a font property field, the Font dialog box opens.
3. Make your changes.

## Working With the Parameters Window

The Parameters window is a dockable window consisting of two panes:

- The top pane contains information on the parameters and WebFOCUS form controls in the layout.
- The bottom pane contains information on parameter values.

### *Procedure* How to Move the Parameters Window

To **undock** the Parameters window, click and hold the bar at the top of the window, then move the window to the desired location in the layout.

To **dock** the Parameters window, click and hold the Parameters title bar, and release in the docking area.

## Associating WebFOCUS Form Controls With Parameters

When a parameter is created in Report Painter, a text box is automatically added to the layout and associated with the parameter. However, you can change the form control and form control type associated with the parameter.

### *Procedure* How to Change the WebFOCUS Form Control Associated With a Parameter

In the Parameters window:

1. Double-click the name of the form control listed in the Type column.
2. Select the form control you want to use from the drop-down list.

Your change is reflected in the layout.

### *Procedure* How to Change the Type of WebFOCUS Form Control Associated With a Parameter

1. Double-click the *Control Type* field in the same row as the parameter you want to change the form control type for.
2. Select a type from the drop-down list.

   The options are:

   - *Unassigned* specifies the parameter is not associated with a form control.
   - *Text box*.
   - *List box*.
   - *Drop-down list*.
   - *Radio button*.
   - *Check box*.

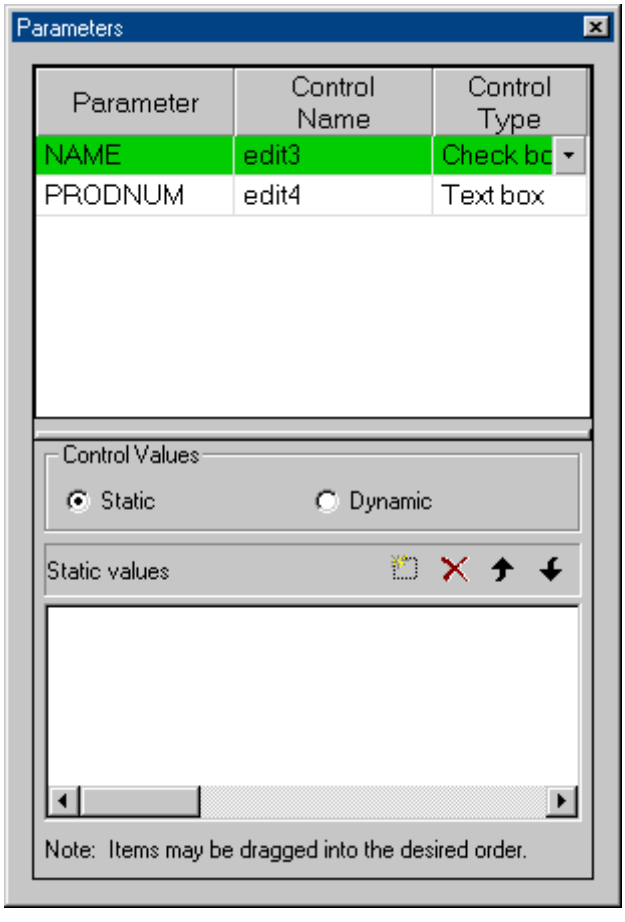
### *Procedure* How to Automatically Delete a Form Control Associated With a Parameter

1. Select *Settings...* from the Tools menu.

   The Settings dialog box opens.
2. In the Other Settings section, select the *Automatically remove input controls if associated parameters are deleted* check box.

You can change this setting at any time if you later choose not to automatically remove the form controls.

## *Reference* Parameters Window (Static list)




The Parameters window contains the following fields/options:

**Parameter**

Contains the name of each parameter in the current procedure.

**Control Name**

Contains the name of the WebFOCUS form control associated with each parameter.

**Control Type**

Contains the type of form control associated with each parameter. The options are:

**Unassigned** specifies the parameter is not associated with a form control.

**Text box**

**List box**

**Drop-down list**

**Radio button**

**Check box**

**Control Values**

Determines whether parameter values are obtained from a static or dynamic list. The options are:

**Static list** uses a static list of parameter values you supply.

**Dynamic list** uses a list of values that are retrieved from a selected data source when the request is executed.

**Static values**

Is a list of supplied values for a static list.

**New**

Creates a new parameter value. For information on creating new parameter values, see *How to Create a Static List of Values* on page 6-50.

**Delete**

Deletes a supplied parameter value from the list.

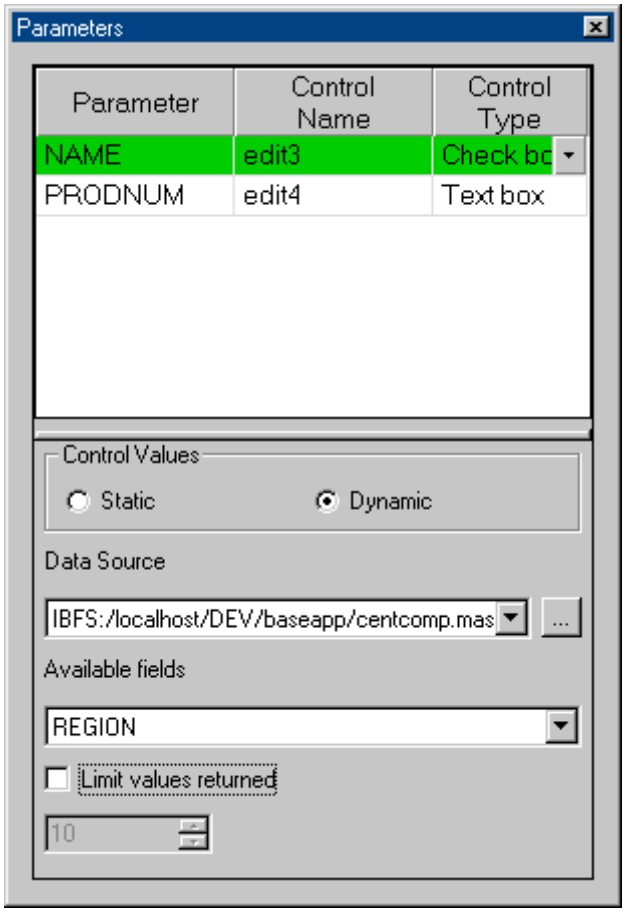
**Move Up**

Moves the selected value up in the list.

**Move Down**

Moves the selected value down in the list.

## *Reference* Parameters Window (Dynamic list)




The Parameters window contains the following fields/options:

**Parameter**

Contains the name of each parameter in the current procedure.

**Control Name**

Contains the name of the WebFOCUS form control associated with each parameter.

**Control Type**

Contains the type of form control associated with each parameter.

**Control values**

Determines whether parameter values are obtained from a static or dynamic list. The options are:

**Static list** uses a static list of parameter values you supply.

**Dynamic list** uses a list of values that are retrieved from a selected data source when the request is executed.

**Data Source**

Is the data source from which the values will be retrieved.

**Available fields**

Is the data source field from which the values will be retrieved.

**Limit values returned**

Specifies that only a certain number of field values will be retrieved from the data source. The number of fields to retrieve is chosen from the spin box below the check box.

## *Procedure* How to Create a Static List of Values

Do the following in the Parameters window:

1. Select *Static list* in the Control Values section.
2. In the Available fields window, select the text and replace it with a value you want included in the list of values.
3. Click the *New* button to create a new value, and enter a value in the box that appears.
4. Repeat Step 3 until the list contains all values you want to include.
5. Optionally, click the *Delete* button to eliminate any values, and the up and down arrows to rearrange the order of the values.

## *Procedure* How to Select a Dynamic List of Values

Do the following in the Parameters window:

1. Select *Dynamic list* in the Control Values section.
2. In the Data Source drop-down list, select the data source you want to retrieve values from, or click the *Browse* button to find a data source not located on the default path. By default, the data source current used in the procedure is made available.
3. In the Available fields drop-down list, select the field whose values you want to populate the form with.
4. Optionally, select *Limit values returned*, and select the number of field values you want to retrieve from the data source from the spin box.

# Styling Your Layout

When you create an HTML page in the Report Layout Painter, the objects in the layout will display in your browser using the browser's default styles. You can customize the appearance of your HTML page by adding a theme or template. A template is provided by Information Builders. A theme is a template accessed through Microsoft FrontPage.

## Adding an HTML Template to the Layout

Templates are supplied by Information Builders and can be added to your layout to determine the look of the Web page. A template is a good way to quickly apply corporate styling to your layout or to assign global styling to a layout. This is an easy method of assigning default styling to the entire layout.

When you add a template to your layout, the image and CSS files it contains will be copied to the local project directory. Sample templates can be found in the ibi\DevStudio52\Templates\Layout directory.

### *Procedure* How to Add an HTML Template to the Layout

1. From the Insert menu, select *HTML Template...*.

   The Get existing html page dialog box opens.

2. Navigate to the directory in which your HTML template resides, select the file, and click *Open*.

## Adding a Theme to the Layout

You can add a Microsoft FrontPage theme to your layout. A FrontPage theme determines the look of the Web page. For more details on FrontPage, see Chapter 7, *Designing a User Interface With FrontPage*.

When you add a template to the Layout, you cannot view it in Design mode. It only appears when you run the procedure or view the layout in Preview mode.

You can edit a template in an editor or in a graphical tool such as FrontPage.

### *Procedure* How to Add a Theme to the Layout

1. From the Insert menu, select *FrontPage Theme...*.

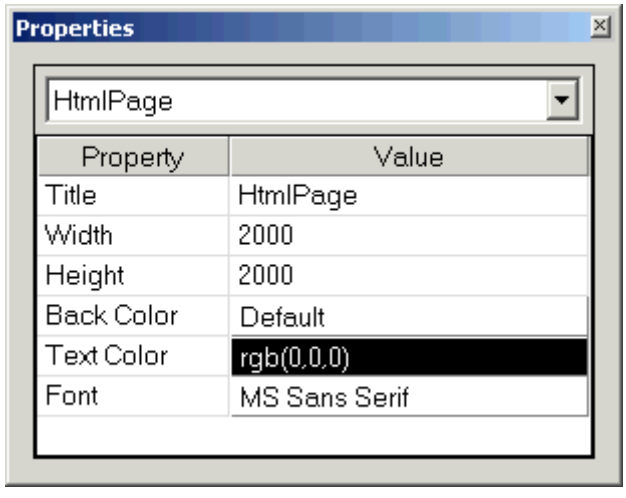
   The Themes window opens.

2. Select a theme from the list box, and click *OK*.

## Setting HTML Page Properties

You can set properties for the HTML page you create in the Report Layout Painter in the HTML Page properties window.

### *Reference* HTML Page Properties Window




The HTML Page properties window contains the following fields/options:

**Title**

Is the title of the HTML page that appears at the top of the browser window.

**Width**

Is the HTML page's width in pixels.

**Height**

Is the HTML page's height in pixels.

**Back Color**

Is the HTML page's background color. The default value is white.

**Text Color**

Is the color of the text.

**Font**

Is the text's font. The default value is default font for your browser unless you set it to a different value.

# Laying Out Objects With the Report Layout Painter

You can change the size and position of objects in your layout in the following ways:

- Set relationships between objects. This is done by setting relationships between objects and a controlling, or dominant, object. The controlling object is the last object selected. For details, see *Setting Relationships Between Objects* on page 6-53.
- Click and drag with your mouse.

## Setting Relationships Between Objects

You can set relationships between objects and a controlling, or dominant, object. The controlling object is the last object or placeholder selected. You can use these relationships to set properties of the selected objects according to the controlling object. Any object on the HTML page layout can be the controlling object.

You can set the following types of relationships:

- Relationships that maintain a distance between objects. This is useful since a report or graph may take up more or less room when your procedure is run than is accounted for in the layout. When setting relationships between objects in this way, you should select the corners of the objects that are closest to each other. This will ensure that the reports do not overlap if either report takes up more room than anticipated.
- Relationships that affect the positioning of objects. This allows you to ensure that objects remain aligned regardless of a change in size or position of one of the objects and that objects remain the same size even when an object's size is changed.

## *Procedure* How to Set Object Size

1. Highlight multiple objects with your mouse by holding the Shift key to select items contiguously, or the Control key to select items non-contiguously. The controlling object is the last object selected.

   The size buttons on the Report Layout Painterbar are active.

2. Click the button in the layout toolbar that corresponds to what you want:

   *Make same width* sets the width of the highlighted objects to the width of the controlling object.

   *Make same height* sets the height of the highlighted objects to the current height of the controlling object.

   *Make same size* sets the height and width of the highlighted objects to the height and width of the controlling object.

All selected objects change in size according to the size of the controlling object. If an image becomes distorted when you resize it, you can restore its original size. Right-click the image, select *Image Properties*, and then *Restore Size*.

## *Procedure* How to Set the Distance Between Objects

1. Highlight multiple elements with your mouse holding the Shift key to select items contiguously, or the Control key to select items non-contiguously. The controlling object is the last object selected.

2. Click the button in the layout toolbar that corresponds to the relationship you want. The selected objects' position will be set according to the controlling object, which is the last object selected. The options are:

   *Relate top left* maintains the distance between the controlling object's top left corner and the highlighted objects' top left corners.

   *Relate top right* maintains the distance between the controlling object's top right corner and the highlighted objects' top left corners.

   *Relate bottom right* maintains the distance between the controlling object's bottom right corner and the highlighted objects' top left corners.

   *Relate bottom left* maintains the distance between the controlling object's bottom left corner and the highlighted objects' top left corners.

   For details on the buttons in the layout toolbar, see *Layout Toolbar* on page 6-55.

## *Procedure* How to Break Relationships

Click *Break* in the toolbar.

## *Reference* Layout Toolbar

The Report Layout Painterbar contains the buttons listed in the following table.

For buttons that use relationships, the relationship is controlled by the controlling object, or dominant object, which is the last object or placeholder selected. These buttons are only highlighted when more than one object is selected.

| Button | Descriptions |
|---|---|
| | Toggles the grid on and off. Use the grid to assist in lining up objects in the layout. This button is recessed when the grids are enabled. |
| | Aligns the highlighted objects' left edge with the left edge of the controlling object. |
| | Aligns the highlighted objects' right edge with the right edge of the controlling object. |
| | Aligns the highlighted objects' top edge with the top edge of the controlling object. |
| | Aligns the highlighted objects' bottom edge with the bottom edge of the controlling object. |
| | Sets the width of the highlighted objects to the width of the controlling object. |
| | Sets the height of the highlighted objects to the current height of the controlling object. |
| | Sets the height and width of the highlighted objects to the height and width of the controlling object. |
| | Maintains the distance between the controlling object's top left corner and the highlighted objects' top left corners. |
| | Maintains the distance between the controlling object's top right corner and the highlighted objects' top left corners. |
| | Maintains the distance between the controlling object's bottom right corner and the highlighted objects' top left corners. |

| Button | Descriptions |
|---|---|
| | Maintains the distance between the controlling object's bottom left corner and the highlighted objects' top left corners. |
| | Breaks the relationship set between highlighted objects. This button is only highlighted when selected objects have relationships set. |
| | Toggles the display of arrows illustrating the relationships between objects. This button is only highlighted when selected objects have relationships set. This button will stay highlighted when relationships are displayed. It will not be highlighted when relationships are not displayed. |

# Customizing the Layout

All HTML pages created in the Report Layout Painter can be edited in any HTML editor. The changes made to a layout in an editor will be preserved when you re-enter the Report Layout Painter.

You can make changes to existing HMTL code, but if you add new HTML objects, they may not appear on your HTML page. This occurs because the code between the <!--startibiitems--> and <!endibiitems--> tags, and the <!--startibilines--> and <!endibilines--> tags are regenerated at run time.

# Controlling the Report Layout Painter Environment

You can set properties for the Report Layout Painter with the Settings dialog box. This dialog box allows you to set page properties such as the location of reports and graphs, as well as the display of a grid in your layouts. The changes that are made in the Settings dialog box are saved to the registry. This allows a customized environment to be created for various users.

*Procedure* **How to Set Page Properties**

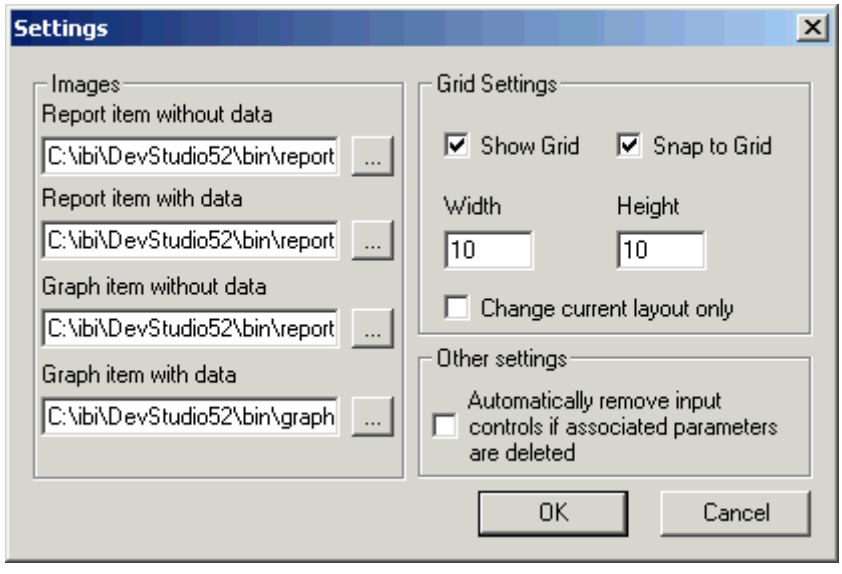
1. Do one of the following:
   - Right-click any area of the layout window and select *Page Properties* from the pop-up menu.

   or

   - Select *Settings...* from the Tools menu.

   The Settings dialog box opens. For details, see *Settings Dialog Box* on page 6-57.
2. Make your changes, and click *OK*.

## *Reference* Settings Dialog Box




The Page Properties and Settings dialog boxes contains the following fields/options:

**Images**

Specifies the default image location of the report and graph object.

**Report Item without data**

Is the image location for the report object when no report has been created for it.

**Report Item with data**

Is the image location for the report object after a report has been created for it.

**Graph Item without data**

Is the image location for the graph object when no graph has been created for it.

**Graph Item with data**

Is the image location for the graph object after a graph has been created for it.

**Grid Settings**

Determines the grid display.

**Show Grid**

Displays a grid. If this is not selected, the grid is turned off for all layouts.

**Snap to Grid**

Causes objects in the layout to snap to grid lines when being positioned.

**Width**

Is the width of the grid in pixels.

**Height**

Is the height of the grid in pixels.

**Change current layout only**

Applies your settings only to the current layout.

**Other Settings**

**Automatically remove input controls if associated parameters are deleted**

Automatically deletes a form control associated with a parameter when that parameter is deleted in the Report Painter.

CHAPTER 7

# Designing a User Interface With FrontPage

**Topics:**

- Creating a Launch Page Overview
- Creating a Main Page

You can use Microsoft FrontPage to create a user interface. The installation procedure for Developer Studio configures support for Microsoft FrontPage 2000. It adds custom menus and controls to FrontPage, facilitating the creation of launch pages for your WebFOCUS applications.

FrontPage enables you to create a launch page graphically. You can preview or run a completed launch page before deploying it to a Web server for user access. You can also modify a launch page to reflect a new design; simply reopen it in FrontPage. The FrontPage HTML editor enables you to access source code and add JavaScript for additional design capabilities.

FrontPage integration with Developer Studio allows you to create a launch page for WebFOCUS. In FrontPage you can also edit a launch page created in another environment. At runtime, all pages are processed identically.

Other FrontPage development components, including FrontPage Explorer, FrontPage Web Publisher, FrontPage Personal Web Server, and Microsoft Image Composer, are not supported with this integration.

The Managed Reporting Administrator provides FrontPage functionality, enabling you to create HTML forms from the Managed Reporting Environment (MRE). For information on using FrontPage from the MRE, see Chapter 2, *Developing and Administering Web Repositories*, in the *Administrator's Reference* manual.

# Creating a Launch Page Overview

Make sure that Microsoft FrontPage 2000 is installed on your desktop before you proceed. You must install FrontPage before Developer Studio.

Here is how you create a new launch page:

- Supply a name for the page and select a type—single page or frameset.
- Add a WebFOCUS form to the launch page. The default WebFOCUS form contains a Clear and Run button. You can view the form properties and edit some of them.

  Alternatively, you can add a hyperlink to the launch page. A hyperlink does not require the insertion of a WebFOCUS form.

  Optionally add controls to the form, such as radio buttons and drop-down menus, and specify their properties to enable interactivity.
- Use the WebFOCUS form with its optional controls as a completed launch page, or customize the page further using FrontPage design features. When customizing a launch page, maximize the efficiency of WebFOCUS integration by avoiding Microsoft FrontPage Server Extensions. See the Microsoft FrontPage 2000 online help for details on extensions.You must place a form on a launch page before placing a control, with the exception of a hyperlink.

## *Procedure* How to Start a Launch Page

1. Create the procedure for the project if you have not done so. In Developer Studio, right-click the HTML Forms folder for the project, and select *New* from the pop-up menu, then *HTML File* from the next menu.

   The New HTML form file dialog box opens.

   You can also access this dialog box from TED. Open the procedure in the text editor, and right-click a blank line. Select *New* from the pop-up menu, and then select *HTML Form*.
2. On the Page tab, type a name for the HTML file (launch page) in the Name of new file field. You must supply a unique name. WebFOCUS automatically appends the extension .HTM to the name when it saves the file.
3. Do one of the following:
   - **To start a single launch page.** On the Page tab, select a template or accept the default blank report template, and click *OK*. The template is copied to a new file in the default project path, and it opens in FrontPage.
   - **To start a frameset.** On the Frames tab, select a template or accept the default (IBI Banner and Contents), and click *OK*. The files in that template are copied to the default project path, and they open in FrontPage.

4. To exit FrontPage, select *Close* from the File menu, and click *Yes* to save the changes.

**Tip:** After you save and close a launch page, you can reopen and edit it. Open the HTML Forms folder, right-click the name of the launch page, and select Open.

## *Procedure* How to Add a WebFOCUS Form

A launch page requires a form to collect user input.

1. In Developer Studio, open the HTML Forms folder for the project. Right-click the name of the launch page, and select *Open* from the pop-up menu. The launch page opens in FrontPage.
   a. On the FrontPage Insert menu, select *WebFOCUS Form*, and then select *Form*.
   b. The Insert WebFOCUS Form dialog box opens. It displays default information.
2. Set the WebFOCUS form properties:
   a. In the Current Project field, accept the project name shown, or select a new one. To select a new one, click *Change*, and choose from the drop-down list. The launch page will be saved to the selected project.
   b. In the Form Name field, type a name for the form, or accept the default.
   c. In the FOCEXEC to run field, select a procedure from the drop-down list, or accept the default.
   d. If you are using a frameset, click the pencil button to open the Target Frame dialog box. Select the frame that will hold the report output.
   e. If you are using a single page, you can select one of the available options.
   f. Click *OK* on the Insert WebFOCUS Form dialog box. Clicking *Cancel* exits the dialog box without creating a form.
3. The form is saved with the defined properties, and a Run and Clear button are added. If the launch page is usable in your project as is, you do not need to develop it further.
4. Optionally, you can add a variety of WebFOCUS controls that collect user input.

## *Procedure* How to Edit WebFOCUS Form Properties

1. Open the launch page in FrontPage. Place the cursor in a WebFOCUS form.
2. On the FrontPage Insert menu, select *WebFOCUS Form*, and then select *Form Properties*.

   The WebFOCUS Form Properties dialog box opens.
3. You can change the Form Name or Target Frame. You cannot change the FOCEXEC to run.

## *Procedure* How to Add a One-Line Text Box, Check Box, or Radio Button

1. Open the launch page in FrontPage. Place the cursor in a WebFOCUS form.
2. On the FrontPage Insert menu, select *WebFOCUS Form*.

   Select *One-Line Text Box*, *Check Box*, or *Radio Button*.

   The control is added to the form.
3. Set the control properties:

   **a.** Double-click the control. The Design-Time Control Properties dialog box opens.

   **b.** In the FOCUS Parameter field, type the variable (parameter) in the procedure that requires a value, or click the down arrow and select from the drop down list.

   **One-Line Text Box.** In the Initial value field, type the value that displays as the default when a user opens the launch page, or accept the default Initial value.

   **Check Box.** In the Value field, assign a value to the check box.

   **Radio Button.** In the Value field, assign a value to the radio button.

   **c.** In the Width in characters field, type the maximum number of characters the user can enter (one-line text box). You can accept the default of 20. This option does not apply to a check box or radio button.

   **d.** Type a number (1 - 999), indicating the tab order for the selected control. For example, type 3 to access the control when you press the Tab key three times. If you specify the same number for two or more controls, the control that is first on the page will come first in the tab order. To remove the control from the tab order, type -1.

   **e.** Additionally, for:

   **One-Line Text Box.** Click *Yes* next to the Password field to enable password entry. When a user types the password, it displays as asterisks for security. A password is only validated when you use the Registration Form Handler or a custom script to handle the form results. For more information, see the Microsoft FrontPage 2000 online help.

   **Check Box or Radio Button.** Click *Checked* (check box) or *Selected* (radio button) for the Initial State if you want the control to be selected by default when a user opens the launch page. Otherwise, accept the default Initial State.

   The Object Tag and Parameters tabs do not apply to these types of controls.

   **f.** Click *OK* on the dialog box.
4. Save the launch page, and exit FrontPage. In Developer Studio, run the page to test it.

## *Procedure* How to Add a Dynamic or Static Drop-Down Menu

1. Open the launch page in FrontPage. Place the cursor in a WebFOCUS form.
2. On the FrontPage Insert menu, select *WebFOCUS Form*.

   Select *Dynamic Drop-Down Menu* or *Static Drop-Down Menu*.

   A drop-down menu is added to the form.
3. Set the drop-down menu properties.

   Double-click the drop-down menu. The Design-Time Control Properties dialog box opens.

   **Dynamic Drop-Down Menu:**

   **a.** In the FOCUS Parameter field, type the variable (parameter) in the procedure that requires a value, or click the down arrow and select from the drop-down list.

   **b.** Select the data source from the List data from file options. The dynamic drop-down menu will be populated by field values from this data source.

   **c.** Select the Choice field. The dynamic drop-down menu will be populated by values for this field.

   **d.** Identify which values to use:

   Select *Choice field*. This is the field identified in the previous step.

   or

   Select *Another field*. This is the field identified in the next step.

   **e.** If you select *Another field*, the Value field drop-down list is enabled. Select another field from the identified data source. The dynamic drop-down menu will be populated by values for this field.

   **f.** In the Where field, type a WHERE command to limit the data retrieved from the request, or click the *Assist* button to formulate your selection criteria.

   **g.** You can chain (append) the WHERE selection criteria from another dynamic drop-down menu to the current dynamic drop-down menu. To chain criteria, select a WHERE command from the Chain the WHERE to drop-down list. If there are no other WHERE commands, this option is disabled.

   Developer Studio does not support the chain option.

**Static Drop-Down Menu:**

**a.** In the FOCUS Parameter field, type the variable (parameter) in the procedure that requires a value, or click the down arrow and select from the drop-down list.

**b.** Click *Add* to open the Add Choice dialog box.

**c.** In the *Choice* field, type the text you want to display in the drop-down menu.

**d.** If the actual field value is different from the text displayed, select *Specify value, and* then type the value.

**e.** Click *Selected* under Initial State to make the specified value the default when the launch page opens. Otherwise, accept the default value for the Initial State.

**f.** Click *OK*.

Repeat these steps for each value you want to add. You can change the order in which values are displayed by clicking a value in the list, and then clicking the *Move Up or Move Down* button.

**4.** In the Height field, type a number indicating how many values will display in the drop-down menu at one time.

**5.** For Allow multiple selections, click *Yes* if the user can make more than one choice from the drop-down menu. If you accept the default value No, the user can make only one choice.

The multiple selection feature requires certain Dialogue Manager commands in the procedure. See Chapter 8, *Coding a User Interface*, for an example.

**6.** Type a number (1 - 999), indicating the tab order for the selected control. For example, type 3 to access the control when you press the Tab key three times. If you specify the same number for two or more controls, the control that is first on the page will come first in the tab order. To remove the control from the tab order, type -1.

The Object Tag and Parameters tabs do not apply to these types of controls.

**7.** Click *OK* on the dialog box.

**8.** Save the launch page, and exit FrontPage. In Developer Studio, run the page to test it.

**Note:** Test a dynamic drop-down menu in WebFOCUS. It requires population on the Reporting Server.

## *Procedure* How to Add a Push Button

1. Open the launch page in FrontPage. Place the cursor in a WebFOCUS form.
2. On the FrontPage Insert menu, select *WebFOCUS Form*, and then select *Push Button*.

   A push button is added to the launch page.
3. Set the push button properties:
   - **a.** Double-click the push button. The Design-Time Control Properties dialog box opens.
   - **b.** In the Name field, type a name for the push button.
   - **c.** In the Value/label field, type a label for the push button as it will appear on the launch page.
   - **d.** Indicate the push button type:

     *Run* - Executes the form.

     *Clear* - Clears all fields on the current form, or resets the value of each field to the default.

     *Execute* - Calls an external procedure. Developer Studio does not support this option.

     *Perform Form* - Opens another HTML launch page. Click *Browse f*or a list of available pages.

     *Inline FOCEXEC* - Allows the user to type commands and edit a procedure in the text editor. Developer Studio does not support this option.

     *Close Form* - Closes the current browser window. Developer Studio does not support this option.
   - **e.** Type a number (1 - 999), indicating the tab order for the selected control. For example, type 3 to access the control when you press the Tab key three times. If you specify the same number for two or more controls, the control that is first on the page will come first in the tab order. To remove the control from the tab order, type -1.

     The Object Tag and Parameters tabs do not apply to this control.
   - **f.** Click *OK* on the dialog box.
4. Save the launch page, and exit FrontPage. In Developer Studio, run the page to test it.

## *Procedure* How to Add a Hyperlink

1. Open the launch page in FrontPage. Click the cursor where you want to place the hyperlink. You can add a hyperlink inside or outside a WebFOCUS form.
2. On the FrontPage Insert menu, select *WebFOCUS Form*, and then select *Hyperlink*.

   The text WebFOCUS appears on the form. Double-click it to open the Design-Time Control Properties dialog box.
3. In the Current Project field, accept the default project shown, or select a new one. To select a new one, click *Change*, and choose from the drop-down list. The launch page will be saved to the selected project.
4. In the Focexec to run field, type the name of the procedure to run, or select a name from the drop down list.
5. In the Text to display field, enter the label that will identify the hyperlink on the launch page. Do not exceed 68 characters.
6. In the FOCUS Parameters field, add any fixed values that must be passed to the procedure:

   **a.** Click *Add*. On the Add Amper Variable dialog box, click the down arrow for the FOCUS Parameter(s) field, and select the parameter.

   In the Initial Value field, type the value that will be passed to the procedure.

   Click *OK* on the dialog box.

   **b.** Repeat step 6.a. for each parameter required by the procedure.

   **c.** To modify a parameter, select the parameter, and click *Modify*.

   **d.** To remove a parameter from the list, select the parameter, and click *Remove*.
7. **For a frameset:** In the Target Frame field, type the name of the target frame that will hold the report output, or click the pencil button to open the Target Frame dialog box and select a frame.

   **For a single page:** You can select one of the available options.
8. The Object Tag and Parameters tabs do not apply to this control.

   Click *OK* on the dialog box.
9. Save the launch page, and exit FrontPage. In Developer Studio, run the page to test it.

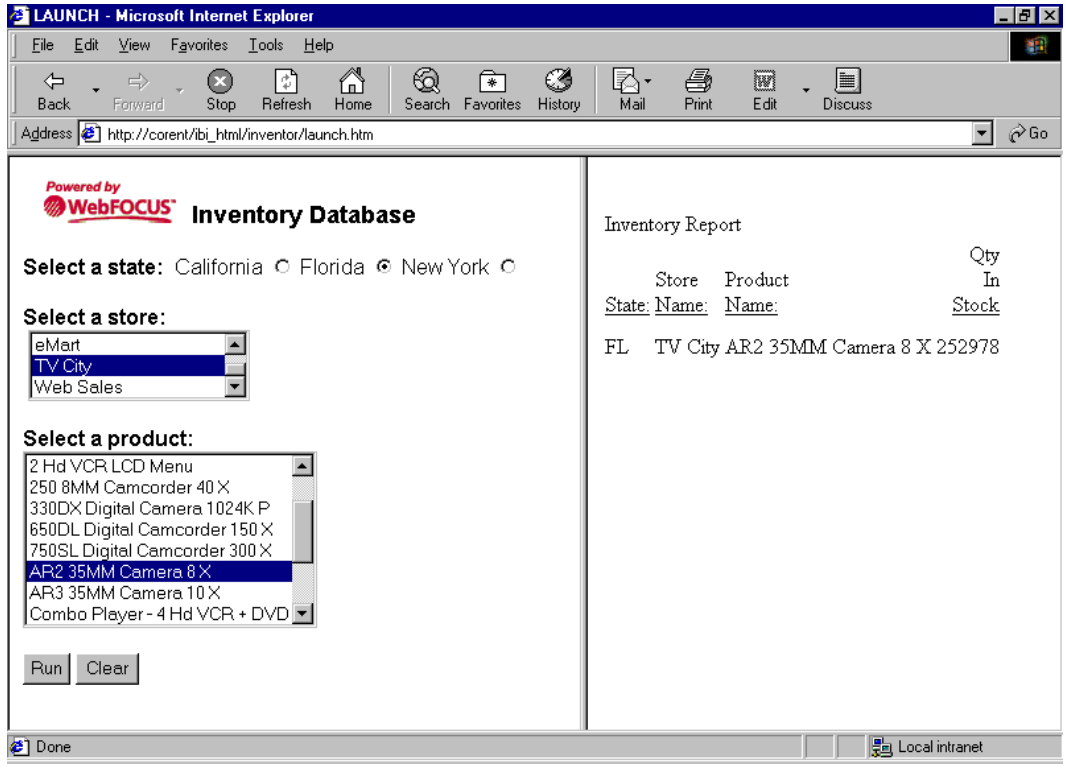
## Creating a Sample Launch Page for an Inventory Report

The examples in this topic create a sample two-frame launch page, which runs an inventory report. A variety of controls in the left frame prompt for values that determine the content of the report. The report displays in the right frame.

The sequence of tasks is:

- Create a project, and create and test a procedure in the project.
- Start a launch page in FrontPage.
- Add a WebFOCUS form to the launch page.
- Add radio buttons to the form.
- Add a static drop-down menu to the form.
- Add a dynamic drop-down menu to the form.
- Style the launch page and report.

The following is a completed sample, displayed in a browser:



## *Example* Creating a Project and Procedure

These steps set up the environment and create the necessary files for your work in FrontPage.

1. Create a new project named Inventory. If required, see Chapter 3, *Creating a Reporting Application*, for instructions.
2. Create a procedure named IN_STOCK.FEX. If required, see Chapter 4, *Creating a Reporting Procedure*, for instructions.

   The procedure reports on quantity in stock for the state (STATE), store (SNAME), and product (PRODNAME) of interest. Using radio buttons, a static drop-down menu, and a dynamic drop-down menu, the finished launch page will prompt the user for the values.

   The code is:

   ```
   TABLE FILE CENTORD
   SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
   ON TABLE SUBHEAD
   "Inventory Report"
   WHERE STATE EQ '&STATE'
   WHERE SNAME EQ '&SNAME'
   WHERE PRODNAME EQ '&PRODNAME'
   END
   ```

3. Make sure that you test the procedure before proceeding. One way to test it is to run it remotely from the desktop. To do that, right-click the procedure, and select *Flag as Remote* from the pop-up menu. Then click *Run* from the same menu.

**4.** In Developer Studio, WebFOCUS prompts for the three required values. Remember, when you complete the launch page in FrontPage, it will contain controls that prompt for the values.

For now, to make sure the procedure runs correctly, enter the following, in the exact case (upper or lower) shown.

For the state (STATE), enter:

```
CA
```

For the store (SNAME), enter:

```
eMart
```

For the product (PRODNAME), enter:

```
2 Hd VCR LCD Menu
```

Enter server credentials if prompted. You may need to do this throughout the examples, depending on your connection status with the server.

You should receive a report like the following. If you do not, check the code in the procedure and retry.

| PAGE 1 | | | |
|---|---|---|---|
| Inventory Report | | | |
| State: | Store Name: | Product Name: | Quantity In Stock: |
| CA | eMart | 2 Hd VCR LCD Menu | 1162836 |

**5.** To continue, go to the example *Starting a Launch Page* on page 7-12.

## *Example* Starting a Launch Page

You are going to create a two-frame launch page.

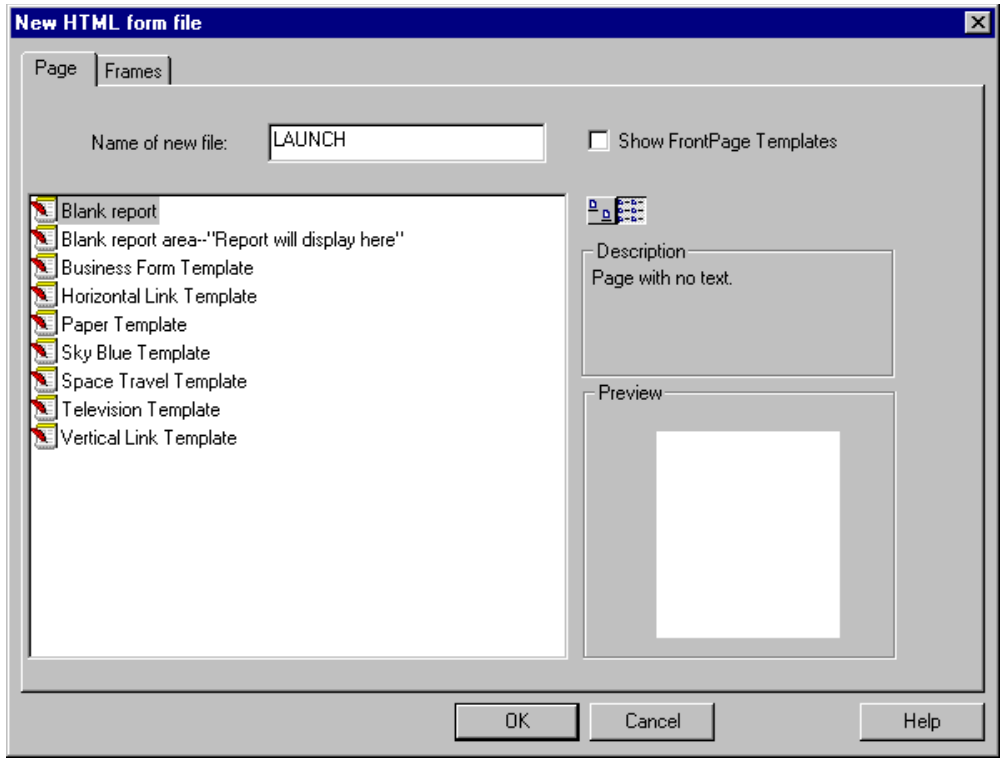
1. Complete the example *Creating a Project and Procedure* on page 7-10 if you have not done so.

   In Developer Studio, right-click the HTML Forms folder for the Inventory project, and select *New* from the pop-up menu, then *HTML File* from the next menu.

   The New HTML form file dialog box opens.

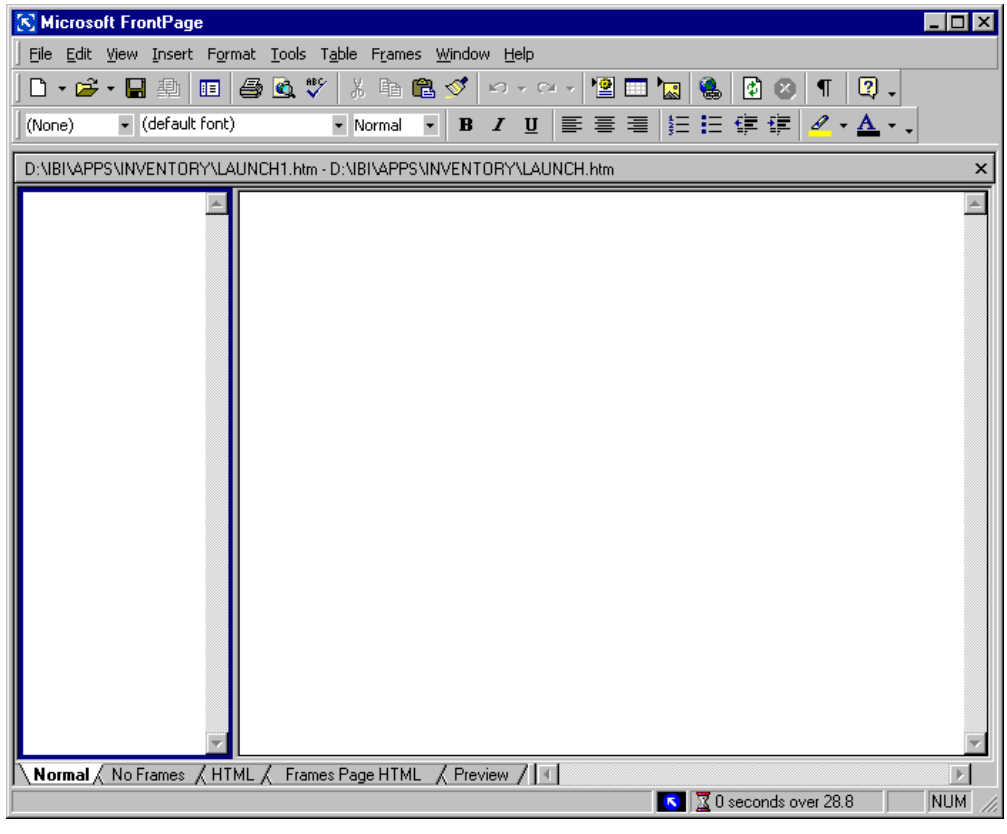
2. On the Page tab, enter *LAUNCH* for the HTML file (the launch page) in the Name of new file field.

   WebFOCUS automatically appends the extension .HTM when it saves the file.



3. The launch page is a frameset with two frames, so click the *Frames* tab. The controls that prompt for user input will appear in the left frame. The report will display in the right frame.

   On the Frames tab, click the *IBI Contents* template and *OK*. The template is copied to a new file in the default path for the project, which in this example is D:\IBI\APPS\INVENTORY. The template then opens in FrontPage.




4. The file for the entire frameset is named LAUNCH.HTM. Click in the left frame. The file for the left frame is named LAUNCH1.HTM.

   Click in the right frame. The file for that frame is LAUNCH2.HTM.

5. To continue, go to the example *Adding a WebFOCUS Form* on page 7-14.

## *Example* Adding a WebFOCUS Form

Complete the example *Starting a Launch Page* on page 7-12, if you have not done so.

You are going to add a WebFOCUS form in the left frame. It will contain a Run button that executes the procedure, and a Clear button that clears the window of current values. Remember, the left frame will also contain the controls that prompt for user input, and the right frame will contain the report.

The default name for the left frame is contents. The default name for the right frame is main.

**1.** Click in the left frame.

On the FrontPage Insert menu, select *WebFOCUS Form*, and then select *Form*.

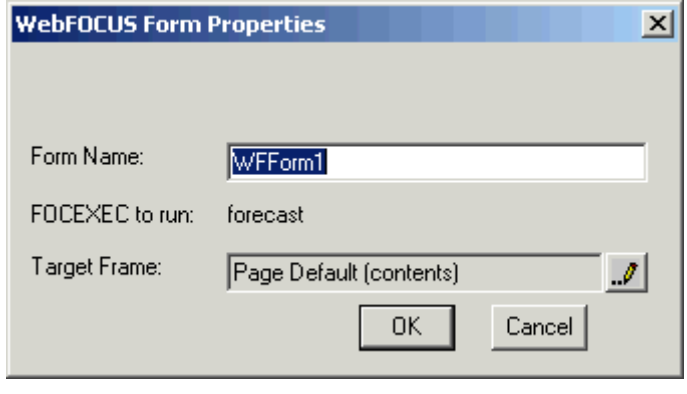
The Insert WebFOCUS Form dialog box opens. It displays default information.

**2.** Set the WebFOCUS form properties:

**a.** For the Current Project, click *Change*, and select *Inventory.gfa* from the drop-down list. The launch page will be saved to that project. If Inventory.gfa is the default, simply accept it.

**b.** For the Form Name, accept the default, *WFForm1*.

**c.** For the FOCEXEC to run, select *IN_STOCK.FEX* from the drop-down list.

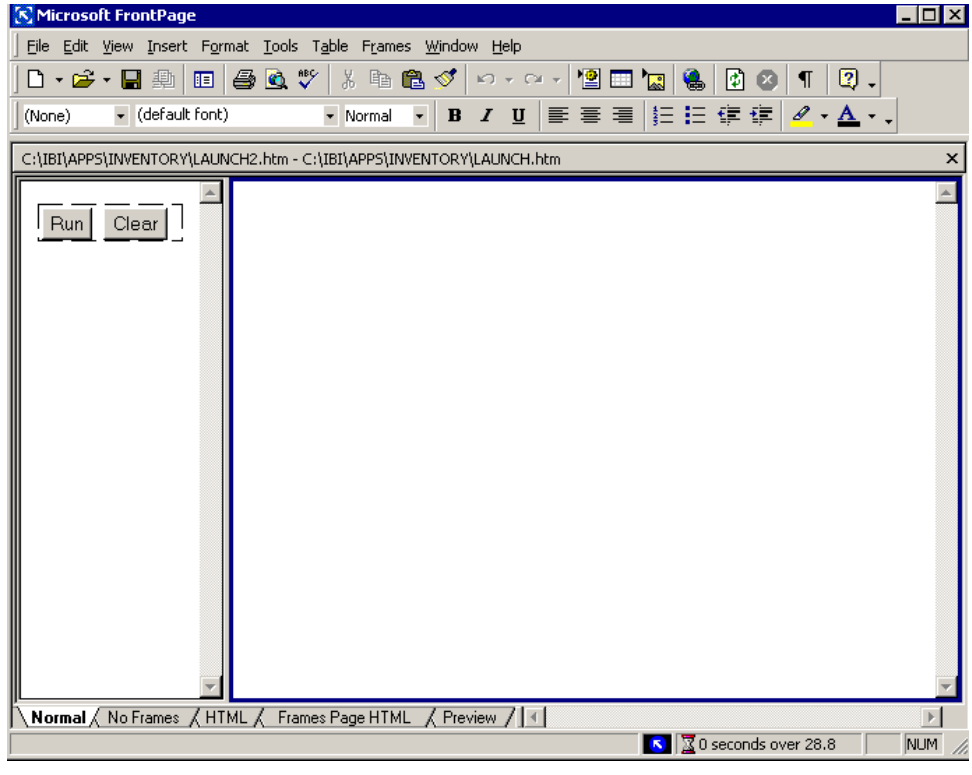


**d.** Click the pencil button to the right of the Target Frame field.

In the Target Frame dialog box, select *main* as the target frame (the frame that will contain the report), and click *OK*.

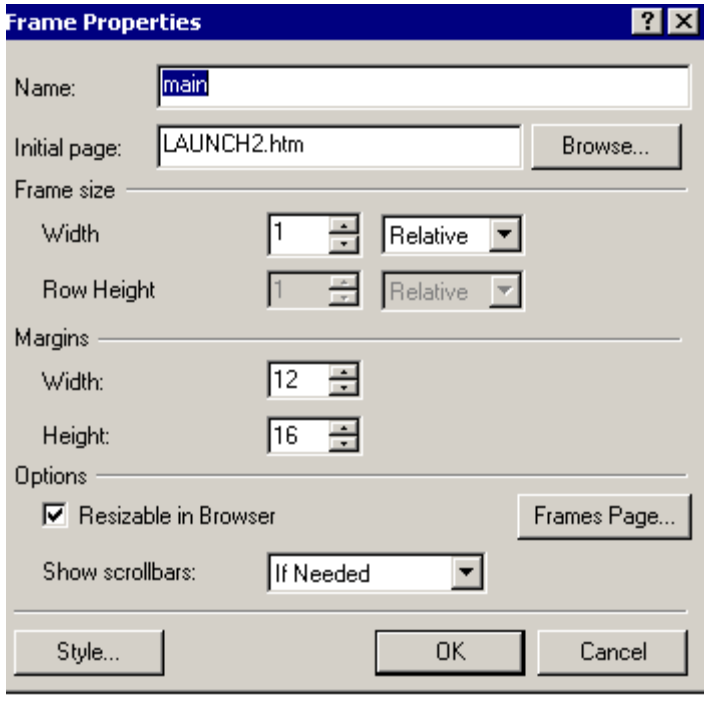
Click *OK* on the Insert WebFOCUS Form dialog box.

WebFOCUS saves the form with the defined properties, and adds a Run and Clear button to the left frame. Later, you will add the WebFOCUS controls that collect the user input required by the procedure.



3. Right-click in the right frame (main), and select *Frame Properties* from the pop-up menu.

   You can use this dialog box to adjust the size and other characteristics of the frame. For now, accept the default values, and click *OK*.




4. Select *Close* from the File menu. Click *Yes* when asked if you want to save the changes.

5. To run the form to see how it looks so far, close FrontPage and return to Developer Studio. Open the HTML Forms folder for the Inventory project, and right-click LAUNCH.HTM. Select *Run* from the pop-up menu.

6. On the launch page in the browser, click *Run* to execute the procedure. WebFOCUS prompts you for the required values.

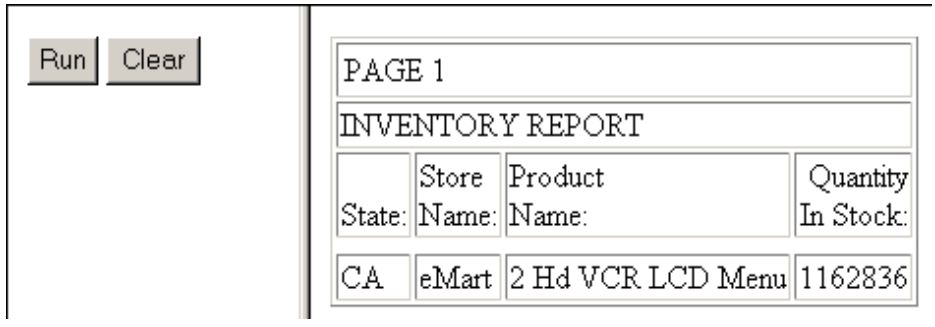
   For the state (STATE), type *CA*.

   For the store (SNAME), type *eMart*.

   For the product (PRODNAME), type *2 Hd VCR LCD Menu*.

   **Caution:** Enter the data using the exact case (upper or lower) shown.

7. The report runs in the right frame in the browser. When you complete the launch page and deploy it with supporting files, the report will run in the right frame in the browser.




8. To continue, go to the example *Adding Radio Buttons* on page 7-17.

## *Example* Adding Radio Buttons

Complete the example *Adding a WebFOCUS Form* on page 7-14 if you have not done so.

You are going to add radio buttons that prompt the user for the state. In this example, the user can select California, Florida, or New York.

1. In Developer Studio, open the launch page named LAUNCH.HTM in FrontPage. In the left frame, place the cursor in the WebFOCUS form, at the end of the line with the Run and Clear buttons. You must position the cursor *inside* the form.

   Press Enter to add a blank line. Type *Select a state:*.

   Press *Enter* again. Type the label for the first radio button directly on the form: *California*.

2. On the FrontPage Insert menu, select *WebFOCUS Form*.

   Select *Radio Button*.

   The radio button is added to the form.

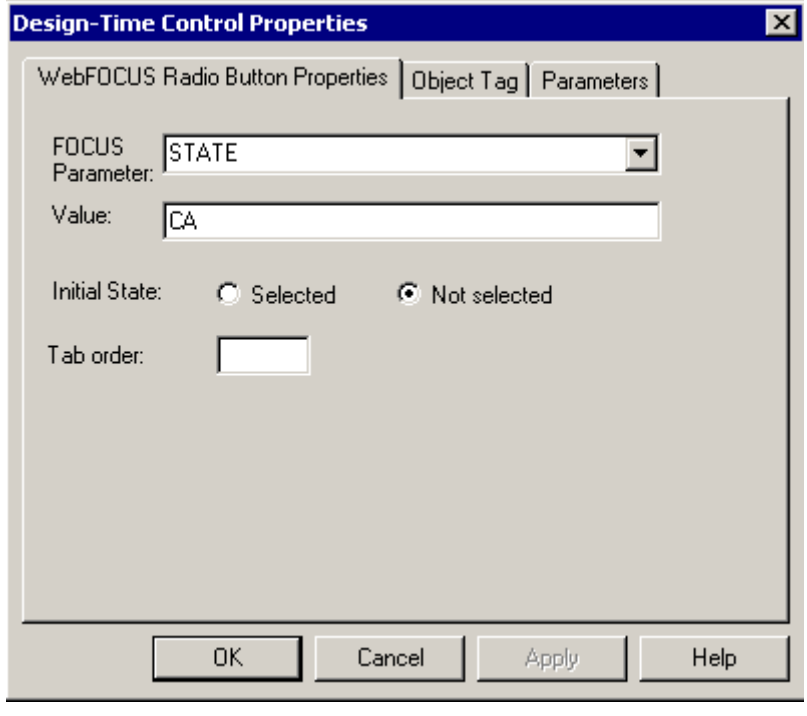
3. Set the radio button properties:

   a. Double-click the radio button. The Design-Time Control Properties dialog box opens.

   b. In the FOCUS Parameter field, click the down arrow, and select *STATE* from the drop down list.

   c. In the Value field, type *CA*.

      In this example, you are not going to assign a default value for the STATE, so leave the Not selected radio button as is.




   d. Click *OK*.

4. Press *Enter* after the radio button to add a blank line. Type the label for the second radio button: *Florida*.

5. On the FrontPage Insert menu, select *WebFOCUS Form*.

   Select *Radio Button*.

   The radio button is added to the form.

6. Set the radio button properties:
   a. Double-click the radio button. The Design-Time Control Properties dialog box opens.
   b. In the FOCUS Parameter field, click the down arrow, and select *STATE* from the drop down list.
   c. In the Value field, type *FL*.
   d. Click *OK*.
7. Press *Enter* after the second radio button to add a blank line. Type the label for the third radio button: *New York*.
8. On the FrontPage Insert menu, select *WebFOCUS Form*.

   Select *Radio Button*.

   The radio button is added to the form.
9. Set the radio button properties:
   a. Double-click the radio button. The Design-Time Control Properties dialog box opens.
   b. In the FOCUS Parameter field, click the down arrow, and select *STATE* from the drop down list.
   c. In the Value field, type *NY*.
   d. Click *OK*.
10. Save the form and exit FrontPage.

**11.** In Developer Studio, right-click the form named LAUNCH.HTM. Click *Run* on the pop-up menu.

In the radio button group that prompts for the state, click *California*.

Click *Run* on the launch page.

**12.** When prompted for the store (SNAME), type *eMart*.

Click *OK*.

When prompted for the product (PRODNAME), type *2 Hd VCR LCD Menu*.

Click *OK*.

**Caution:** Enter the data using the exact case (upper or lower) shown.

**13.** The report runs in the right frame in the browser.

**14.** To continue, go to the example *Adding a Static Drop-Down Menu* on page 7-21.

## *Example* Adding a Static Drop-Down Menu

You are going to add a static drop-down menu that prompts the user for the store. In this example, the user can select from Audio Expert, Consumer Merchandise, eMart, TV City, or Web Sales.

1. Complete the example *Adding Radio Buttons* on page 7-17 if you have not done so.

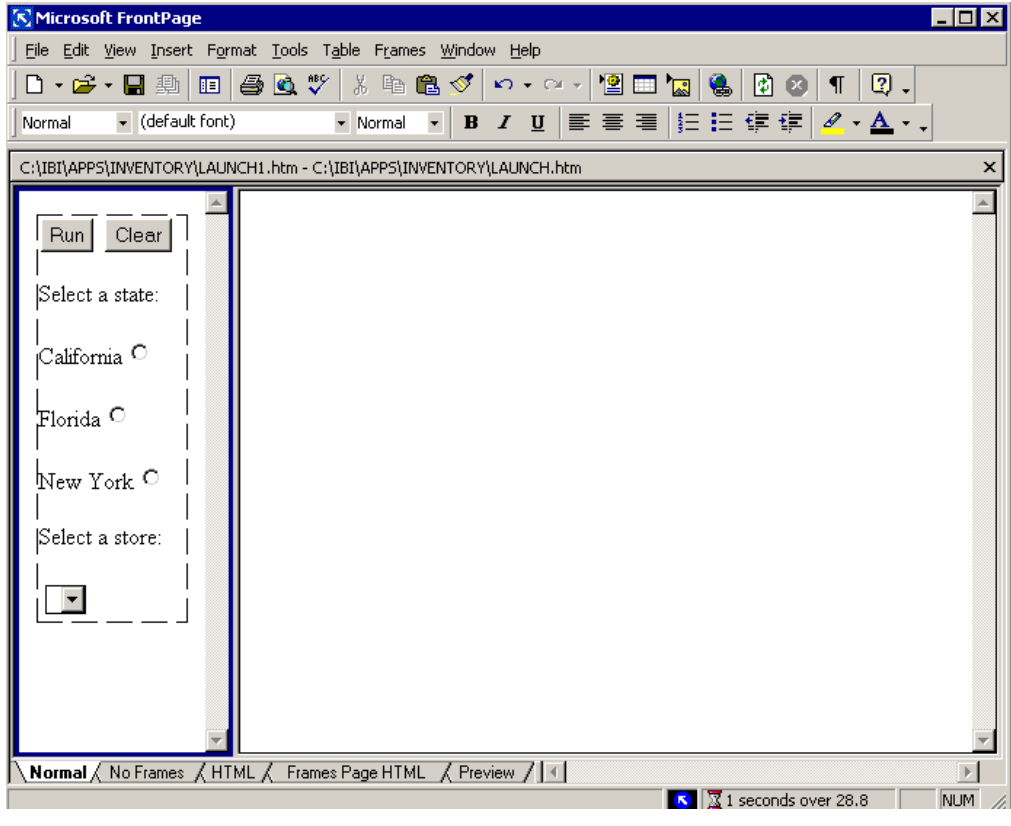
   In the HTML Forms folder, open the launch page named LAUNCH.HTM in FrontPage. After the third radio button (labeled New York), press *Enter* to add a blank line. Type *Select a store*.

   Press Enter again.

2. On the FrontPage Insert menu, select *WebFOCUS Form*.

   From the drop-down list, select *Static Drop-Down Menu*.

   A drop-down menu is added to the form.

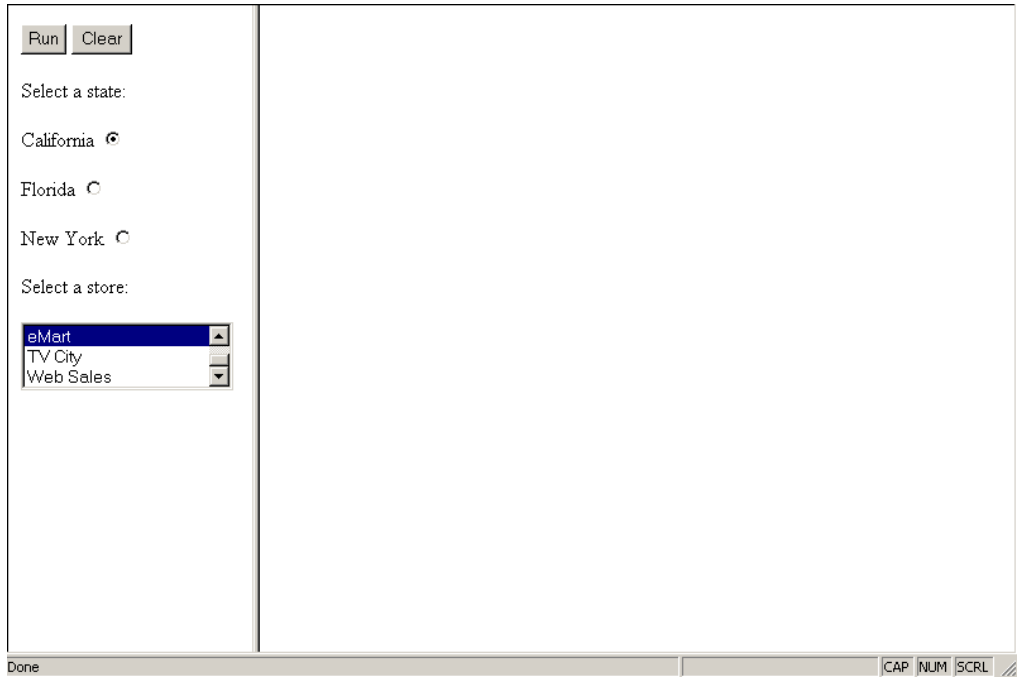


3. Set the drop-down menu properties:
   a. Double-click the drop-down menu. The Design-Time Control Properties dialog box opens.
   b. In the FOCUS Parameter field, select *SNAME* from the drop-down list.
   c. Click *Add*. The Add Choice dialog box opens.
   d. In the Choice field, type *Audio Expert*.
   e. Click *OK*.
4. Repeat step 3.c., typing each of the following values in the Choice field:
   - Consumer Merchandise
   - eMart
   - TV City
   - Web Sales

   These values will be listed in the static drop-down menu. None of the values will be selected by default.
5. In the Height field, type the number *3*, indicating that three values will display at one time in the drop-down menu.
6. Click *OK* on the Design-Time Control Properties dialog box.
7. Save the form and exit FrontPage.
8. In Developer Studio, right-click the form named LAUNCH.HTM. Select *Run* from the pop-up menu.

**9.** For the state, click the radio button labeled *California*.

For the store, select *eMart* from the static drop-down menu.




Click *Run* on the launch page.

**10.** When prompted for the product (PRODNAME), type *2 Hd VCR LCD Menu*.

Click *OK*.

**11.** The report displays in the right frame of the browser.

**12.** To continue, go to the example *Adding a Dynamic Drop-Down Menu* on page 7-24.

## *Example* Adding a Dynamic Drop-Down Menu

You are going to add a dynamic drop-down menu populated by data source values retrieved live at runtime. The menu will prompt the user for the product.

1. Complete the example *Adding a Static Drop-Down Menu* on page 7-21 if you have not done so.

   Open the launch page named LAUNCH.HTM in FrontPage. Place the cursor after the static drop-down menu, and press *Enter* to add a blank line. You must position the cursor *inside* the form. Type *Select a product:*.

   Press Enter again.

2. On the FrontPage Insert menu, select *WebFOCUS Form*.

   From the drop-down list, select *Dynamic Drop-Down Menu*.

   A drop-down menu is added to the form.

3. Set the drop-down menu properties:

   **a.** Double-click the drop-down menu. The Design-Time Control Properties dialog box opens.

   **b.** In the FOCUS Parameter field, select *PRODNAME* from the drop-down list.

   **c.** In the List data from file field, select the data source *CENTORD*.

   **d.** In the Choice field, select *PRODNAME* from the drop-down list. The dynamic drop-down menu on the launch page will be populated by PRODNAME field values from the CENTORD data source.

4. In the Height field, type the number *8*, indicating how many values will display at one time in the dynamic drop-down menu.

5. Click *OK* to close the dialog box.

6. Save the launch page and exit FrontPage.

7. A dynamic drop-down menu requires population on the Reporting Server. Use the Deploy Wizard to deploy the launch page and procedure:

   **a.** In Developer Studio, right-click the *Inventory* project. Select *Deploy* from the pop-up menu.

   **b.** On the Partition Your Application dialog box, select these files to send to the Web server: LAUNCH.HTM, LAUNCH1.HTM, and LAUNCH2.HTM.

   Select this file to send to the Reporting Server: IN_STOCK.FEX.

   Click *Next* to close the dialog box.

   **c.** On the Choose WebFOCUS Servers dialog box, select the defined server in the left column, and click *Add* to add it to the right column.

   Click *Next* to close the dialog box.

   **d.** On the Save/Deploy Action dialog box, select *Save and deploy new or changed files*. Click *Finish*. If prompted, log on to the server.

   The files are deployed to the appropriate servers.

8. In WebFOCUS, run LAUNCH.HTM by entering the URL for your site. The following is an example; the Web server is named server1, and the alias ibi_html has been established on that server:

   http://server1/ibi_html/inventor/launch.htm

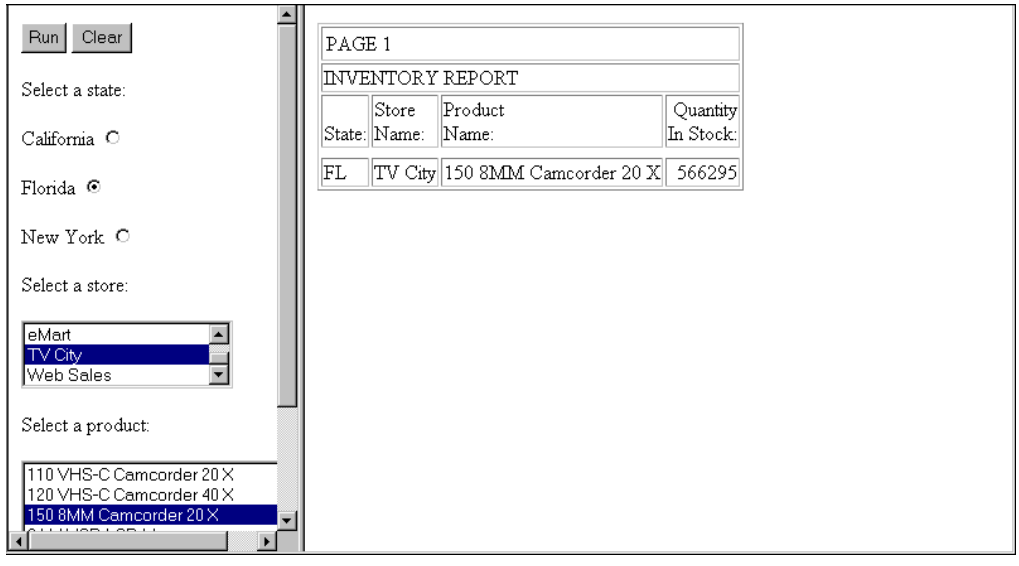
   In deployment, the Deploy Wizard truncates the target directory name to eight characters.

9. For the state, click *Florida*.

   For the store, select *TV City* from the static drop-down menu.

   For the product, select *150 8MM Camcorder 20X* from the dynamic drop-down menu.

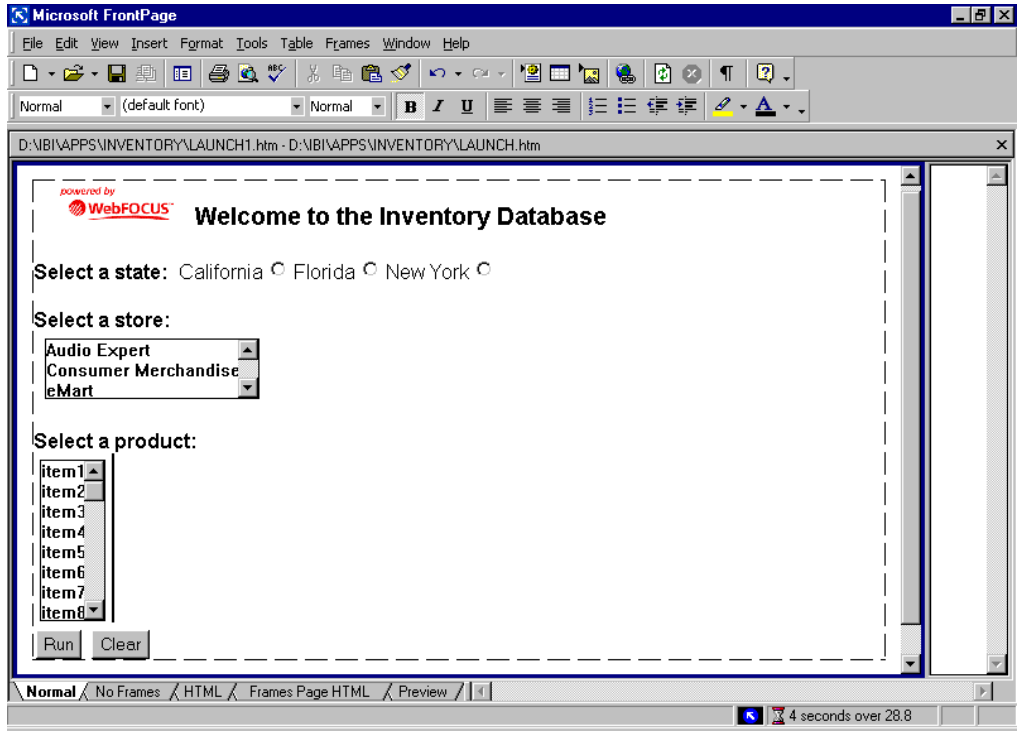
**10.** Click *Run* on the launch page to display the report.



## *Example* Styling the Launch Page and Report

1. Use standard FrontPage features to style the launch page, making it look more professional. You can add descriptive text, change the font style and size, experiment with color, reposition the controls, and add images. See your FrontPage documentation for details on styling capabilities.

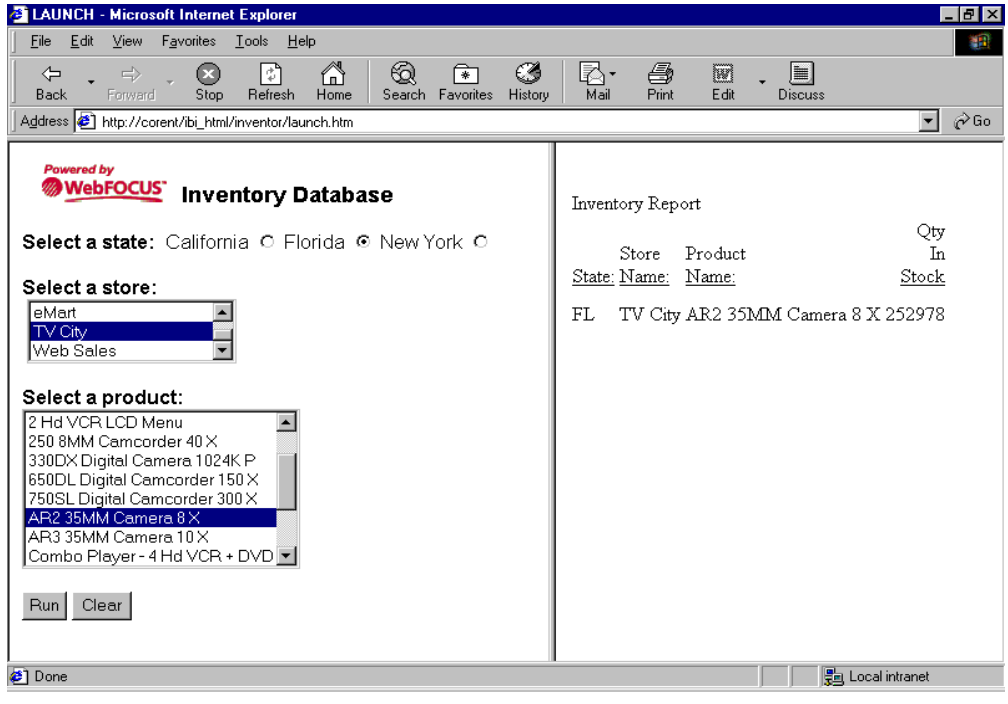
   The sample launch page is being styled in FrontPage:




2. Add styling features to the procedure using a WebFOCUS StyleSheet. See the *Creating Reports With WebFOCUS Language* manual for details on StyleSheets.

3. When you are done, deploy the launch page and procedure. See Chapter 12, *Partitioning and Deploying Project Files*, for instructions.

   The following is an example of the launch page as styled in FrontPage. A StyleSheet was used in the procedure to remove the grid from the report, and WebFOCUS commands were used to omit the page number and modify the fourth column title.




## Adding JavaScript to Automatically Run a Report

With the FrontPage HTML editor, you can access the source code for a launch page. One design technique is to add JavaScript to automatically run a default report the same time a launch page opens in the browser.

For more information on using JavaScript with WebFOCUS, see Chapter 8, *Coding a User Interface*.

The examples in this topic create a sample launch page that automatically runs a procedure when it opens. The sequence of tasks is:

- Create and test a procedure.
- Create a launch page with a form and radio button group.
- Add JavaScript to the HTML code.

## *Example* Creating and Testing a Procedure

This example uses the Inventory project developed in *Creating a Sample Launch Page for an Inventory Report* on page 7-9. If you did not create that project, do it now, or use a project of your choice.

1. Create a procedure named QARPT.FEX. If required, see Chapter 4, *Creating a Reporting Procedure*, for instructions.

   The procedure reports on product problems in a plant of interest. The default report for the Boston plant displays when the launch page opens.

   The code is:

   ```
   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"
   END
   ```

2. Test the procedure by running it remotely. To do that, right-click the procedure, and select *Flag as Remote* from the pop-up menu. Then click *Run* from the same menu. Enter server credentials if prompted.

3. WebFOCUS prompts for the required value. Enter *BOS* for the plant (PLANT). You should receive a quality assurance report for Boston. If you do not, check the code in the procedure and retry.

4. To continue, go to the example *Creating a Launch Page* on page 7-30.

## *Example* Creating a Launch Page

1. Complete the example *Creating and Testing a Procedure* on page 7-29 if you have not done so.

   In Developer Studio, right-click the HTML Forms folder for the Inventory project. Select *New* from the pop-up menu, then *HTML File* from the next menu.

   The New HTML form file dialog box opens.

2. On the Page tab, type AUTORUN for the launch page in the Name of new file field.

   WebFOCUS automatically appends the extension .HTM when it saves the file.

3. The launch page is a frameset with two frames, so click the *Frames* tab. When you complete the project, a default report will display in the right frame the same time the launch page opens.

   On the Frames tab, click the *IBI Contents* template and *OK*. The template is copied to a new file in the default path for the project and then opens in FrontPage.

4. Click in the left frame.

   On the FrontPage Insert menu, select *WebFOCUS Form*, and then select *Form*.

   The Insert WebFOCUS Form dialog box opens.

5. Set the WebFOCUS form properties:

   **a.** For the Form Name, accept the default, *WFForm1*.

   **b.** For the FOCEXEC to run, select *QARPT.FEX* from the drop-down list.

   **c.** Click the pencil button to the right of the Target Frame field.

   **d.** In the Target Frame dialog box, select *main* as the target frame (the frame that will contain the report), and click *OK*.

   **e.** Click *OK* on the Insert WebFOCUS Form dialog box.

   WebFOCUS saves the form with the defined properties, and adds a Run and Clear button to the left frame.

6. Right-click in the right frame (main), and select *Frame Properties* from the pop-up menu.

   You can use this dialog box to adjust the size and other characteristics of the frame. For now, accept the default values, and click *OK*.

7. In the left frame, place the cursor in the WebFOCUS form, at the end of the line with the Run and Clear buttons. You must position the cursor *inside* the form.

   Press *Enter* to add a blank line. Type *Select a plant:*.

   Press *Enter* again. Type the label for the first radio button directly on the form: *Boston*.

8. On the FrontPage Insert menu, select *WebFOCUS Form*.

   Select *Radio Button*.

   A radio button is added to the form.

9. Set the radio button properties:

   a. Double-click the radio button. The Design-Time Control Properties dialog box opens.

   b. In the FOCUS Parameter field, click the down arrow, and select *PLANT* from the drop down list.

   c. In the Value field, type *BOS*.

      Click *Selected* to assign BOS as the default value for PLANT.

   d. Click *OK*.

10. Create a blank line under the first radio button, and type a label for the second button: *Dallas*.

    Add a radio button that has the value DAL for the field PLANT. Click *Not selected* for this value.

11. To continue, go to the example *Adding JavaScript* on page 7-31.

## *Example* Adding JavaScript

1. Complete the example *Creating a Launch Page* on page 7-30 if you have not done so.

   With AUTORUN.HTM open in FrontPage, click the *HTML* tab at the bottom of the window.

2. In the left frame, scroll down to the end of the JavaScript section. The following code marks the end:

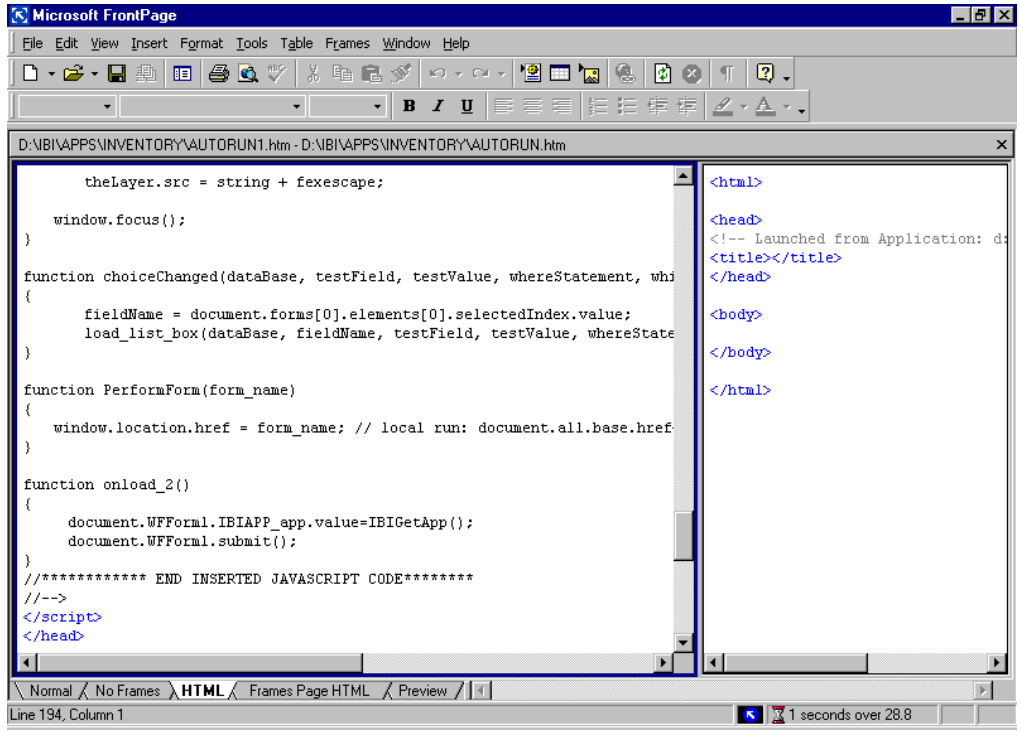
   ```
   //********** END INSERTED JAVASCRIPT CODE **********
   ```

3. Place the cursor before this code, and press *Enter* to add a blank line.

**4.** Place the cursor in the blank line, and add the following JavaScript:

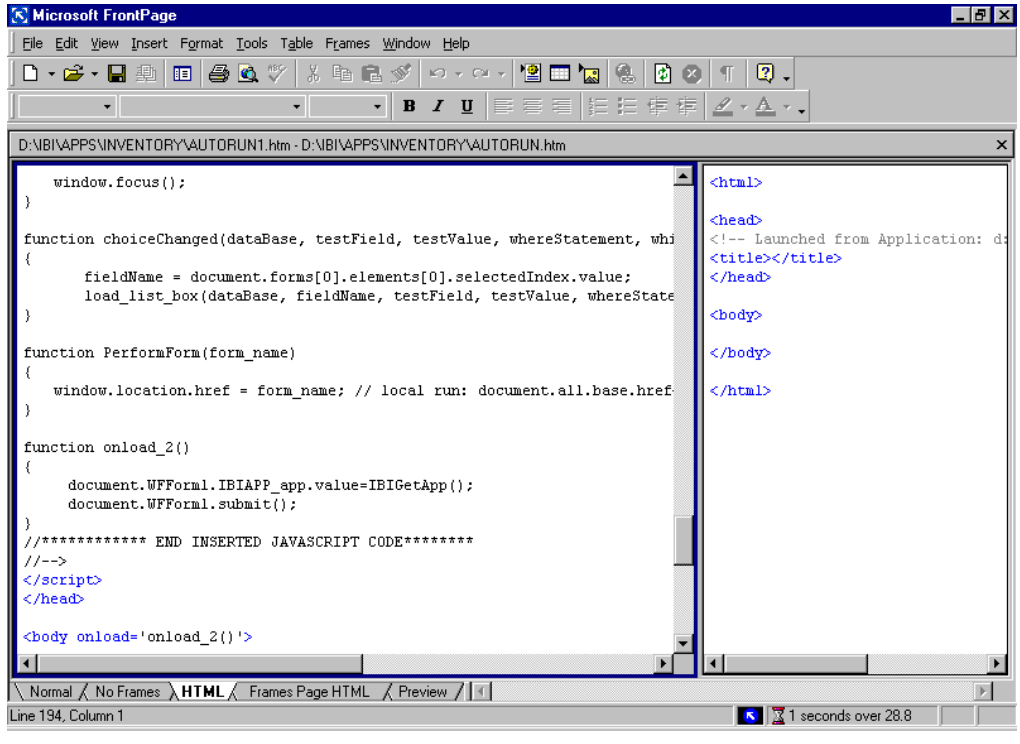
```
function onload_2()
{
     document.WFForm1.IBIAPP_app.value=IBIGetApp();
     document.WFForm1.submit();
}
```

The code in the left frame will look like this:



**5.** Scroll down to the first <body> tag. Replace it with the following code:

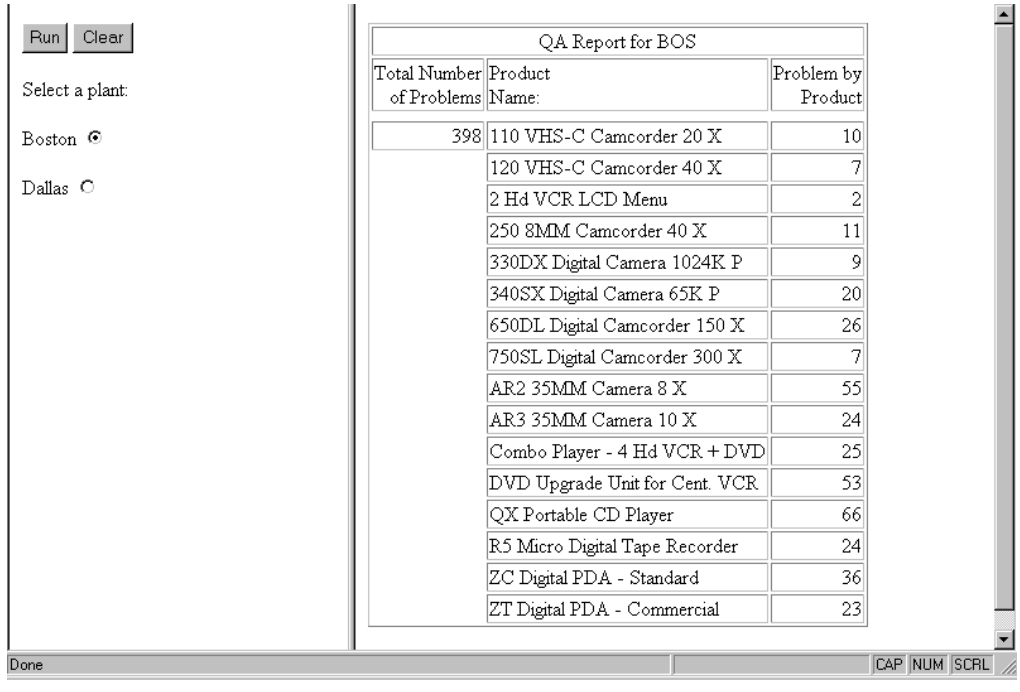
```
<body onload='onload_2()'>
```




**6.** Save the launch page, and exit FrontPage.

**7.** In Developer Studio, right-click and run the form named AUTORUN.HTM. Enter server credentials if prompted.

The launch page displays in the left frame, and the report for Boston automatically displays in the right frame. Remember, you chose BOS as the initial (default) value for PLANT.




QA Report for BOS

| Total Number of Problems | Product Name: | Problem by Product |
|---|---|---|
| 398 | 110 VHS-C Camcorder 20 X | 10 |
| | 120 VHS-C Camcorder 40 X | 7 |
| | 2 Hd VCR LCD Menu | 2 |
| | 250 8MM Camcorder 40 X | 11 |
| | 330DX Digital Camera 1024K P | 9 |
| | 340SX Digital Camera 65K P | 20 |
| | 650DL Digital Camcorder 150 X | 26 |
| | 750SL Digital Camcorder 300 X | 7 |
| | AR2 35MM Camera 8 X | 55 |
| | AR3 35MM Camera 10 X | 24 |
| | Combo Player - 4 Hd VCR + DVD | 25 |
| | DVD Upgrade Unit for Cent. VCR | 53 |
| | QX Portable CD Player | 66 |
| | R5 Micro Digital Tape Recorder | 24 |
| | ZC Digital PDA - Standard | 36 |
| | ZT Digital PDA - Commercial | 23 |

**8.** Select *Dallas,* and click *Run* to display the report for that plant.

**9.** You can now use standard FrontPage features to style the launch page; see your FrontPage documentation for details. You can also style the report; see the *Creating Reports With WebFOCUS Language* manual for information on StyleSheets.

When you are done, deploy the launch page and procedure. See Chapter 12, *Partitioning and Deploying Project Files*, for instructions.

## Displaying a Report and Graph

The examples in this topic create a sample launch page with hyperlinks that display a report and graph. The sequence of tasks is:

- Create a project, and create and test two procedures.
- Create a launch page with hyperlinks.

### *Example* Creating a Project and Two Procedures

1. Create a new project named ProdSale.
2. Create two procedures, one named TABRPT.FEX and one named GRAPH.FEX.

   The first procedure displays product sales data in tabular format. The second displays the same data in graph format.

   The code for TABRPT.FEX is:

   ```
   TABLE FILE GGSALES
   SUM UNITS DOLLARS
   BY PRODUCT
   WHERE (PRODUCT EQ 'Mug' OR 'Thermos')
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF,$
   ENDSTYLE
   END
   ```

   The code for GRAPH.FEX is:

   ```
   GRAPH FILE GGSALES
   SUM UNITS DOLLARS
   ACROSS PRODUCT
   WHERE (PRODUCT EQ 'Mug' OR 'Thermos')
   END
   ```

3. Test the procedures by running them remotely from the desktop. To do that, right-click the procedure, and select *Flag as Remote* from the pop-up menu. Then click *Run* from the same menu. If either one does not work, check the code and retry.
4. To continue, go to the example *Creating a Launch Page With Hyperlinks* on page 7-36.

## *Example* Creating a Launch Page With Hyperlinks

1. Complete the example *Creating a Project and Two Procedures* on page 7-35 if you have not done so.

   In Developer Studio, right-click the HTML Forms folder for ProdSale. Select *New* from the pop-up menu, then *HTML File* from the next menu to open the New HTML form file dialog box.

2. On the Page tab, type *RPTGRAPH* for the launch page name in the Name of new file field.

   WebFOCUS automatically appends the extension .HTM when it saves the file.

3. The launch page is a frameset with four frames, so click the *Frames* tab.

   On the Frames tab, click the *IBI Banner, Contents, Report & Graph* template, and *OK*. The template is copied to a new file in the default project path and then opens in FrontPage.

4. Click in the far left frame underneath the banner frame.

   On the FrontPage Insert menu, select *WebFOCUS Form*, and then select *Hyperlink*.

   The word WebFOCUS displays in the left frame.

5. Double-click *WebFOCUS* to open the Design-Time Control Properties dialog box.

   For the Focexec to run, select *TABRPT.FEX*.

   For the Text to display, type *Tabular Report*.

6. Click the pencil button to the right of the Target Frame field. On the Target Frame dialog box, select *bleft,* and click *OK*.

   Click *OK* to close the Design-Time Control Properties dialog box.

7. Repeat steps 4, 5, and 6 to add a hyperlink that runs GRAPH.FEX. For the Text to display, type *Graph*.

   For the Target Frame, select *bright*.

   Click *OK* to close all dialog boxes.

8. Click in the banner frame, and type *Product Sales Report*.

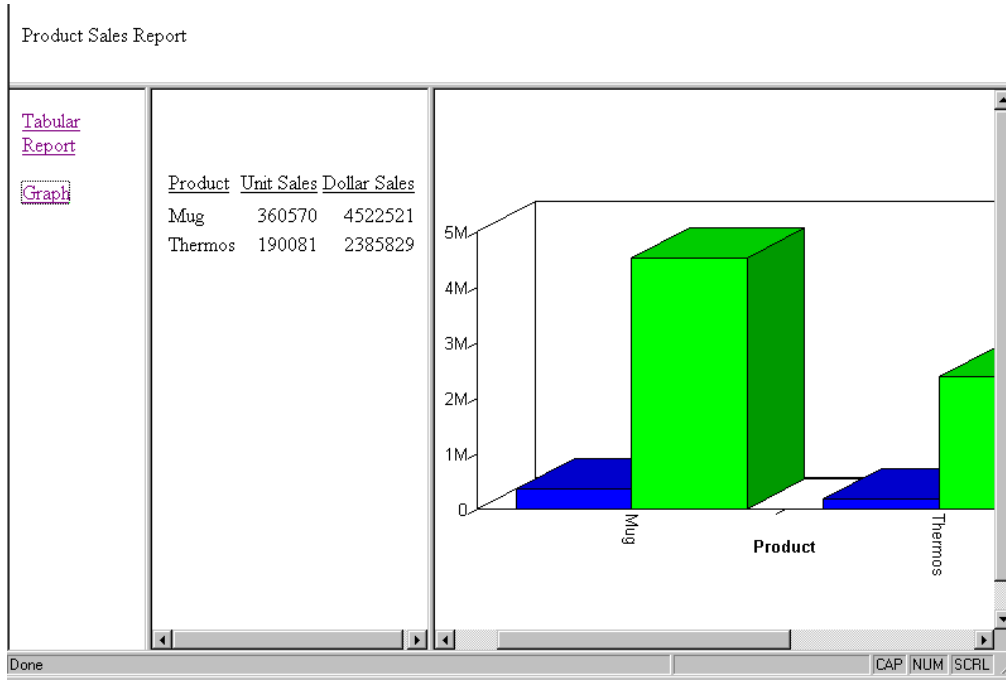
9. Save the launch page, and exit FrontPage.

**10.** In Developer Studio, right-click and run the form named RPTGRAPH.HTM. Enter server credentials if prompted.

Click *Tabular Report* to display the report in the frame to the right of the hyperlinks.

Click *Graph* to display the graph in the next frame.

Each frame is scrollable so you can see the entire report or graph.




**11.** You can now use standard FrontPage features to style the launch page. See your FrontPage documentation for details.

When you are done, deploy the launch page and supporting files. See Chapter 12, *Partitioning and Deploying Project Files*, for instructions.

# Creating a Main Page

You can use FrontPage menu options with WebFOCUS options to create a main page that controls a project. From the main page in our example, a user can:

- Run the report created in *Displaying a Report and Graph* on page 7-35.
- Link to the Information Builders' Web site.
- Clear current report or Web site content and return home.

The sequence of tasks is:

- Start a main page as a frameset.
- Add the hyperlinks to the report and Web site.
- Add the home link.

The example uses images supplied with Developer Studio software, for the purpose of illustration.

## *Example* Starting a Main Page

This example uses the ProdSale project, developed in *Displaying a Report and Graph* on page 7-35. If you did not create that project, do it now, or use a project of your choice.

1. Right-click the HTML Forms folder for the ProdSale project, and select *New* from the pop-up menu, then select *HTML File* in the next menu.
2. In the Name of new file field, type *mainpage*.
3. Click the *Frames* tab, select the *IBI Contents* template, and click *OK*. FrontPage opens a copy of the template.
4. Right-click the left frame, and select *Page Properties* from the pop-up menu.
5. Click the pencil button to open the Target Frame dialog box.

   In the Current frames page box, click the right frame and *OK*.

   Click *OK* to close the Page Properties dialog box.
6. To continue, go to the example *Adding the Hyperlinks* on page 7-39.

## *Example* Adding the Hyperlinks

1. Complete the example *Starting a Main Page* on page 7-38 if you have not done so.
2. With MAINPAGE.HTM open in FrontPage, place the cursor in the top left corner of the left frame.
3. On the Insert menu, select *Picture*. Select *From File* from the cascading menu.
4. Access the directory *install_drive*:\Ibi\Apps\Samples\DevStudio\Images, where *install_drive* applies to your setup.

   Select *webfocus1*. Click *OK* to insert the image into the frame.
5. Right-click the image in the frame, and select *Hyperlink* from the pop-up menu.
6. In the URL field on the Create Hyperlink dialog box, type *rptgraph.htm*.

   RPTGRAPH.HTM is the name of the launch page you created in the example in *Displaying a Report and Graph* on page 7-35.

   Click *OK* to close the dialog box.
7. Position the cursor after the image, and press *Enter*.

   On the Insert menu, select *Picture*. Select *From File* from the cascading menu.
8. Access the directory *install_drive*:\Ibi\Apps\Samples\DevStudio\Images, where *install_drive* applies to your setup.

   Select *maglass*. Click *OK* to insert the image into the frame.
9. Right-click the image in the frame, and select *Hyperlink* from the pop-up menu.
10. In the URL field on the Create Hyperlink dialog box, type *http://www.informationbuilders.com*.

    Click *OK* to close the dialog box.
11. To continue, go to the example *Adding the Home Link* on page 7-40.

## *Example* Adding the Home Link

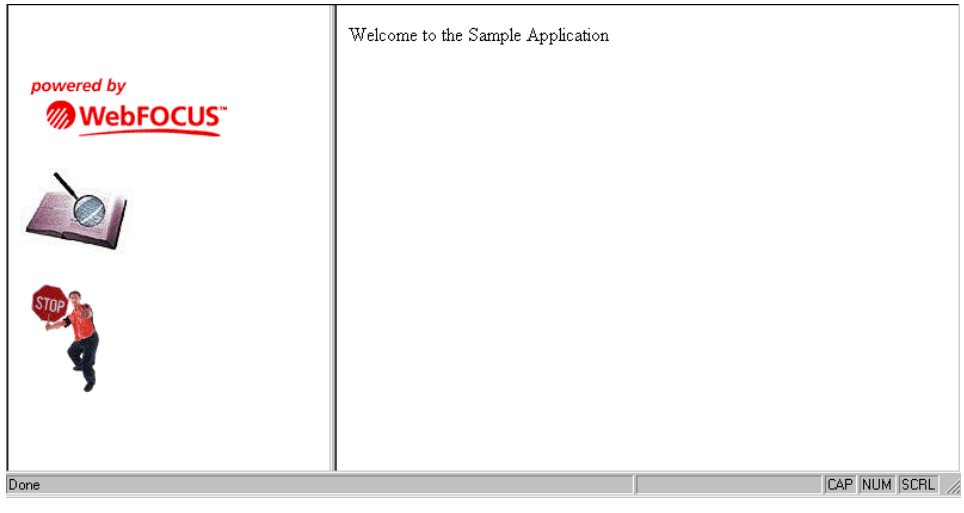
1. Complete the example *Adding the Hyperlinks* on page 7-39 if you have not done so.
2. With MAINGPAGE.HTM open in FrontPage, press *Enter* after the second image.

   On the Insert menu, select *Picture*. Select *From File* from the cascading menu.
3. Access the directory *install_drive*:\Ibi\Apps\Samples\DevStudio\Images, where *install_drive* applies to your setup.

   Select *stopguy*. Click *OK* to insert the image into the frame.

   This link is the home link. It will call mainpag2.htm, clearing it of current content.
4. Right-click in the right frame, and select *Frame Properties* from the pop-up menu.

   The initial page for this frame is mainpag2.htm. Accept this value, and click *OK*.
5. Right-click the stopguy image, and select *Hyperlink*.
6. In the URL field on the Create Hyperlink dialog box, type *mainpag2.htm* and click *OK*.
7. Position the cursor at the top of the right frame. Type *Welcome to the Sample Application*.
8. Save the page, and exit FrontPage.
9. In Developer Studio, right-click and run the launch page named MAINPAGE.HTM.

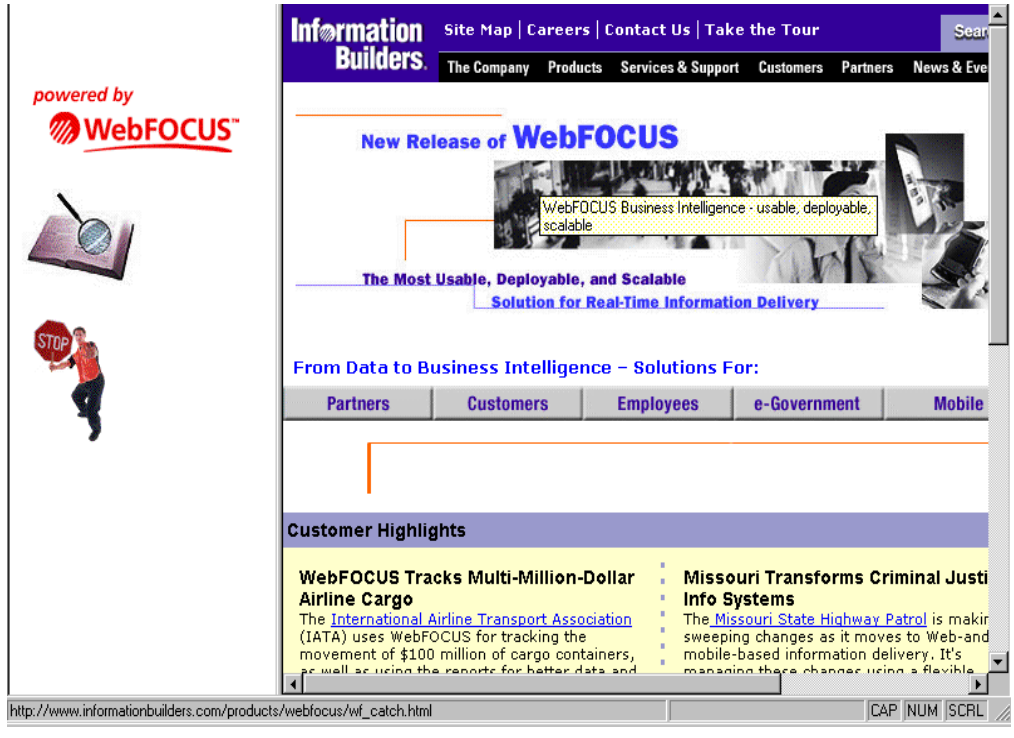



Click *powered by WebFOCUS* to display the report and graph launch page.

Click *Tabular Report*. Enter server credentials if prompted for them.

Click *Graph*.

**10.** Click the magnifying glass to go to Information Builders' Web site.




Click the stopguy to return home.

**11.** You can now use standard FrontPage features to complete and style the main page. See your FrontPage documentation for details.

When you are done, use the Deploy Wizard to deploy the main page and supporting files. Deploy the image files using the option Add Extra Deployable Files. See Chapter 12, *Partitioning and Deploying Project Files*, for instructions.

In the source code for MAINPAG1.HTM, change the pointer to the image files to the path in WebFOCUS.

CHAPTER 8

# Coding a User Interface

**Topics:**

- Which Tools Can You Use?
- The WebFOCUS Client

**Launch Page**

- Using the CGI
- Using ISAPI
- Using the WebFOCUS Servlet
- Using a Dynamic Multi-Select Drop-Down List
- Enabling Ad Hoc Reporting
- Validating a Form With JavaScript
- Amper Auto-Prompting in CGI/Servlet

**Display Page**

- Displaying a Report on the Default WebFOCUS Page
- Designing an HTML Page for Report Display

This topic describes how to code a user interface. You will see how to call WebFOCUS from a hyperlink or form on a launch page to run a report. A form may contain one or more controls that prompt for values required by a procedure.

JavaScript adds dynamic features to a user interface. This topic illustrates the use of JavaScript to check for a required entry on a form.

You will also see how to use the Dialogue Manager command -HTMLFORM to embed report output on a display page that you design.

User interface templates are supplied with your WebFOCUS software. They include a variety of launch pages that call different types of reports. You can modify these templates for your application.

Additional coding capabilities are described in Chapter 9, *Enhancing a User Interface*.

# Which Tools Can You Use?

You can manually code an HTML-based user interface page using:

- The Report Layout Painter allows you to graphically create an HTML page that incorporates WebFOCUS forms, reports, graphs, and Web objects. For details on the Report Layout Painter, see Chapter 6, *Designing a User Interface for a Web Application With the Report Layout Painter*.

- The Microsoft® FrontPage® 2000 HTML source editor in the Developer Studio. Working with FrontPage allows you to take advantage of the Deploy Wizard, which distributes finished components of an application to the appropriate servers. See Chapter 12, *Partitioning and Deploying Project Files*, for details.
- The Developer Studio text editor (TED). This editor automatically associates a user interface page with your application, and the Deploy Wizard can access that page. For details on this editor, see your *WebFOCUS Application Development Getting Started* manual.
- A standard text editor of your choice. With this option, you can copy or move a user interface page to a project directory in the Developer Studio. The Deploy Wizard can access the page.

Developers who have a preferred editor and deployment method can use them outside of the Developer Studio for coding and server distribution.

To code a procedure called by a user interface page, use the Report Painter, Report Assistant, or TED. Deploy the procedure with the Deploy Wizard.

## *Procedure* How to Use the FrontPage HTML Source Editor

1. Open the project for which you are coding an interface page.

   Right-click the HTML Forms folder, and click *New* on the pop-up menu.

2. On the New HTML Form File dialog box, respond to the prompts as described in Chapter 6, *Designing a User Interface for a Web Application With the Report Layout Painter*. Click *OK* to open FrontPage.
3. At the bottom left of the FrontPage window, click the *HTML* tab. The editor opens, displaying default HTML code.
4. Type your HTML code in the window. When you are done, select *Save* from the File menu.

   WebFOCUS saves the file and associates it with the project.

## Choosing Search Path Logic

You can use FrontPage:

- In graphical mode, as described in Chapter 6, *Designing a User Interface for a Web Application With the Report Layout Painter*. If you use this mode, it automatically generates the variable IBIAPP_app so that the WebFOCUS Reporting Server searches for the application components using APP PATH. When this variable is passed from the HTML interface page to the WebFOCUS Client, the Client signals the Reporting Server to switch to APP PATH logic and redefine APP PATH with the project directory.
- In edit mode, using the FrontPage HTML source editor. If you use the FrontPage HTML source editor, or another editor (for example, TED or Microsoft Notepad) or a third-party graphical tool, to create a user interface page, decide whether to run the application with EDAPATH or APP PATH logic.

You can set either EDAPATH or APP PATH logic in the Reporting Server global profile (EDASPROF.PRF), a user profile, or a procedure. For either kind of logic, make sure that the following files are accessible by the Reporting Server:

- Procedures.
- WebFOCUS StyleSheets.
- Custom HTML display pages called by the -HTMLFORM command.
- Images embedded in report output.

The directories in which the preceding files reside must be included on EDAPATH or APP PATH, depending on the logic used.

### *Procedure* How to Use EDAPATH Logic

If you create a user interface page with a text editor and deploy your project files with the Deploy Wizard, make sure that the path created by the Deploy Wizard on the Reporting Server is defined in EDAPATH.

For details on the path created for deployed files, see Chapter 12, *Partitioning and Deploying Project Files*. Also see your iWay™ Server configuration documentation on specifying files.

## *Procedure* How to Use APP PATH Logic

If you create a user interface page with a text editor and deploy your project files with the Deploy Wizard, do the following.

In the interface page, set the variable IBIAPP_app as follows

```
<INPUT TYPE="HIDDEN" NAME="IBIAPP_app" VALUE="app_directory">
```

where:

*app_directory*

Is the directory on the Reporting Server to which procedures are deployed.

Once you deploy a procedure using APP PATH logic, you can execute it only from the specified APP PATH directory. You cannot change the location of the project without changing the value of the variable IBIAPP_app.

For a full description of all commands related to APP PATH logic, see the iWay Server administration manual for your platform.

For details on the path created for deployed files, see Chapter 12, *Partitioning and Deploying Project Files*. Also see your iWay Server configuration documentation on specifying files.

## *Example* Coding APP PATH Logic

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. In a project named SALESDIR, create a procedure named SALESSE. It reports on total unit and dollar sales in the Southeast for a category requested by the user.

   **Procedure: SALESDIR.FEX**

   ```
   TABLE FILE GGSALES
   HEADING
   "&CATEGORY Sales for Southeast"
   SUM UNITS AND DOLLARS
   WHERE (CATEGORY EQ '&CATEGORY') AND (REGION EQ 'Southeast');
   END
   ```

2. Using a third-party text editor such as Notepad, manually code a launch page named LAUNCHSE. It contains a form that uses the CGI to call WebFOCUS to run SALESSE. A one-line text box prompts the user for a value for CATEGORY.

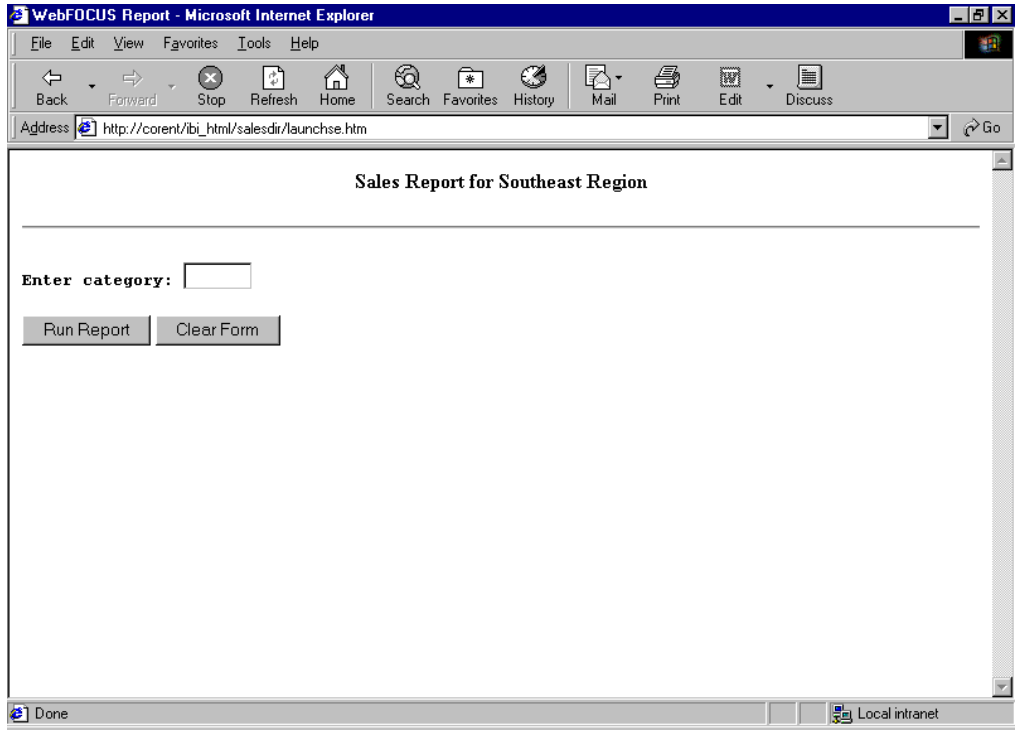
   The launch page contains code that sets the variable IBIAPP_app to the directory (SALESDIR) in which the procedure (SALESSE) resides after deployment by the Deploy Wizard.

   **Launch Page: LAUNCHSE.HTML**

```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
</HEAD>
<BODY>
<H4 ALIGN=CENTER>Sales Report for Southeast Region</H4>
<HR>
<FORM ACTION="/cgi-bin/ibi_cgi/ibiweb.exe" METHOD="get">
<INPUT TYPE="HIDDEN" NAME="IBIAPP_app" VALUE="SALESDIR">
<INPUT NAME="IBIF_ex" VALUE="salesse" TYPE="hidden">
<P ALIGN=LEFT NOWRAP><PRE>
<B>Enter category: </B><INPUT NAME="CATEGORY" TYPE="text" SIZE="6">
</PRE></P>
<INPUT NAME="submit" TYPE=SUBMIT VALUE="Run Report">
<INPUT NAME="reset" TYPE=RESET VALUE="Clear Form">
</FORM>
</BODY>
</HTML>
```

3. Deploy the procedure and launch page using the Deploy Wizard. See Chapter 12, *Partitioning and Deploying Project Files*, for details.

**4.** Run the launch page in your browser. It will look similar to this:




**5.** In the one-line text box that prompts for the category, type:

```
Food
```

Click *Run Report* to receive the report:

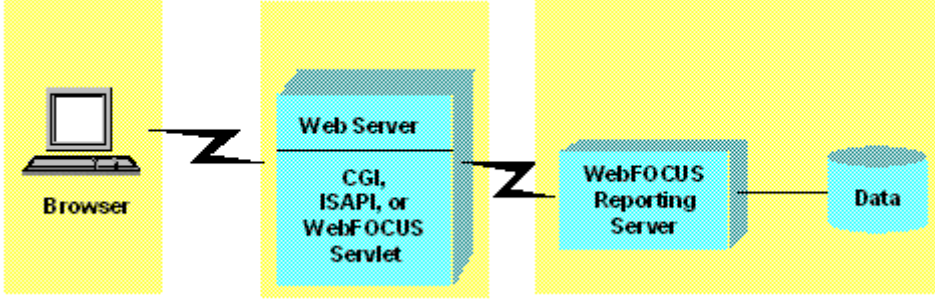
| PAGE 1 | |
|---|---|
| Food Sales for Southeast | |
| Unit Sales | Dollar Sales |
| 349829 | 4308731 |

## The WebFOCUS Client

When WebFOCUS is installed it is configured to invoke the WebFOCUS client in one of the following ways:

- CGI (Common Graphical Interface)
- ISAPI (proprietary API for the Microsoft Internet Information Server)
- WebFOCUS Servlet

Each option resides on the same machine as the Web server and routes requests from that server to the WebFOCUS Reporting Server, as the diagram illustrates.




Each option includes complete support for WebFOCUS features and accepts variables from the standard WebFOCUS script (WFS) files, installed by default in the WebFOCUS Client. You can determine what option you are using in the WebFOCUS Console.

*Procedure* **How to Determine Whether WebFOCUS Is Using the CGI, ISAPI, or the WebFOCUS Servlet**

1. In the WebFOCUS Console, click *Change/Add*.
2. In the next window, the value for the Admin name parameter tells you what option you are using:

   *WFServlet* indicates that the WebFOCUS Servlet. This is the default.

   *IBIWEB.EXE* indicates that the CGI is being used.

   *WEBAPI.DLL* indicates that  ASAPI is being used.

## Using the CGI

If you are using the CGI to run a dynamic report and pass variable values, you can call it:

- From an HTML hyperlink to run a procedure without passed values, or with fixed values.

- From an HTML form to run a procedure with user-supplied values. Provide controls on the form, such as radio buttons and check boxes, which prompt for the values.

## *Syntax* How to Call the CGI From a Hyperlink

```
<A HREF="/alias/ibiweb.exe?IBIF_ex=procedure[&var=value[&var=value]...]">
text</A>
```

where:

*alias*

Points to the directory in which the WebFOCUS CGI (ibiweb.exe) 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 the WebFOCUS CGI on another Web server, specify a fully qualified URL, such as:

```
http://web_server/alias/ibiweb.exe...
```

*procedure*

Is the name of the procedure to run.

*var=value*

Is a variable and its corresponding value. The call can include both HTTP environment variables and application variables passed to the procedure.

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.

*text*

Is the text on the launch page that serves as the hyperlink that runs the procedure.

## *Example* Calling the CGI From a Hyperlink

This example creates a launch page with four hyperlinks. Each hyperlink calls the CGI to pass:

- The name of a procedure to run.
- Two values that the procedure requires.

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a procedure named CSALES. It generates a dynamic report that shows total unit and dollar sales for a category and region.

   **Procedure: CSALES.FEX**

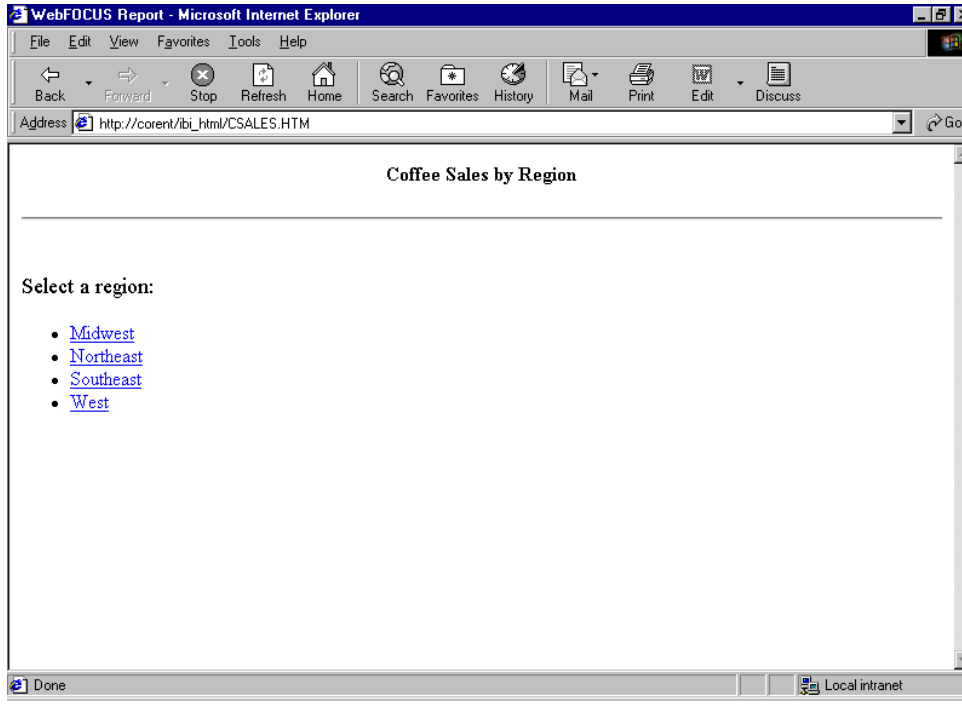
```
TABLE FILE GGSALES
HEADING
"&CATEGORY Sales for &RGN Region"
SUM UNITS AND DOLLARS
WHERE (CATEGORY EQ '&CATEGORY') AND (REGION EQ '&RGN');
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

2. Create a launch page named CSALES. Each instance of the A HREF code calls the WebFOCUS CGI and passes it the name of the procedure, CSALES. Each instance also passes the value Coffee for CATEGORY, and a value for RGN (Midwest, Northeast, Southeast, or West). The hyperlinks on the launch page prompt the user for the region of interest.

   **Launch Page: CSALES.HTM**

```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
</HEAD>
<BODY>
<H4 ALIGN=CENTER>Coffee Sales by Region</H4>
<HR>
<P><FONT SIZE=+2></FONT><BR>
<FONT SIZE=+1>Select a region:</FONT>
</P>
<UL TYPE=SQUARE>
<LI><A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=csales&CATEGORY=
Coffee&RGN=Midwest">Midwest</A>
<LI><A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=csales&CATEGORY=
Coffee&RGN=Northeast">Northeast</A>
<LI><A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=csales&CATEGORY=
Coffee&RGN=Southeast">Southeast</A>
<LI><A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=csales&CATEGORY=
Coffee&RGN=West">West</A>
</UL>
</BODY>
</HTML>
```

**3.** Run the launch page in your browser. It will look similar to this:




**4.** Click the hyperlink *Midwest* to receive the report:

Coffee Sales for Midwest Region

| Unit Sales | Dollar Sales |
|---|---|
| 332777 | 4178513 |

***Syntax*** **How to Call the CGI From a Form**

```
<FORM ACTION="/cgi-bin/ibi_cgi/ibiweb.exe" METHOD="get">
<INPUT NAME="IBIF_ex" VALUE="procedure" TYPE="hidden">
    .
    .
    .
</FORM>
```

where:

*procedure*

Is the name of the procedure to run. The call can include both HTTP environment variables and application variables passed to the procedure.

***Reference*** **Conventions for Creating a Form**

- To begin and end a form, use the HTML <FORM tag.
- To specify the name of the procedure to run, use the HTML <INPUT tag.
- Include the desired controls on the form, such as text boxes, check boxes, or drop-down lists.
- Include a Submit button to execute the CGI and pass the values collected on the form to the procedure.

## *Example* Calling the CGI From a Form

This example creates a launch page with a form that calls the CGI to pass:

- The name of a procedure to run.
- Two values that the procedure requires. The form prompts the user for the first value with a one-line text box, and for the second value with radio buttons.

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. If you have not done so in a previous example, create a procedure named CSALES. It generates a dynamic report that shows total unit and dollar sales for the category and region requested by the user.

   **Procedure: CSALES.FEX**

   ```
   TABLE FILE GGSALES
   HEADING
   "&CATEGORY Sales for &RGN Region"
   SUM UNITS AND DOLLARS
   WHERE (CATEGORY EQ '&CATEGORY') AND (REGION EQ '&RGN');
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF,$
   ENDSTYLE
   END
   ```

2. Create a launch page named SALES.HTM. It contains a form that calls the CGI to pass the name of the procedure, CSALES. A one-line text box prompts the user for a value for CATEGORY, and radio buttons prompt for a value for RGN.

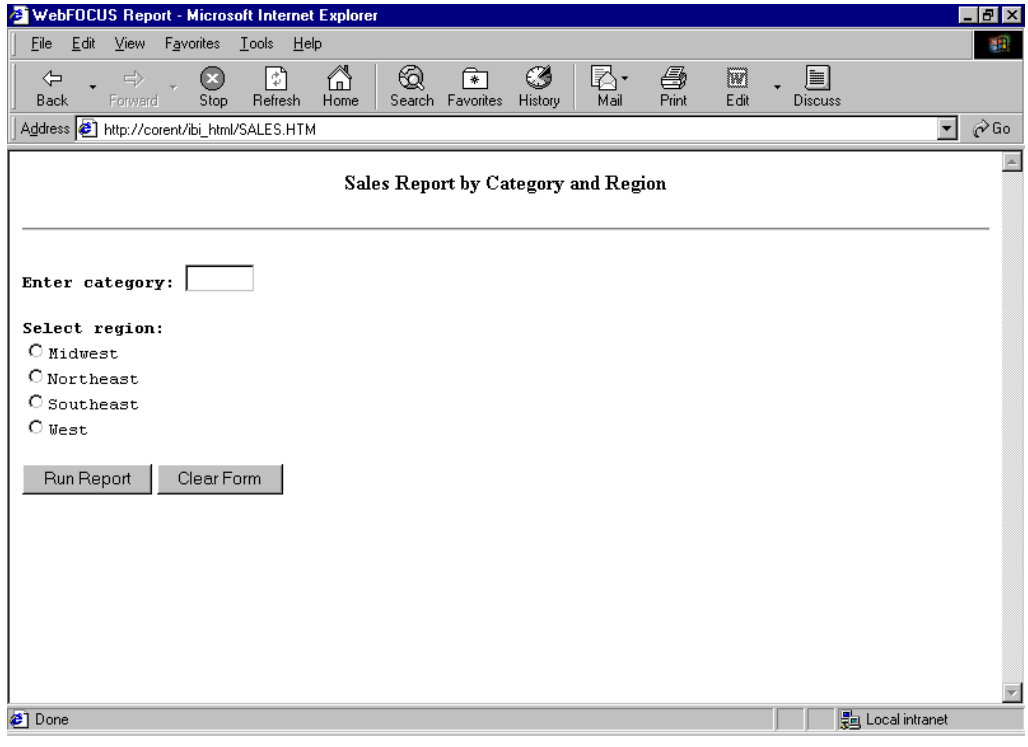
   The form also contains a Submit button (labeled Run Report) and a Reset button (labeled Clear Form). When you prompt for user input on a form, you must include a Submit button.

   *Adding a Control to a Form* in Chapter 5, *Understanding User Interface Basics* provides more information on a one-line text box, radio button, and Submit button.

   **Launch Page: SALES.HTM**

```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
</HEAD>
<BODY>
<H4 ALIGN=CENTER>Sales Report by Category and Region</H4>
<HR>
<FORM ACTION="/cgi-bin/ibi_cgi/ibiweb.exe" METHOD="get">
<INPUT NAME="IBIF_ex" VALUE="csales" TYPE="hidden">
<P ALIGN=LEFT NOWRAP><PRE>
<B>Enter category: </B><INPUT NAME="CATEGORY" TYPE="text" SIZE="6">
</PRE></P>
<P><PRE><B>Select region: </B>
<INPUT NAME="RGN" TYPE=RADIO VALUE=Midwest>Midwest
<INPUT NAME="RGN" TYPE=RADIO VALUE=Northeast>Northeast
<INPUT NAME="RGN" TYPE=RADIO VALUE=Southeast>Southeast
<INPUT NAME="RGN" TYPE=RADIO VALUE=West>West
</PRE></P>
<P>
<INPUT NAME="submit" TYPE=SUBMIT VALUE="Run Report">
<INPUT NAME="reset" TYPE=RESET VALUE="Clear Form">
</P>
</FORM>
</BODY>
</HTML>
```

3. Run the launch page in your browser. It will look similar to this:




4. In the one-line text box, type:

   `Coffee`

5. Select the *Southeast* radio button.

6. Click *Run Report* to receive the report:

Coffee Sales for Southeast Region

| Unit Sales | Dollar Sales |
|---|---|
| 350948 | 4415408 |

# Using ISAPI

You can use the proprietary API for the Microsoft Internet Information Server—known as ISAPI—to call WebFOCUS. ISAPI works the same way as the CGI, providing identical functionality.

It also provides reduced memory use and quicker response time. However, you can only use it with a Microsoft Internet Information Server on Windows NT/2000/XP. Other operating environments must use the CGI or WebFOCUS Servlet.

*Syntax*

### How to Call ISAPI From a Hyperlink

```
<A HREF="/cgi-bin/ibi_cgi/webapi.dll?IBIF_ex=procedure[&var=value
[&var=value]...]">text</A>
```

where:

*procedure*

Is the name of the procedure to run.

*var=value*

Is a variable and its corresponding value. The call can include both HTTP environment variables and application variables passed to the procedure.

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.

*text*

Is the text on the launch page that serves as the hyperlink that runs the procedure.

*Syntax*

### How to Call ISAPI From a Form

```
<FORM ACTION="/cgi-bin/ibi_cgi/webapi.dll" METHOD="get">
<INPUT NAME="IBIF_ex" VALUE="procedure" TYPE="hidden">
    .
    .
    .
</FORM>
```

where:

*procedure*

Is the name of the procedure to run. The call can include both HTTP environment variables and application variables passed to the procedure.

For guidelines on creating a form, see *Conventions for Creating a Form* on page 8-11.

# Using the WebFOCUS Servlet

You can use the WebFOCUS Servlet in place of the CGI or ISAPI. The WebFOCUS Servlet allows greater scalability by providing sharable threads, and can support more users on one machine. It also provides easier cross-platform code movement of the WebFOCUS Client while configuring the application environment.

The WebFOCUS Servlet (WFServlet) is packaged in the supplied WFServlet.jar file. This file contains all the required classes.

The WebFOCUS Servlet is available for self-service applications. You cannot use it with the Managed Reporting Environment (MRE) or Workbench.

*Reference* **Requirements for WebFOCUS Servlet Configuration**

Your Web server must be servlet-enabled and have servlet support. The following types of servlet support are acceptable:

- Web server with native servlet support (for example, iPlanet™ or WebSphere®).
- Web server with a servlet engine plug-in (for example, ServletExec™ or JRun™).
- Web Application Server (for example, WebSphere, WebLogic®, or SilverStream®).

## Configuring the WebFOCUS Servlet

You must configure the WebFOCUS Servlet. The configuration procedure depends on the servlet engine. For details, see your servlet engine documentation.

## *Procedure* How to Configure the WebFOCUS Servlet

1. Depending on the servlet engine, you may need to stop your Web server or Web Application Server before beginning configuration.

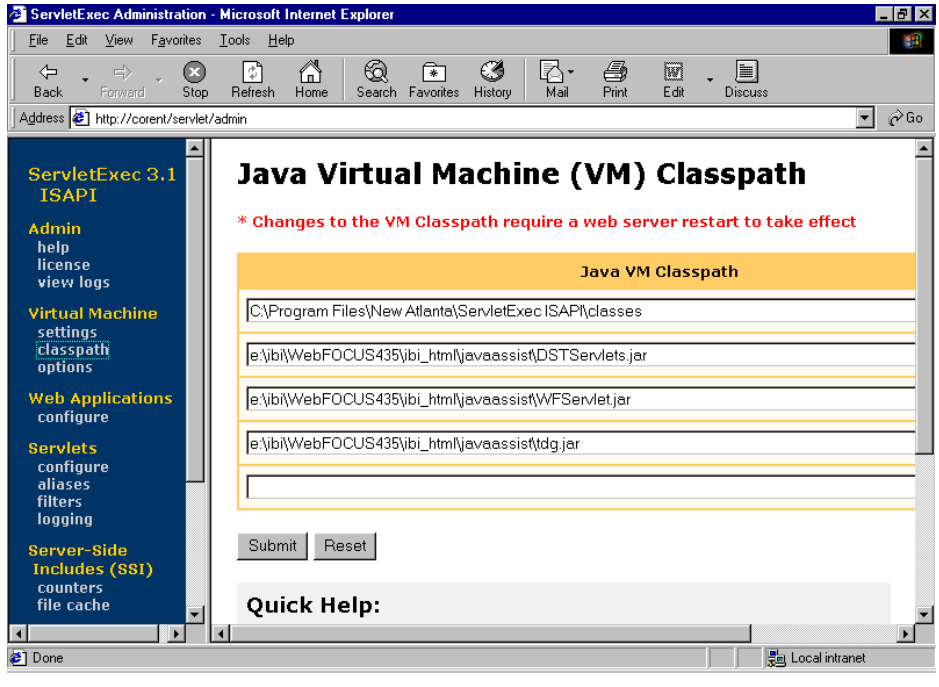
2. Add the WFServlet.jar and tdg.jar files to the Java™ Virtual Machine classpath. The WFServlet.jar and tdg.jar files are located by default in:

   **Windows NT/2000/XP:** *install_drive:*\ibi\WebFOCUS52\ibi_html\javaassist

   **UNIX:** /ibi/WebFOCUS52/ibi_html/javaassist

   **OS/390:** /ibi/WebFOCUS52/ibi_html/javaassist

   The following is an example of the classpath for WebFOCUS on Windows NT:



**3.** Identify servlets to the Web server as follows:

Servlet Name: `WFServlet`

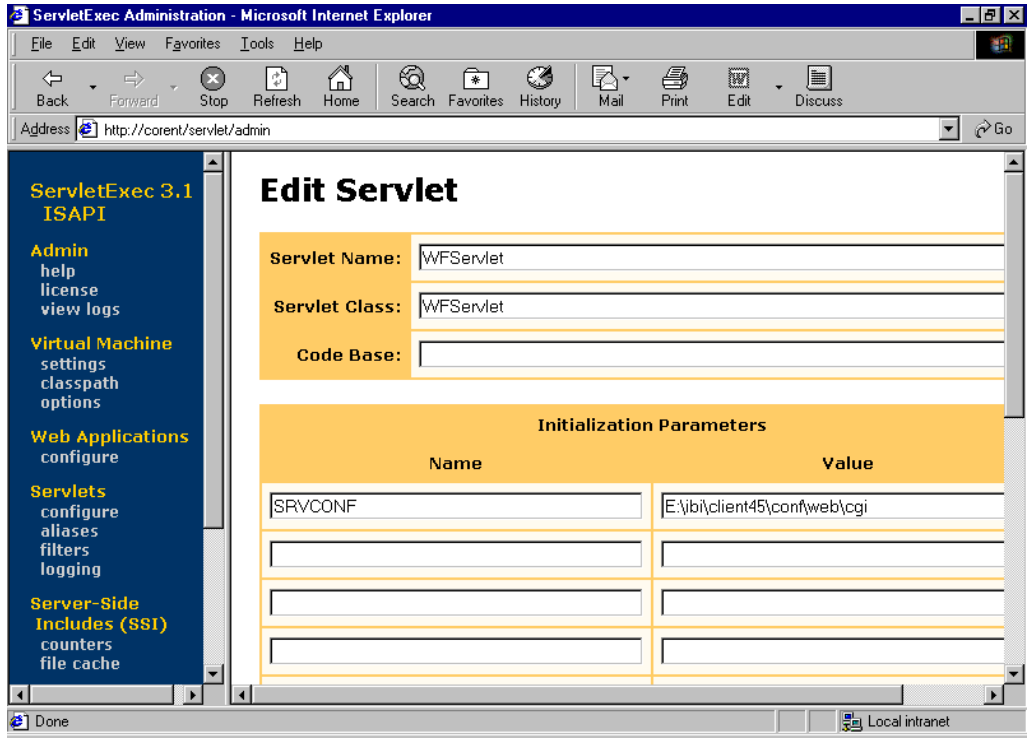
Servlet Class: `WFServlet`

Initialization Parameters (with default locations):

- **Windows NT/2000/XP:** SRVCONF=install_drive:\ibi\client52\wfc\web\cgi
- **UNIX:** SRVCONF=/ibi/client52/wfc/web/cgi
- **OS/390:** SRVCONF=/ibi/client52/wfc/web/cgi

For additional information on the fields, see your servlet engine documentation.

The following is an example for WebFOCUS on Windows NT:




**4.** If you shut down your Web server or Web Application Server before configuration, restart it.

### Calling the WebFOCUS Servlet

The URL code that calls the WebFOCUS Servlet is compatible with standard URL calls to the CGI or ISAPI. Instead of calling the CGI (ibiweb.exe) or ISAPI (webapi.dll), you call WFServlet.

*Example* **Calling the WebFOCUS Servlet**

The following calls the WebFOCUS Servlet on a Web server named server1 to run a procedure named stateinv:

"http://server1/servlet/WFServlet?IBIF_ex=stateinv"

WFServlet replaces a call to ibiweb.exe or webapi.dll.

See *Displaying a Dynamic Report on the Default WebFOCUS Page* on page 8-37 for the full example.

## Using a Dynamic Multi-Select Drop-Down List

You can add a dynamic multi-select drop-down list whose values are populated by field values retrieved live from a data source.

*Procedure* **How to Add a Dynamic Multi-Select Drop-Down List**

This procedure uses the Dialogue Manager command –HTMLFORM to populate a drop-down list with values from a data source. Use this technique when selection values change frequently, to ensure that you always derive a list of current valid values.

For details on -HTMLFORM, see *Designing an HTML Page for Report Display* on page 8-42.

1. Create a procedure that populates a drop-down list:

   a. Allocate a text file that will contain the dynamic values.

      For example, on Windows NT/2000/XP, the command is

      `FILEDEF textfile DISK drive:\directory\textfile.TXT`

      where:

      *textfile* is the name of the file.

      *drive* is the drive on which the file is saved.

      *directory* is the directory in which the file is saved.

      See Chapter 16, *Defining and Allocating WebFOCUS Files*, for details on platform-specific commands.

   b. Include a DEFINE command to create a temporary field that identifies the values.

**c.** Include the following command to save the values to the allocated file.

```
ON TABLE HOLD FORMAT ALPHA as textfile
```

where:

*textfile* is the name of the file. The name can be from 1 to 8 characters.

**d.** Include the following command to send the report output to the launch page that displays the values.

```
-HTMLFORM valuespg
```

where:

*valuespg* is the name of the launch page.

**2.** Create the launch page. Use a comment to indicate where the values display

```
!IBI.FIL.textfile;
```

or

```
<!--WEBFOCUS TABLE textfile>
```

where:

*textfile*

Is the name of the file that contains the values.

The Reporting Server must be able to locate the launch page using APP PATH or EDAPATH. See *Storing and Searching for Application Files* on page 2-12, for details on search paths.

**3.** Create a procedure that accepts multiple values to generate a dynamic report.

## *Example* Adding a Dynamic Multi-Select Drop-Down List

This example runs on WebFOCUS for Windows NT. If you are using another platform, substitute the appropriate platform-specific command for the FILEDEF command. See Chapter 16, *Defining and Allocating WebFOCUS Files*.

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a procedure named DYNAMMUL, which populates a drop-down list from the field COUNTRY in the data source SHORT.

   **Procedure: DYNAMMUL.FEX**

```
FILEDEF DYNAMLST DISK C:\IBI\APPS\GGDEMO\DYNAMLST.TXT
DEFINE FILE SHORT
OPTCOUNTRY/A40 = '<option>'|COUNTRY;
END
TABLE FILE SHORT
SUM OPTCOUNTRY
BY COUNTRY NOPRINT
ON TABLE HOLD FORMAT ALPHA AS DYNAMLST
END
-RUN
-HTMLFORM DYNAMIC2
-RUN
```

2. Create a launch page named DYNAMIC2.HTM. The Reporting Server must be able to locate the page in APP PATH or EDAPATH. See *Storing and Searching for Application Files* in Chapter 2, *WebFOCUS Application Logic*, for details on search paths.

   The sample launch page uses the CGI.

   **Launch Page: DYNAMIC2.HTM**

```
<HTML>
<TITLE> DYNAMIC DROP-DOWN LIST REPORT </TITLE>
<H4> PROJECTED RETURN BY COUNTRY </H4>

<FORM ACTION="/cgi-bin/ibi_cgi/ibiweb.exe" METHOD="GET">
<INPUT TYPE="HIDDEN" NAME="IBIF_ex" VALUE="MULRPT">
<SELECT NAME="COUNTRY" SIZE="3" MULTIPLE><!--WEBFOCUS TABLE
DYNAMLST></SELECT>
<BR><BR><INPUT TYPE="SUBMIT" VALUE="RUN REPORT!"></FORM>
</BODY>
</HTML>
```

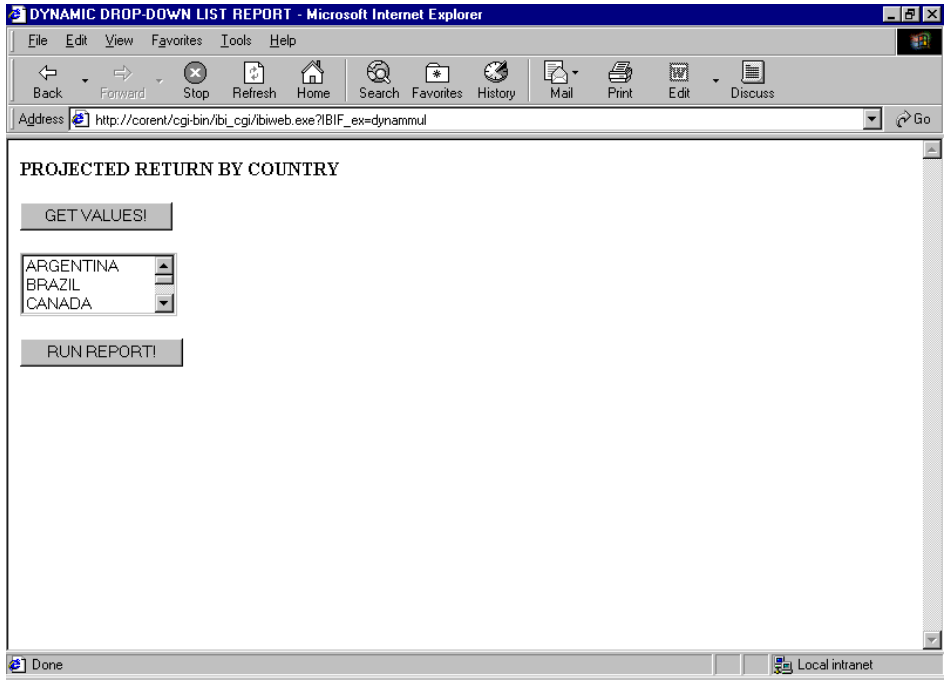
3. Create a procedure named MULRPT, which displays projected returns for a selected country or countries.

   **Procedure: MULRTP.FEX**

```
-SET &COUNTER=1;
TABLE FILE SHORT
SUM PROJECTED_RETURN
BY COUNTRY
-IF &COUNTRY.EXISTS THEN GOTO LOOP1 ELSE GOTO DONE;
-LOOP1
WHERE COUNTRY EQ '&COUNTRY'
-IF &COUNTRY0.EXISTS NE 1 THEN GOTO OUTLOOP;
-REPEAT OUTLOOP FOR &COUNTER FROM 2 TO &COUNTRY0;
OR '&COUNTRY.&COUNTER'
-OUTLOOP
-DONE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

**4.** Run the procedure DYNAMMUL. You can also use JavaScript to run a procedure when a launch page opens. See *Adding JavaScript to Automatically Run a Report* on page 7-28, for an example.

A page similar to the following displays:




**5.** Select *CANADA*, *HONG KONG*, and *MEXICO*. Click *RUN REPORT!* to receive the report:

| Country | Projected Annualized Return |
|---|---|
| CANADA | 3.990 |
| HONG KONG | 4.550 |
| MEXICO | 5.040 |

# Enabling Ad Hoc Reporting

A text area enables a user to enter an ad hoc report request. The request can include WebFOCUS report commands (for example, TABLE FILE *filename*) and Dialogue Manager commands and variables. When the user submits the request, it is assigned in its entirety to the variable associated with the text area and is passed to WebFOCUS.

**Caution:** Before enabling ad hoc reporting, make sure that you consider all the possible consequences. A careless user, for instance, could enter a request which uses so many resources that response time is adversely affected for all users.

A site may disable ad hoc reporting in the file IBIDIR.WFS, which is located by default as follows:

**Windows NT/2000/XP:** *install_drive*:\ibi\client52\wfc\etc

**UNIX:** /ibi/client52/wfc/etc

**OS/390:** /ibi/client52/wfc/etc

If the following line is commented (the default), you can implement ad hoc reporting:

```
# <SET> IBIF_adhocfex(protect)
```

If the line is uncommented, you cannot implement ad hoc reporting. Uncommented, it prohibits the passing of the variable IBIF_adhocfex to the WebFOCUS Client from a browser.

The variable is generated internally. MRE uses it for every procedure.

*Syntax*

## How to Enable Ad Hoc Reporting

You can add optional attributes to the syntax shown.

```
<TEXTAREA NAME="IBIF_adhocfex" VALUE="value" ROWS=rows COLS=cols>
</TEXTAREA>
```

where:

*value*

Is a default request that displays in the text area. To display an empty text area, code:

```
VALUE=""
```

*rows*

Is the number of rows in the text area.

*cols*

Is the number of columns in the text area.

## *Example* Enabling Ad Hoc Reporting

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

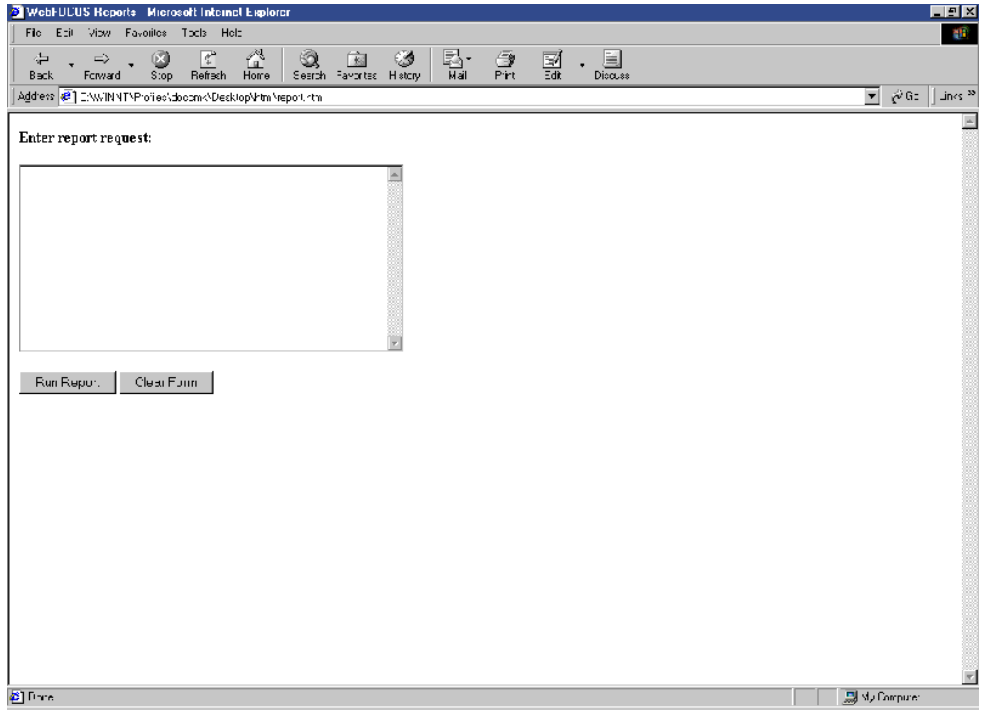
1. Create a launch page named ADHOC, which contains a text area that prompts for a report request. This page must be accessible to the Web server.

   The sample launch page uses ISAPI.

   **Launch Page: ADHOC.HTM**

```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
</HEAD>
<BODY>
<H4>Enter report request:</H4>
<FORM METHOD="get" ACTION="/cgi-bin/ibi_cgi/webapi.dll">
<P ALIGN=LEFT NOWRAP><PRE>
<TEXTAREA NAME="IBIF_adhocfex" VALUE="" ROWS=12 COLS=48 ALIGN=LEFT>
</TEXTAREA>
</PRE></P>
<P>
<INPUT NAME="submit" TYPE=SUBMIT VALUE="Run Report">
<INPUT NAME="reset" TYPE=RESET VALUE="Clear Form">
</P>
</FORM>
</BODY>
</HTML>
```

**2.** Run the launch page, which will look similar to this:




**3.** Enter the following request in the text area:

```
TABLE FILE CENTORD
SUM QUANTITY BY HIGHEST 1 ORDER_DATE
BY PRODNAME
END
```

**4.** Click *Run Report* to receive the report:

| PAGE 1 | | |
|---|---|---|
| Date Of Order: | Product Name: | Quantity: |
| 2000/12/30 | 110 VHS-C Camcorder 20 X | 1 |
| | 120 VHS-C Camcorder 40 X | 1 |
| | 150 8MM Camcorder 20 X | 2 |
| | 2 Hd VCR LCD Menu | 1 |
| | 250 8MM Camcorder 40 X | 1 |
| | 650DL Digital Camcorder 150 X | 1 |
| | AR2 35MM Camera 8 X | 1 |
| | Combo Player - 4 Hd VCR + DVD | 2 |
| | DVD Upgrade Unit for Cent. VCR | 2 |
| | QX Portable CD Player | 1 |
| | R5 Micro Digital Tape Recorder | 2 |
| | ZC Digital PDA - Standard | 1 |
| | ZT Digital PDA - Commercial | 2 |

# Validating a Form With JavaScript

JavaScript is an object-oriented, interpretive language used with HTML to provide dynamic capabilities to otherwise static Web pages.

A JavaScript function included in an HTML page is processed and executed by the Web browser. Integrating a JavaScript function into an HTML page that calls WebFOCUS, or a page that displays a report, enables you to:

- Validate entries on a form before submitting them to WebFOCUS. Validating entries prevents a user from requesting restricted or unavailable data. This topic describes how to code a JavaScript function that checks for a user entry on a form.
- Perform calculations on data returned in a WebFOCUS report. JavaScript offloads data processing demands from the Reporting Server to the Web browser. Chapter 9, *Enhancing a User Interface*, contains an example of a drill-down report illustrating this feature.
- Automatically submit a form or run a procedure when an HTML page loads in a browser.

  For example, suppose you create a page that displays a dynamic report. By including the procedure in a JavaScript function, you can provide text on the page, such as:

  ```
  Please wait; processing your report request. Do not click Stop.
  ```

  When a user accesses the page, the procedure runs automatically, and the informative message displays until the report output is returned to the browser.

  For an example of the use of JavaScript to automatically run a procedure when an HTML page loads, see Chapter 6, *Designing a User Interface for a Web Application With the Report Layout Painter*.

You have many additional design capabilities with JavaScript. See your JavaScript documentation for details on JavaScript techniques and syntax.

## *Example* Checking For a User-Supplied Value on a Form

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a launch page named VALIDATE, which calls a JavaScript function named checkInput. This procedure must be accessible to the Web server.

   The sample launch page uses the CGI.

   **Launch Page: VALIDATE.HTM**

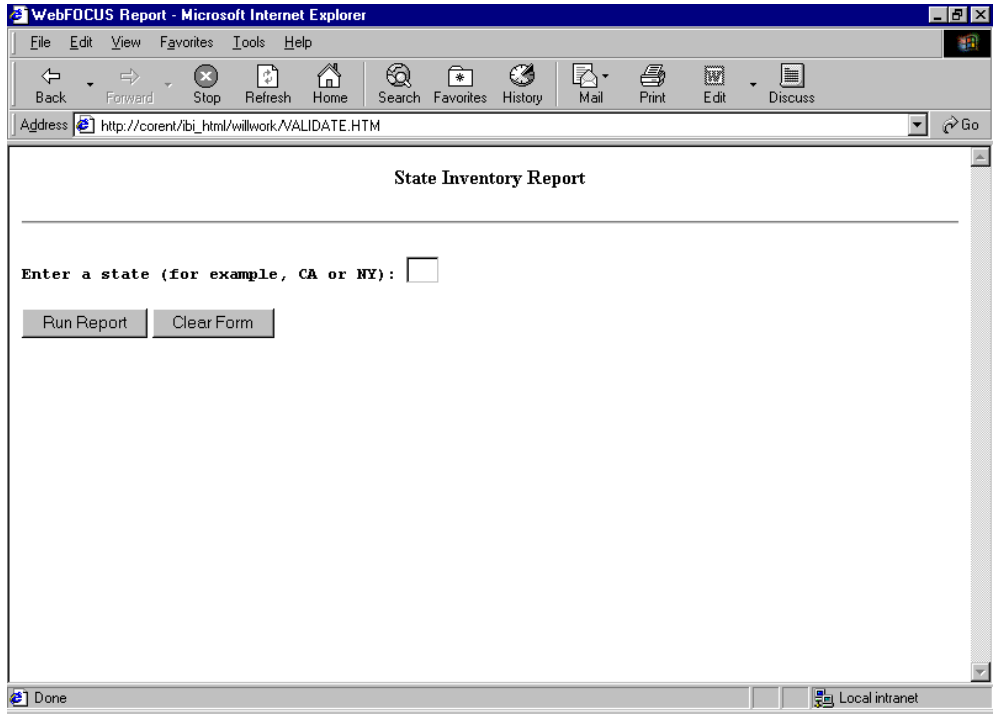
```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
<SCRIPT LANGUAGE=JAVASCRIPT TYPE="TEXT/JAVASCRIPT">
function checkInput() {
  if(document.form2.STATE.value == "") {
    alert("You must enter a value for the state!");
    return false;
  }
    else {
      return true;
    }
  document.form2.submit();
}
</SCRIPT>
</HEAD>
<BODY>
<H4 ALIGN=CENTER>State Inventory Report</H4>
<HR>
<FORM ACTION="/cgi-bin/ibi_cgi/ibiweb.exe" METHOD="get"
TARGET="_blank" NAME="form2" onSubmit="return checkInput()">
<INPUT NAME="IBIF_ex" VALUE="stsales" TYPE="hidden">
<P ALIGN=LEFT NOWRAP><PRE>
<B>Enter a state (for example, CA or NY): </B><INPUT NAME="STATE"
TYPE="text" SIZE="2">
</PRE></P>
<P><PRE></PRE></P>
<P>
<INPUT NAME="submit" TYPE=SUBMIT VALUE="Run Report">
<INPUT NAME="reset" TYPE=RESET VALUE="Clear Form">
</P>
</FORM>
</BODY>
</HTML>
```

2. Create a procedure named STSALES, which generates an inventory report for eMart, in the state the user requests.

   **Procedure: STSALES.FEX**

```
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
WHERE STATE EQ '&STATE'
WHERE SNAME EQ 'eMart'
END
```

3. Run the launch page.



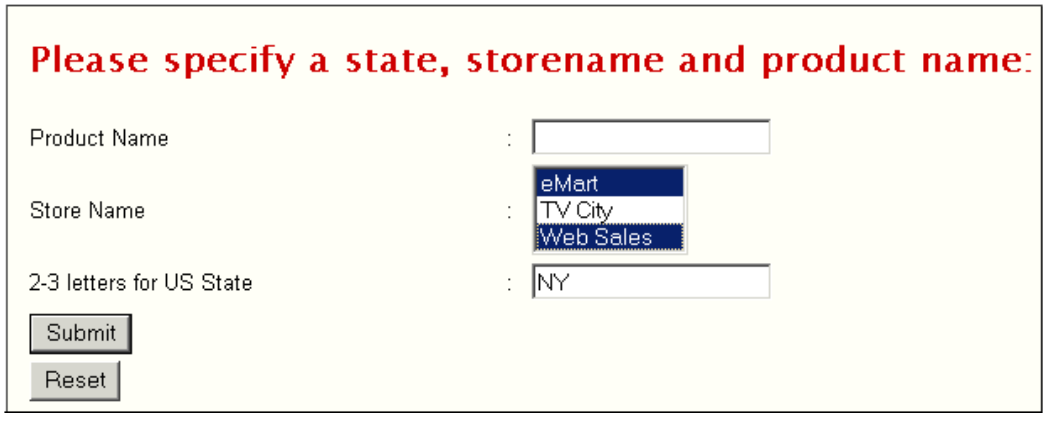

4. Without entering a value for the state, click *Run Report*. The following message displays:

# Amper Auto-Prompting in CGI/Servlet

The amper auto-prompting feature enables you to prompt users for the amper variables necessary to execute a procedure. With this feature, you can also design the form that will prompt for the amper variables.

## Amper Auto-Prompting Examples

The following is an example of how you can enhance a form that will prompt a user for amper variables:




All of the sample procedures that follow use the amper auto-prompting feature. Each example builds on the following simple procedure:

```
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE'
WHERE SNAME EQ '&SNAME'
WHERE PRODNAME EQ '&PRODNAME'
END
```

**Note:** The code that is added in each subsequent procedure appears in bold.

### *Example* Adding a Description

The following procedure provides a description of the procedure. Note that this tag is only visible in the procedure code, not in the HTML form.

```
 -<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE'
WHERE SNAME EQ '&SNAME'
WHERE PRODNAME EQ '&PRODNAME'
END
```

### *Example* Adding a Heading

The following procedure provides a textual prompt for the form.

```
-<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
-<heading> Please specify a state, storename and product name: </heading>
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE'
WHERE SNAME EQ '&SNAME'
WHERE PRODNAME EQ '&PRODNAME'
END
```

### *Example* Setting a Variable Description

The following procedure provides a more descriptive name for the STATE, SNAME, and PRODNAME fields within the form. The field descriptions will precede the field controls on the auto-prompt form.

```
-<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
-<heading> Please specify a state, storename and product name: </heading>
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2-3 letters for US State.'
WHERE SNAME EQ '&SNAME.Store Name.'
WHERE PRODNAME EQ '&PRODNAME.Product Name.'
END
```

### *Example* Setting a Default Variable Value

The following procedure sets a default value of NY for the STATE field. Note that there must be an '&' in front of the field name in the -DEFAULT command for the amper variable to contain a default attribute in the form.

```
-<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
-<heading> Please specify a state, storename and product name: </heading>
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2-3 letters for US State.'
WHERE SNAME EQ '&SNAME.Store Name.'
WHERE PRODNAME EQ '&PRODNAME.Product Name.'
END
```

### *Example* Adding a Single Selectable List of Values

The following procedure provides a list of values that are valid for the SNAME field. In a single selectable list, you can select only one value from the list.

```
-<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
-<heading> Please specify a state, storename and product name: </heading>
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2-3 letters for US State.'
WHERE SNAME EQ '&SNAME.(eMart,TV City,Web Sales).Store Name.'
WHERE PRODNAME EQ '&PRODNAME.Product Name.'
END
```

## *Example* Adding a Multi Selectable List of Values

The following procedure provides a list of values that are valid for the SNAME field. In a multi selectable list, you can select more than one value from the list. An AND or an OR operation can be used for multi selection.

```
-<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
-<heading> Please specify a state, storename and product name: </heading>
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2-3 letters for US State.'
WHERE SNAME EQ '&SNAME.(OR(eMart,TV City,Web Sales)).Store Name.'
WHERE PRODNAME EQ '&PRODNAME.Product Name.'
END
```

## *Example* Adding a Single Selectable Dynamic List of Values

The following procedure provides a list of values that are valid for the product name field. This list is populated with values from the CENTORD data source. In a single selectable list, you can select only one value from the list.

```
-<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2-3 letters for US State.'
WHERE SNAME EQ '&SNAME.(AND(eMart,TV City,Web Sales)).Store Name.'
WHERE PRODNAME EQ '&PRODNAME.(FIND PRODNAME IN CENTORD).Product Name.'
END
```

## *Example* Adding a Multi Selectable Dynamic List of Values

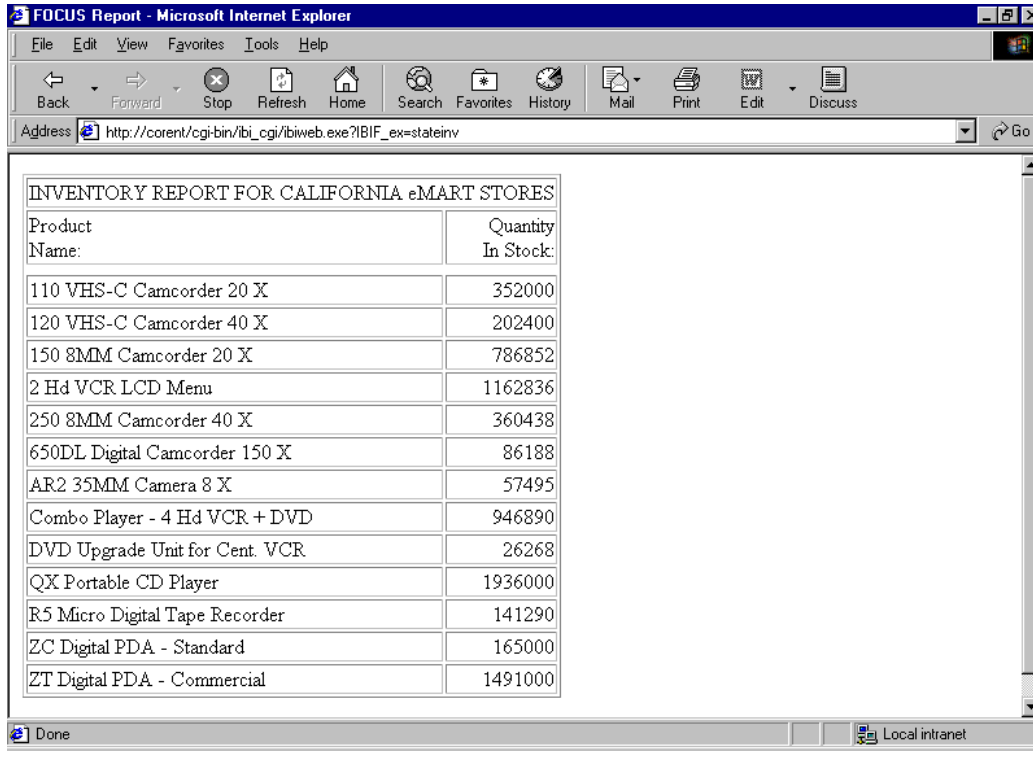
The following procedure provides a list of values that are valid for the state field. This list is populated with values from the CENTORD data source. In a multi selectable list, you can select more than one value from the list. To select more than one value, hold the SHIFT key or the CTRL key while clicking on the options.

```
-<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
-<heading> Please specify a state, storename and product name: </heading>
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY SNAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.(OR(FIND STATE IN CENTORD)).2-3 letters for US
State.'
WHERE SNAME EQ '&SNAME.(AND(eMart,TV City,Web Sales)).Store Name.'
WHERE PRODNAME EQ '&PRODNAME.(FIND PRODNAME IN CENTORD).Product Name.'
END
```

# Displaying a Report on the Default WebFOCUS Page

The simplest way to display a report is to use the default WebFOCUS page. You do not need to design the page on which the report will appear; WebFOCUS automatically generates an HTML page that contains the necessary tagging. The report output is formatted on the page using the HTML <TABLE tag.

Following is a report displayed on the default WebFOCUS page:




| INVENTORY REPORT FOR CALIFORNIA eMART STORES | |
|---|---|
| Product Name: | Quantity In Stock: |
| 110 VHS-C Camcorder 20 X | 352000 |
| 120 VHS-C Camcorder 40 X | 202400 |
| 150 8MM Camcorder 20 X | 786852 |
| 2 Hd VCR LCD Menu | 1162836 |
| 250 8MM Camcorder 40 X | 360438 |
| 650DL Digital Camcorder 150 X | 86188 |
| AR2 35MM Camera 8 X | 57495 |
| Combo Player - 4 Hd VCR + DVD | 946890 |
| DVD Upgrade Unit for Cent. VCR | 26268 |
| QX Portable CD Player | 1936000 |
| R5 Micro Digital Tape Recorder | 141290 |
| ZC Digital PDA - Standard | 165000 |
| ZT Digital PDA - Commercial | 1491000 |

If you save the output from a dynamic report using the command ON TABLE HOLD FORMAT HTML in a procedure, you create a *static report*, which can be called and displayed.

In applications where users do not require real-time information, static reports can substantially reduce overhead on the system. Reports can then be refreshed during off-peak times.

## *Procedure* How to Display a Dynamic Report on the Default WebFOCUS Page

1. Create a procedure that generates a dynamic report.
2. Create a launch page from which a user can run the report.
3. Run the report from the launch page. WebFOCUS returns the report to the browser as a complete HTML file. You can view it using browser options, for example, *View/Source* in Microsoft Internet Explorer.

## *Example* Displaying a Dynamic Report on the Default WebFOCUS Page

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a procedure named STATEINV, which generates a dynamic report. This report must be accessible to the WebFOCUS Reporting Server.

   **Procedure: STATEINV.FEX**

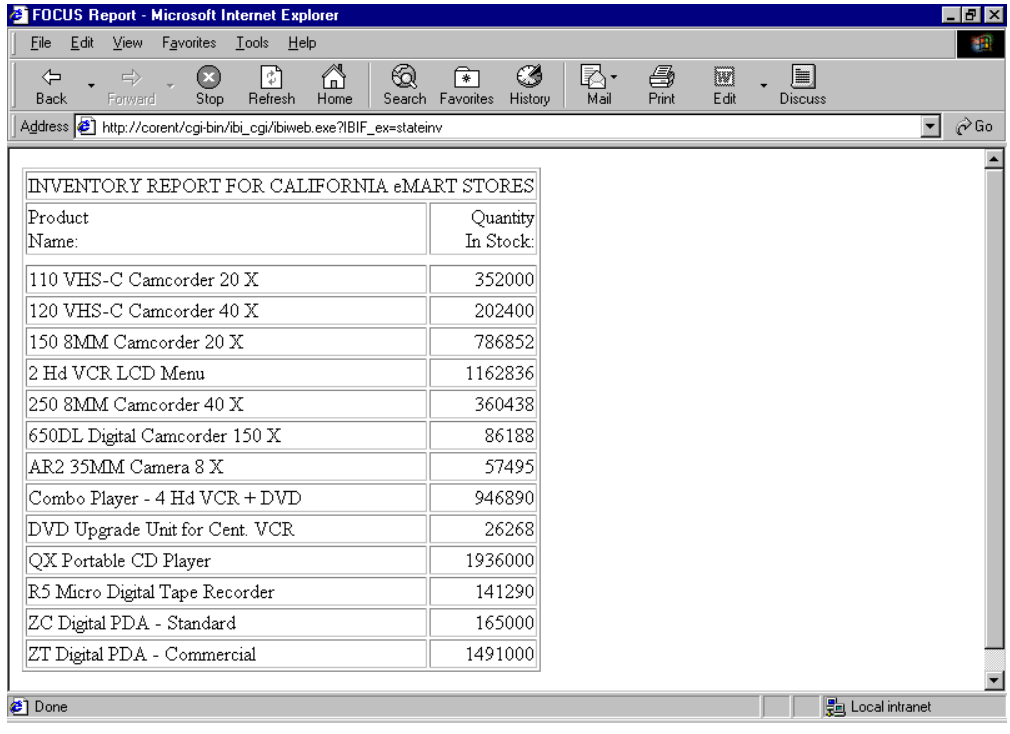
   ```
   SET PAGE-NUM = OFF
   TABLE FILE CENTORD
   SUM QTY_IN_STOCK BY PRODNAME
   ON TABLE SUBHEAD
   "INVENTORY REPORT FOR CALIFORNIA eMART STORES"
   WHERE STATE EQ 'CA'
   WHERE SNAME EQ 'eMart'
   END
   ```

2. Create a launch page from which a user can run the report. The following is a simple launch page named RUNDYNAM. It uses the WebFOCUS Servlet:

   **Launch Page: RUNDYNAM.HTM**

   ```
   <HTML>
   <BODY>
   <A HREF="http://server1/servlet/WFServlet?IBIF_ex=stateinv">Click here
   to run an inventory report for eMart in California.</A>
   </BODY>
   </HTML>
   ```

**3.** Run the launch page, and click the link. The dynamic report displays in the browser on the default WebFOCUS page:




| INVENTORY REPORT FOR CALIFORNIA eMART STORES | |
|---|---|
| Product Name: | Quantity In Stock: |
| 110 VHS-C Camcorder 20 X | 352000 |
| 120 VHS-C Camcorder 40 X | 202400 |
| 150 8MM Camcorder 20 X | 786852 |
| 2 Hd VCR LCD Menu | 1162836 |
| 250 8MM Camcorder 40 X | 360438 |
| 650DL Digital Camcorder 150 X | 86188 |
| AR2 35MM Camera 8 X | 57495 |
| Combo Player - 4 Hd VCR + DVD | 946890 |
| DVD Upgrade Unit for Cent. VCR | 26268 |
| QX Portable CD Player | 1936000 |
| R5 Micro Digital Tape Recorder | 141290 |
| ZC Digital PDA - Standard | 165000 |
| ZT Digital PDA - Commercial | 1491000 |

## *Procedure* How to Display a Static Report on the Default WebFOCUS Page

1. Create a procedure:

   a. Identify the HTML file that will contain the report output and save it to the designated location. For example, on Windows NT/2000/XP, use a FILEDEF command. See Chapter 16, *Defining and Allocating WebFOCUS Files*, for details on platform-specific commands.

      The Web server must be able to locate the file for browser display, so specify a location under a Web server alias.

   b. Include the following command

      ```
      ON TABLE HOLD FORMAT HTML AS report
      ```

      where:

      *report* is the name of a virtual file that contains the report output. The name can be from 1 to 8 characters. Do not include an extension. This file is not saved to disk.

2. Run the procedure to create the report as an HTML file. There are several ways to do this, including:

   - Call the WebFOCUS Client. See *Displaying a Static Report on the Default WebFOCUS Page* on page 8-40 for an example.
   - Schedule the report to run with the WebFOCUS ReportCaster. See your ReportCaster documentation for details.
   - In WebFOCUS, access the following default directory

     **Windows NT/2000/XP:** *install_drive*:\ibi\srv52\wfs\bin

     **UNIX:** /ibi/srv52/wfs/bin

     **OS/390:** /ibi/srv52/wfs/bin

     and issue the following command:

     ```
     edastart -t
     ```

     Execute the procedure from the WebFOCUS prompt >>:

     ```
     ex procedure_name
     ```

     To return to the WebFOCUS directory, type:

     ```
     fin
     ```

3. Create a launch page with a hyperlink that links to the report.

## *Example* Displaying a Static Report on the Default WebFOCUS Page

This example runs on WebFOCUS for Windows NT. If you are using another platform, substitute the appropriate platform-specific command for the FILEDEF command. See Chapter 16, *Defining and Allocating WebFOCUS Files*, for details.

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a procedure that saves a report as an HTML file. The following sample procedure is named STATDATA. The letters on the left correspond to the notes explaining the code.

   **Procedure: STATDATA.FEX**

```
a. FILEDEF STATRPT DISK
      E:\IBI\WEBFOCUS52\IBI_HTML\NCP\STATRPT.HTM
   TABLE FILE GGORDER
   HEADING
   "Orders Summary"
   SUM QUANTITY AS 'Units Ordered'
   BY ORDER_DATE AS 'Order Date'
   BY PRODUCT_DESCRIPTION
   WHERE ORDER_DATE EQ '01/01/96'
b. ON TABLE HOLD FORMAT HTML AS STATRPT
   END
```

   **a.** This command allocates the file STATRPT and saves it to a designated directory.

   **b.** This command saves the report output as an HTML file named STATRPT.HTM.

2. Run the procedure STATDATA to create and save STATRPT. One way to do this is to call the WebFOCUS Client.

   The following simple HTML page uses the CGI to run STATDATA. An administrator can use this kind of page to create the static reports that users will access.

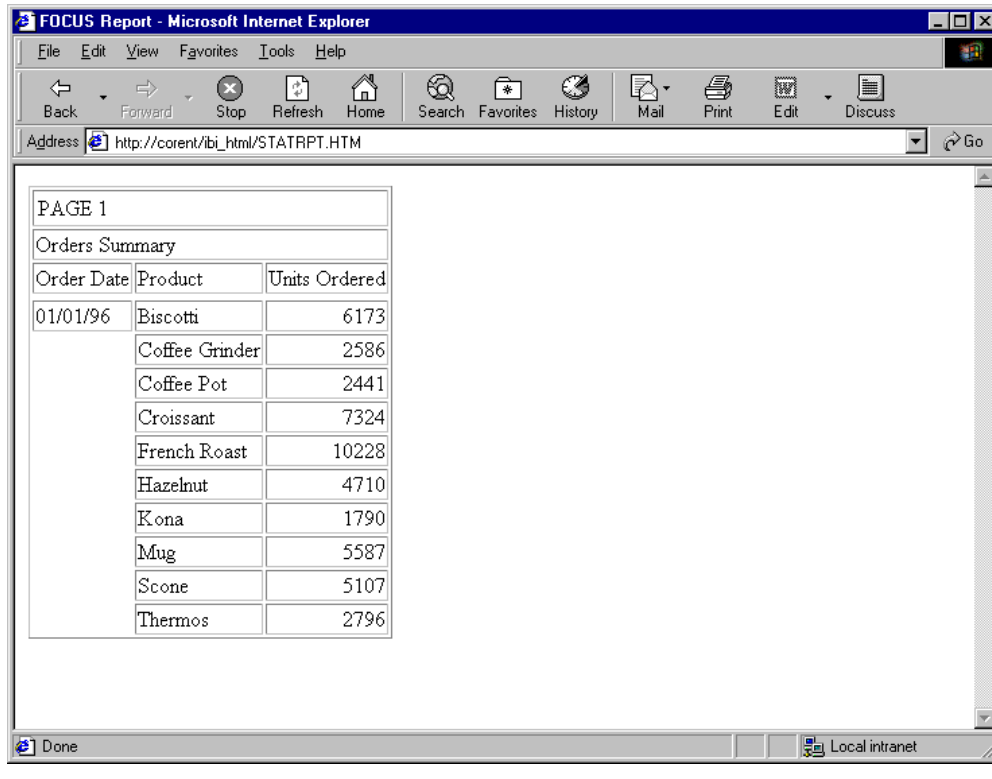
```
<HTML>
<BODY>
<A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=statdata">Click here
to run STATDATA and create a static report stored as STATRPT.HTM.</A>
</BODY>
</HTML>
```

   WebFOCUS displays a message indicating that an HTML file was saved.

3. Create a launch page from which a user can run the report. The following is a simple launch page named STATIC:

```
<HTML>
<BODY>
<A HREF="STATRPT.HTM">Click here to view the Orders Summary report</A>
</BODY>
</HTML>
```

4. Run the launch page, and click the link. The static report displays in the browser on the default WebFOCUS page on which it was saved:




| PAGE 1 | | |
|---|---|---|
| Orders Summary | | |
| Order Date | Product | Units Ordered |
| 01/01/96 | Biscotti | 6173 |
| | Coffee Grinder | 2586 |
| | Coffee Pot | 2441 |
| | Croissant | 7324 |
| | French Roast | 10228 |
| | Hazelnut | 4710 |
| | Kona | 1790 |
| | Mug | 5587 |
| | Scone | 5107 |
| | Thermos | 2796 |

# Designing an HTML Page for Report Display

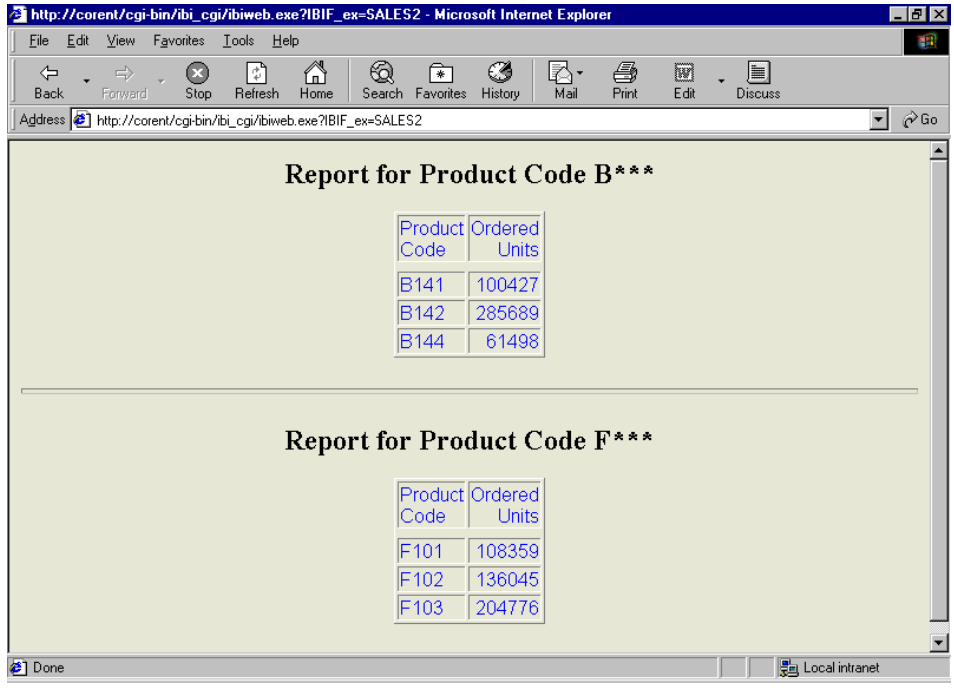
Use the Dialogue Manager command -HTMLFORM to design the HTML page on which a report displays. You control page style and format, and can imaginatively enhance a report. A custom HTML page can include HTML elements such as frames, tables, and graphic images. It must be a complete HTML page with all required HTML tags. It also contains a special HTML comment that indicates where to insert the WebFOCUS report output.

You can do one of the following:

- Call an external, existing HTML file that defines the display page. This method uses the command -HTMLFORM *filename* in a procedure. The procedure is in one file, and the HTML page is in another file.
- Create a custom HTML page within the procedure, using the commands -HTMLFORM BEGIN and –HTMLFORM END. In this topic, this type of page is called *inline*.

You can display a single report or multiple reports on a page. A report can be dynamic or static.

Following are multiple reports displayed on a custom HTML page:




For more information on -HTMLFORM and its capabilities, see *Enhancing an HTML Web Page With a Procedure* in Chapter 10, *Managing Flow of Control in an Application*.

## *Procedure* How to Display a Dynamic Report on an Existing HTML Page

1. Create the HTML page that will incorporate the report output. Use the following comment to indicate where the output will display

   ```
   !IBI.FIL.report;
   ```

   or

   ```
   <!--WEBFOCUS TABLE report>
   ```

   where:

   *report*

   Is the name of a virtual file that holds the report output. That file is generated in step 2.

   The Reporting Server must be able to locate the page using APP PATH or EDAPATH. See *Storing and Searching for Application Files* on page 2-12, for details on search paths.

   When the page displays, the comment is replaced by report output.

2. Create a procedure:

   **a.** Include the following command

   ```
   ON TABLE HOLD FORMAT HTMTABLE AS report
   ```

   where:

   *report* is the name of a virtual file that contains the report output. The name can be from 1 to 8 characters. Do not include an extension. This file is not saved to disk.

   **b.** At the end of the procedure, include the Dialogue Manager command

   ```
   -HTMLFORM filename
   ```

   where:

   *filename* is the name of the HTML page that will incorporate the report output. You create this page in step 1.

3. Create a launch page from which a user can run the report. WebFOCUS incorporates the report output into the HTML page you designed, and returns a complete HTML file to the browser.

## *Example* Displaying a Dynamic Report on an Existing HTML Page

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create an HTML page named RPTPG.HTM. The Reporting Server must be able to locate the page using APP PATH or EDAPATH. See Chapter 2, *WebFOCUS Application Logic*, for details on search paths.

   The letters on the left correspond to the notes explaining the code.

   **HTML Page: RPTPG.HTM**

```
    <HTML>
    <BODY BGCOLOR="#CCCCCC">
    <FONT FACE="Arial" COLOR="Black">
    <H2>Orders Summary</H2>
a.  !IBI.FIL.ORDERS;
    </BODY>
    </HTML>
```

   **a.** WebFOCUS reads the HTML comment and replaces it with the report output held in ORDERS (step 2).

2. Create a procedure named ORDERS. The letters on the left correspond to the notes explaining the code.

   **Procedure: ORDERS.FEX**

```
   SET PAGE-NUM = OFF
   TABLE FILE GGORDER
   SUM QUANTITY AS 'Units Ordered'
   BY HIGHEST 1 ORDER_DATE AS 'Order Date'
   BY PRODUCT_DESCRIPTION
a. ON TABLE HOLD FORMAT HTMTABLE AS ORDERS
   END
b. -HTMLFORM RPTPG
```

   **a.** This command saves the report output to a virtual file in HTML format. The file is named ORDERS.

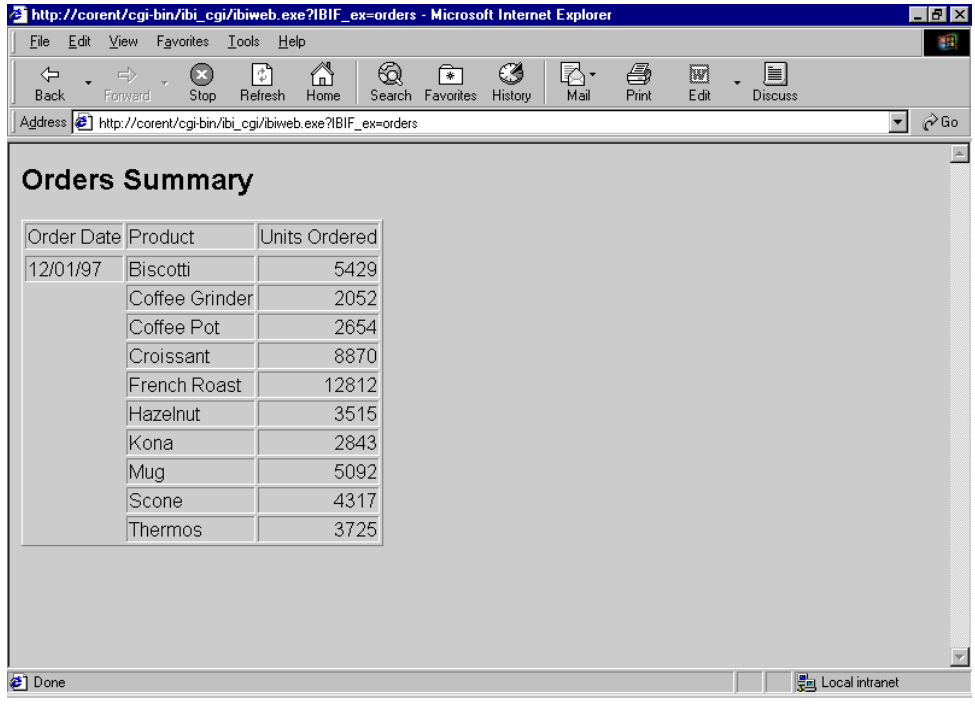
   **b.** This command sends the report output held in ORDERS to the HTML page named RPTPG.HTM (step 1).

3. Create a launch page from which a user can run the report. The following simple launch page is named RUNRPT.HTM. It uses ISAPI:

   **Launch Page: RUNRPT.HTM**

```
<HTML>
<BODY>
<A HREF="/cgi-bin/ibi_cgi/webapi.dll?IBIF_ex=orders">Click here for
the Orders Summary report.</A>
</BODY>
</HTML>
```

4. Run the launch page, and click the link. The dynamic report displays on the custom HTML page:



## *Syntax* How to Display a Report on an Inline HTML Page

```
-HTMLFORM BEGIN
html_code
html_code
html_code
.
.
.
-HTMLFORM END
```

where:

*html_code*

Is a line of standard HTML code. It cannot exceed 80 characters.

## *Example* Displaying a Dynamic Report on an Inline HTML Page

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a procedure named ORDERS2. The letters on the left correspond to the notes explaining the code.

   **Procedure: ORDERS2**

```
    SET PAGE-NUM = OFF
    TABLE FILE GGORDER
    SUM QUANTITY AS 'Units Ordered'
    BY HIGHEST 1 ORDER_DATE AS 'Order Date'
    BY PRODUCT_DESCRIPTION
a. ON TABLE HOLD FORMAT HTMTABLE AS ORDERS2
    END
b. -HTMLFORM BEGIN
   <HTML><HEAD><STYLE>TD {FONT-FAMILY: ARIAL; COLOR:
    BLUE;}</STYLE></HEAD>
    <BODY BGCOLOR="FAEBD7"><DIV ALIGN="CENTER">
c. !IBI.FIL.ORDERS2;
    </DIV></BODY></HTML>
b. -HTMLFORM END
```

   **a.** This command saves the report output to a virtual file in HTML format. The file is named ORDERS2.

   **b.** These commands delimit the inline HTML page that defines style and format. The lines in-between the commands are standard HTML code. Each line is less than 80 characters.

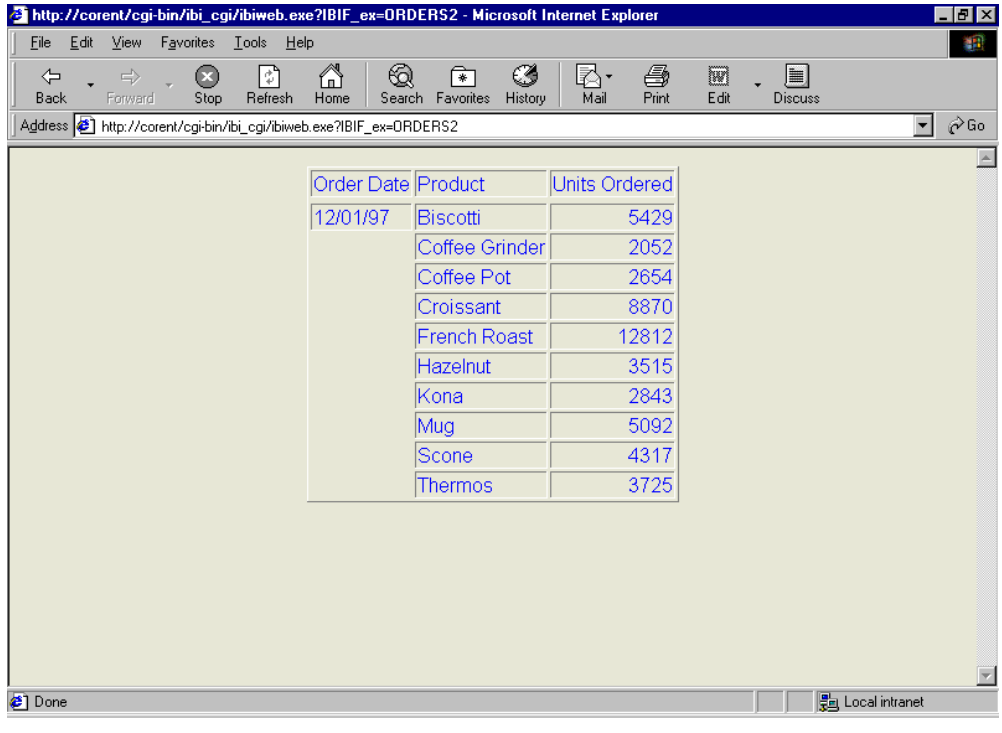
   **c.** WebFOCUS reads this HTML comment and replaces it with the report output held in ORDERS2.

2. Create a launch page from which a user can run the report. The following simple launch page is named RUNRPT.HTM. It uses ISAPI:

   **Launch Page: RUNRPT.HTM**

```
<HTML>
<BODY>
<A HREF="/cgi-bin/ibi_cgi/webapi.dll?IBIF_ex=orders2">Click here for
the Orders Summary report.</A>
</BODY>
</HTML>
```

3. Run the launch page, and click the link. The dynamic report displays on the custom inline HTML page:




| Order Date | Product | Units Ordered |
|---|---|---|
| 12/01/97 | Biscotti | 5429 |
| | Coffee Grinder | 2052 |
| | Coffee Pot | 2654 |
| | Croissant | 8870 |
| | French Roast | 12812 |
| | Hazelnut | 3515 |
| | Kona | 2843 |
| | Mug | 5092 |
| | Scone | 4317 |
| | Thermos | 3725 |

## *Example* Displaying Two Dynamic Reports on an Existing HTML Page

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create an HTML page named FIRST.HTM. The Reporting Server must be able to locate the page using APP PATH or EDAPATH. See *Storing and Searching for Application Files* on page 2-12, for details on search paths.

   The letters on the left correspond to the notes explaining the code.

   **HTML Page: FIRST.HTM**

```
    <HTML>
    <BODY BGCOLOR="#CCCCCC">
a. <H2>Report for Product Code B***</H2>
b. <!--WEBFOCUS TABLE UPPER>
    <HR SIZE=5>
a. <H2>Report for Product Code F***</H2>
b. <!--WEBFOCUS TABLE LOWER>
    </BODY>
    </HTML>
```

   **a.** This text identifies the two reports on the HTML page.

   **b.** WebFOCUS reads these HTML comments. UPPER and LOWER are the names assigned to the virtual files in step 2. On the HTML page, WebFOCUS replaces the comments with the designated report output.

2. Create a procedure named TWOSALES. The letters on the left correspond to the notes explaining the code.

   **Procedure: TWOSALES.FEX**

```
   SET PAGE-NUM = OFF
   TABLE FILE GGORDER
   SUM QUANTITY BY PCD
   IF PCD EQ 'B$$$'
a. ON TABLE HOLD FORMAT HTMTABLE AS UPPER
   END
   TABLE FILE GGORDER
   SUM QUANTITY BY PCD
   IF PCD EQ 'F$$$'
a. ON TABLE HOLD FORMAT HTMTABLE AS LOWER
   END
b. -RUN
c. -HTMLFORM FIRST
```

   **a.** These commands generate and save the report output from each request to virtual files in HTML format. The files are named UPPER and LOWER.

   **b.** This command runs the report requests.

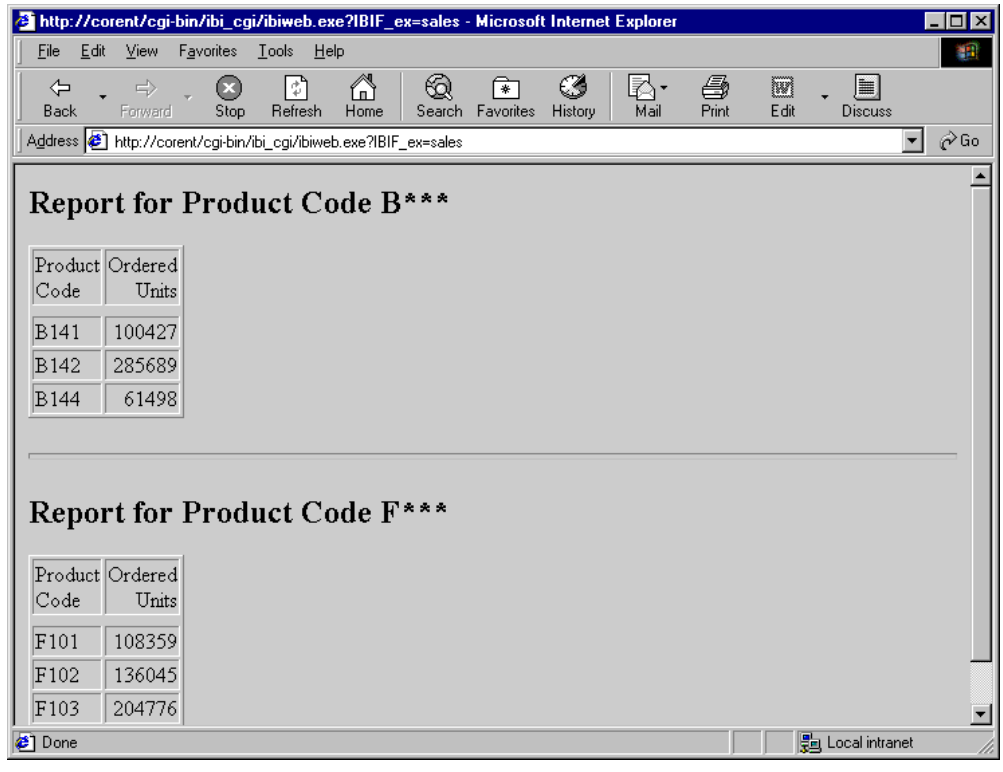
   **c.** This command sends the report output to the HTML page named FIRST.HTM. You create this page in step 1.

3. Create a launch page from which a user runs the reports. The following simple launch page is called MULTIRPT.HTM. It uses the CGI:

   **Launch Page: MULTIRPT.HTM**

```
<HTML>
<BODY>
<A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=twosales">Click here for
sales reports for two product codes.</A>
</BODY>
</HTML>
```

**4.** Run the launch page, and click the link. The two dynamic reports display on the custom HTML page:



## *Example* Displaying Two Dynamic Reports on an Inline HTML Page

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a procedure named SALES2. The letters on the left correspond to the notes explaining the code.

   **Procedure: SALES2.FEX**

```
     SET PAGE-NUM = OFF
     TABLE FILE GGORDER
     SUM QUANTITY BY PCD
     IF PCD EQ 'B$$$'
a.   ON TABLE HOLD FORMAT HTMTABLE AS UPPER
     END
     TABLE FILE GGORDER
     SUM QUANTITY BY PCD
     IF PCD EQ 'F$$$'
a.   ON TABLE HOLD FORMAT HTMTABLE AS LOWER
     END
b.  -RUN
c.  -HTMLFORM BEGIN
    <HTML><HEAD><STYLE>TD {FONT-FAMILY: ARIAL; COLOR:
     BLUE;}</STYLE></HEAD>
    <BODY BGCOLOR="FAEBD7"><DIV ALIGN="CENTER">
   <H2>Report for Product Code B***</H2>
d.  <!--WEBFOCUS TABLE UPPER>
    <HR SIZE=5>
    <H2>Report for Product Code F***</H2>
d.  <!--WEBFOCUS TABLE LOWER>
    </DIV></BODY></HTML>
c.  -HTMLFORM END
```

   **a.** These commands generate and save the report output from each request to virtual files in HTML format. The files are named UPPER and LOWER.

   **b.** This command runs the report requests.

   **c.** These commands delimit the inline HTML page that defines style and format. The lines in-between the commands are standard HTML code. Each line is less than 80 characters.

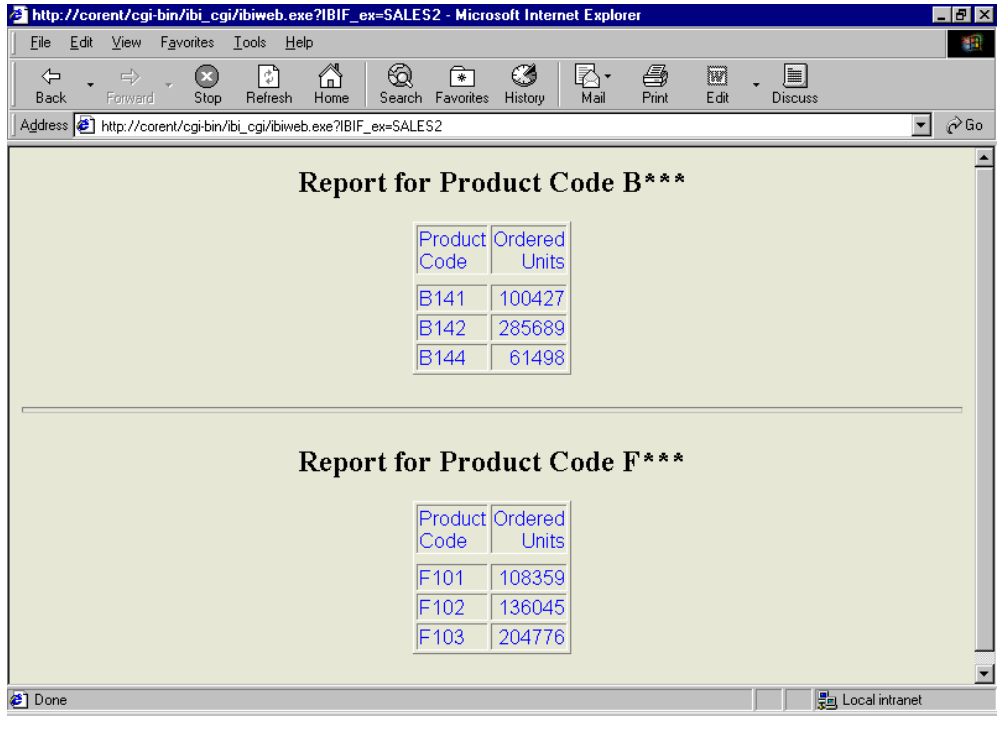
   **d.** WebFOCUS reads these HTML comments. UPPER and LOWER are the names assigned to the virtual files. On the HTML page, WebFOCUS replaces the comments with the designated report output.

2. Create a launch page from which a user can run the reports. The following simple launch page is called MULTIRPT.HTM. It uses the CGI:

   **Launch Page: MULTIRPT.HTM**

```
<HTML>
<BODY>
<A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=sales2">Click here for
sales reports for two product codes.</A>
</BODY>
</HTML>
```

3. Run the launch page, and click the link. The two dynamic reports display on the custom inline HTML page:



## *Procedure* How to Display a Static Report on an Existing HTML Page

To create a procedure:

1. Allocate the HTML file that will combine the report output and the custom HTML page.

   For example, on Windows NT/2000/XP, the command is

   ```
   FILEDEF htmlpage DISK drive:\directory\htmlpage.HTM
   ```

   where:

   *htmlpage*

   Is the name of the combined report output and custom HTML page file. The file name can be from 1 to 8 characters.

   *drive*

   Is the drive on which to save the file.

   *directory*

   Is the directory in which to save the file.

   For other platform-specific commands, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

2. Include the following command

   ```
   ON TABLE HOLD FORMAT HTMTABLE AS report
   ```

   where:

   *report*

   Is the name of a virtual file that contains the report output. The name can be from 1 to 8 characters. Do not include an extension. This file is not saved to disk.

3. At the end of the procedure, include the Dialogue Manager command

   ```
   -HTMLFORM filename SAVE AS htmlpage
   ```

   where:

   *filename*

   Is the name of the custom HTML page that will incorporate the report output. You create this page in step 2. The Reporting Server must be able to locate the page using APP PATH or EDAPATH. See *Storing and Searching for Application Files* on page 2-12, for details on search paths.

   *htmlpage*

   Is the name of the HTML file that will combine the report output and the custom HTML page. The Web server must be able to locate this file.

4. Create the custom HTML page. Add the following comment to indicate where the output will display

   ```
   !IBI.FIL.report;
   ```

   or

   ```
   <!--WEBFOCUS TABLE report>
   ```

   where:

   *report*

   Is the name of the virtual file that holds the report output. That file was created in step 1.

   The Reporting Server must be able to locate the page using APP PATH or EDAPATH. See *Storing and Searching for Application Files* on page 2-12, for details on search paths.

   On the HTML page, WebFOCUS replaces the comment with the report output.

5. Run the procedure from step 1 to create the static report on the custom HTML page.
6. Create a launch page with a hyperlink that links to the report.

## *Example* Displaying a Static Report on an Existing HTML Page

This example runs on WebFOCUS for Windows NT. If you are using another platform, substitute the appropriate platform-specific command for the FILEDEF command. See Chapter 16, *Defining and Allocating WebFOCUS Files*, for more information.

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a procedure named CONTACTS. The letters on the left correspond to the notes explaining the code.

   **Procedure: CONTACTS.FEX**

   ```
   a. FILEDEF WEBPAGE DISK
          E:\IBI\WEBFOCUS52\IBI_HTML\APPDIR\WEBPAGE.HTM
      SET PAGE-NUM = OFF
      TABLE FILE GGSTORES
      PRINT STORE_NAME ADDRESS1
      BY CITY
   b. ON TABLE HOLD FORMAT HTMTABLE AS AREPORT
      END
   c. -HTMLFORM DATAOUT SAVE AS WEBPAGE
   ```

**a.** This command allocates the file WEBPAGE.HTM and saves it to a directory accessible to the Web server. It will combine the report output and the custom HTML page.

**b.** This command saves the report output to a virtual file in HTML format. The file is named AREPORT. It is not saved on disk.

**c.** This command merges the report output with the custom HTML page named DATAOUT, which you create in step 2. It also specifies a second HTML file, WEBPAGE, in which the combined HTML from the report output and the custom page are saved.

**2.** Create a custom HTML page named DATAOUT.HTM. The Reporting Server must be able to locate the page using APP PATH or EDAPATH. See *Storing and Searching for Application Files* on page 2-12, for details on search paths.

**HTML Page: DATAOUT.HTM**

```
    <HTML>
    <BODY BGCOLOR="#CCCCCC">
a.  <H2>Store Contacts</H2>
b.  <!--WEBFOCUS TABLE AREPORT>
    </BODY>
    </HTML>
```

**a.** This text identifies the report.

**b.** WebFOCUS reads the HTML comment and replaces it with the report output held in AREPORT.

**3.** Run the procedure CONTACTS to create the static report on the custom HTML page. One way to do this is to call the WebFOCUS Client:

```
<HTML>
<BODY>
<A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=contacts">Click here
to run CONTACTS and create a static report.</A>
</BODY>
</HTML>
```

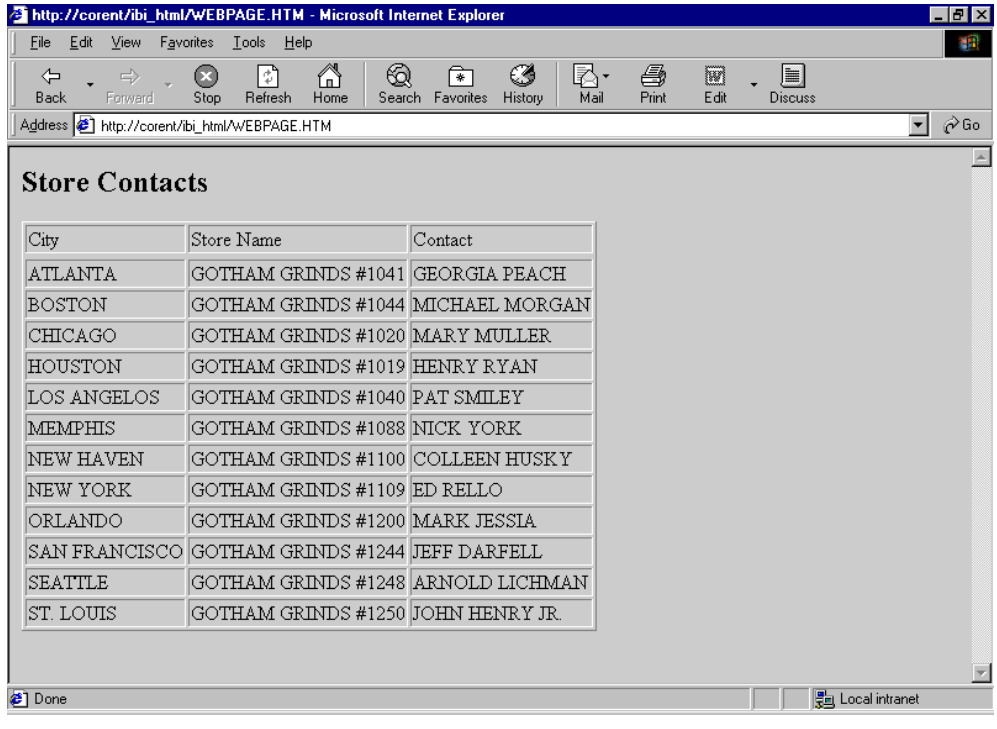
The message Done in your browser indicates that file was created.

4. Create a launch page from which a user can run the report. This simple launch page is named ONESTAT.HTM:

   **Launch Page: ONESTAT.HTM**

```
<HTML>
<BODY>
<A HREF="WEBPAGE.HTM">Click here to view store contacts.</A>
</BODY>
</HTML>
```

5. Run the launch page, and click the link. The static report displays on the custom HTML page:




| City | Store Name | Contact |
|---|---|---|
| ATLANTA | GOTHAM GRINDS #1041 | GEORGIA PEACH |
| BOSTON | GOTHAM GRINDS #1044 | MICHAEL MORGAN |
| CHICAGO | GOTHAM GRINDS #1020 | MARY MULLER |
| HOUSTON | GOTHAM GRINDS #1019 | HENRY RYAN |
| LOS ANGELOS | GOTHAM GRINDS #1040 | PAT SMILEY |
| MEMPHIS | GOTHAM GRINDS #1088 | NICK YORK |
| NEW HAVEN | GOTHAM GRINDS #1100 | COLLEEN HUSKY |
| NEW YORK | GOTHAM GRINDS #1109 | ED RELLO |
| ORLANDO | GOTHAM GRINDS #1200 | MARK JESSIA |
| SAN FRANCISCO | GOTHAM GRINDS #1244 | JEFF DARFELL |
| SEATTLE | GOTHAM GRINDS #1248 | ARNOLD LICHMAN |
| ST. LOUIS | GOTHAM GRINDS #1250 | JOHN HENRY JR. |

## *Example* Displaying Two Static Reports on an Existing HTML Page

This example runs on WebFOCUS for Windows NT. If you are using another platform, substitute the appropriate platform-specific command for the FILEDEF command. See Chapter 16, *Defining and Allocating WebFOCUS Files*, for more information.

**Note:** For information on where to store the files created in this example, see Chapter 16, *Defining and Allocating WebFOCUS Files*.

1. Create a procedure named DEMOG. The letters on the left correspond to the notes explaining the code.

   **Procedure: DEMOG.FEX**

```
a. FILEDEF RPTPAGE DISK
      E:\IBI\WEBFOCUS52\IBI_HTML\APPDIR\RPTPAGE.HTM
   SET PAGE-NUM = OFF
   TABLE FILE GGDEMOG
   SUM HH AS 'Number of,Households' AVGHHSZ98 AS 'Avg.,Size'
   MEDHHI98 AS 'Avg.,Income'
   BY ST
   WHERE ST EQ 'CA'
b. ON TABLE HOLD FORMAT HTMTABLE AS UPPER
   END
   TABLE FILE GGDEMOG
   SUM HH AS 'Number of,Households' AVGHHSZ98 AS 'Avg.,Size'
   MEDHHI98 AS 'Avg.,Income'
   BY ST
   WHERE ST EQ 'NY'
c. ON TABLE HOLD FORMAT HTMTABLE AS LOWER
   END
   -RUN
d. -HTMLFORM RPTOUT SAVE AS RPTPAGE
```

   **a.** This command allocates the file RPTPAGE.HTM and saves it to a directory accessible to the Web server. It will combine the report output and the custom HTML page.

   **b.** This command saves the output from the first request to a virtual file named UPPER.

   **c.** This command saves the output from the second request to a virtual file named LOWER.

   **d.** This command merges the report output with the custom HTML page named RPTOUT, which you create in step 2. It also specifies a second HTML file, RPTPAGE, in which the combined HTML from the report output and the custom page are saved.

2. Create a custom HTML page named RPTOUT.HTM. The Reporting Server must be able to locate the page using APP PATH or EDAPATH. See *Storing and Searching for Application Files* on page 2-12, for details on search paths.

   **HTML Page: RPTOUT.HTM**

```
    <HTML>
    <BODY BGCOLOR="#CCCCCC">
a.  <H3>Demographic Report for California</H3>
b.  <!--WEBFOCUS TABLE UPPER>
    <HR SIZE=5>
a.  <H3>Demographic Report for New York</H3>
b.  <!--WEBFOCUS TABLE LOWER>
    </BODY>
    </HTML>
```

   **a.** This text identifies the two reports.

   **b.** WebFOCUS reads these HTML comments and replaces UPPER and LOWER with the report output.

3. Run the procedure DEMOG to create the static reports on the custom HTML page. One way to do this is to call the WebFOCUS Client:

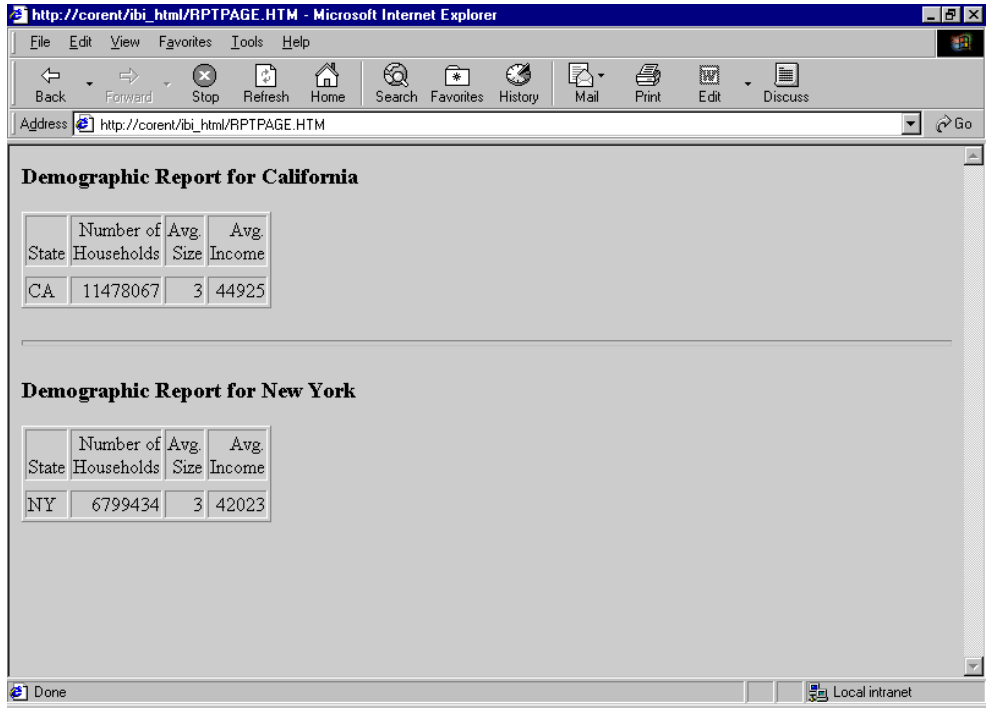
```
<HTML>
<BODY>
<A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=demog">Click here
to run DEMOG and create two static reports.</A>
</BODY>
</HTML>
```

4. Create a launch page from which a user can run the reports. This simple launch page is called TWOSTATS.HTM:

   **Launch Page: TWOSTATS.HTM**

```
<HTML>
<BODY>
<A HREF="RPTPAGE.HTM">Click here to view the demographic reports.</A>
</BODY>
</HTML>
```

**5.** Run the launch page, and click the link. The static reports display on the custom HTML page:



CHAPTER 9

# Enhancing a User Interface

**Topics:**

**Report Display and Manipulation**

- 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

**Launch Page Features**

- Passing a User ID From HTML for a Custom Menu

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.

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.

# Displaying a Report in a Helper Application

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 NT 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.

| MIME Type | Application | File Extension |
|---|---|---|
| application/postscript | GhostView | .PS |
| application/x-prn | Printer (using PCL printer driver) | .PRN |
| application/x-dif | Microsoft Excel or other spreadsheet | .DIF |
| application/x-doc | Microsoft Word | .DOC |

| MIME Type | Application | File Extension |
|---|---|---|
| application/x-xls<br>application/vnd.ms-excl | Microsoft Excel | .XLS, .E97, .WK1, .XHT |
| application/pdf | Adobe Acrobat Reader | .PDF |
| text/plain | Microsoft Notepad or other text editor | .WP or .HTS |
| text/html | Native browser format | .HTML |
| text/htm | Native browser format | .HTM |
| image/gif | Native browser format | .GIF |
| image/jpeg | Native browser format | .JPG, .JPEG |
| XML | Native browser format | .XML |

## *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.

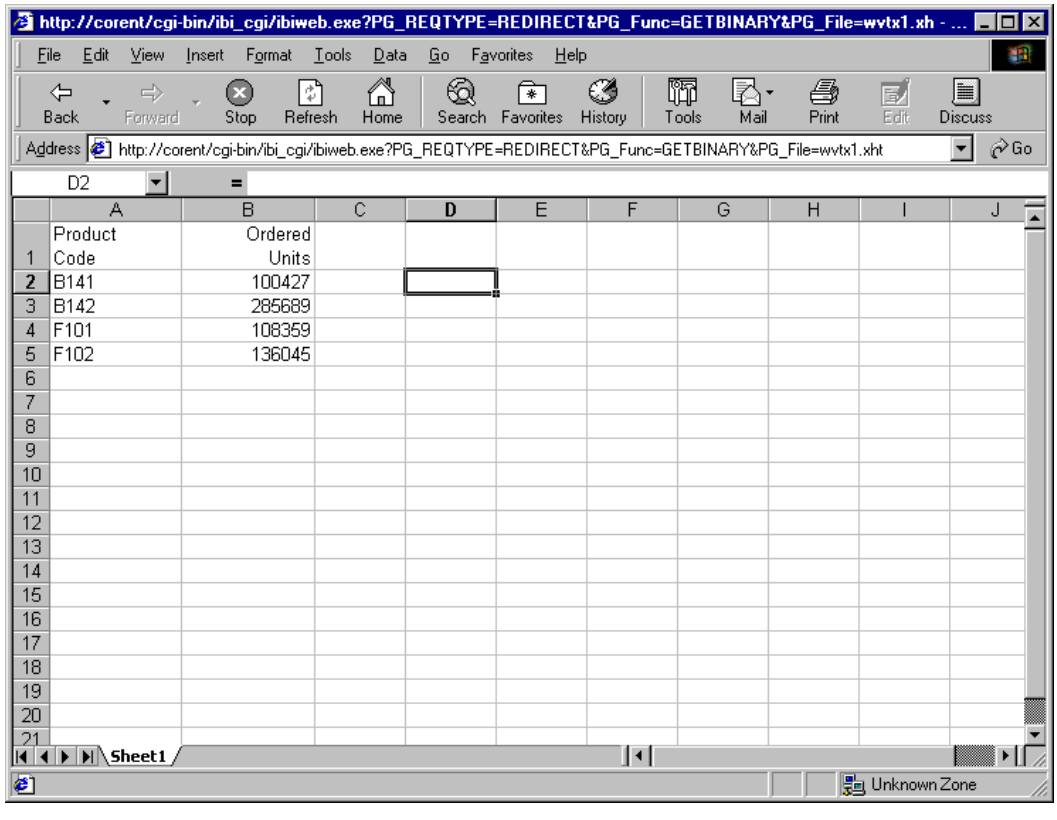
3. Deploy the procedure and launch page using the Deploy Wizard. See *Step 2. Create a Deployment Scenario* in Chapter 12, *Partitioning and Deploying Project Files* for instructions. When you run the launch page, WebFOCUS returns the data to the browser in the format specified. The browser displays it using the appropriate helper application.

## *Example* Displaying a Report in 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* in Chapter 12, *Partitioning and Deploying Project Files* 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:



## *Example* 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:

```
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
```

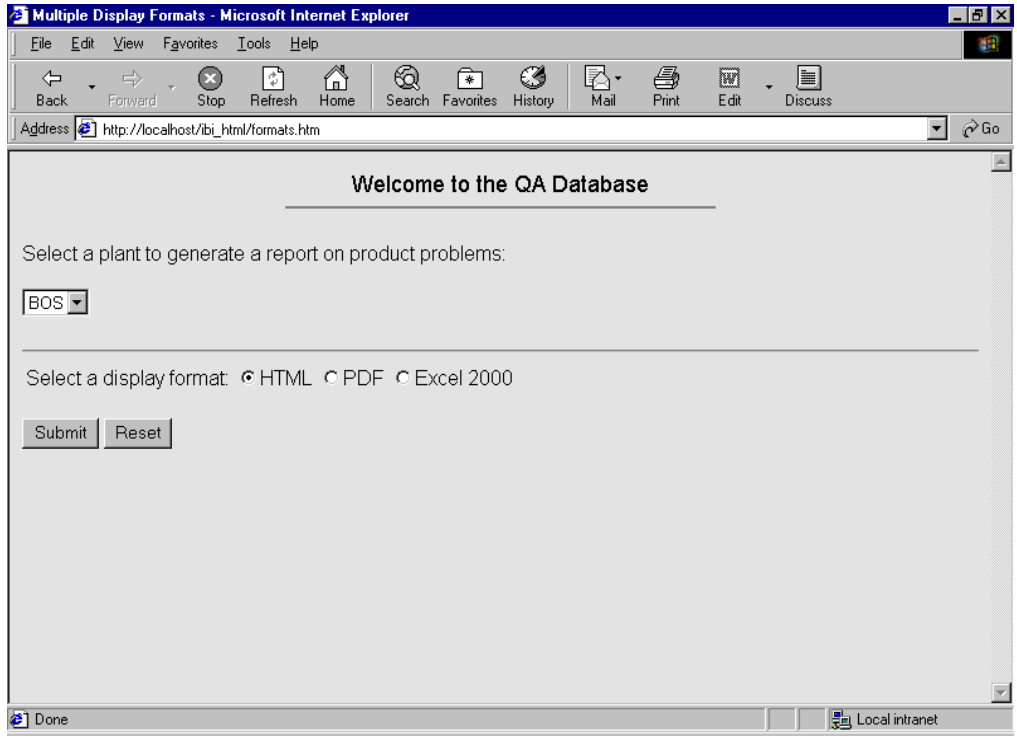
2. Create a launch page from which a user can run the report. The following sample launch page is named FORMATS. The letter on the left corresponds to the note explaining the code.

```
<HTML>
<HEAD>
<TITLE>Multiple Display Formats</TITLE>
</HEAD>
<BODY BGCOLOR="#E3E3E3">
<FONT FACE="Arial">
<CENTER>
<FONT SIZE="+1">Welcome to the QA Database</FONT>
<HR WIDTH="45%" NOSHADE>
</CENTER>
<FORM ACTION="/cgi-bin/ibi_cgi/ibiweb.exe" 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.

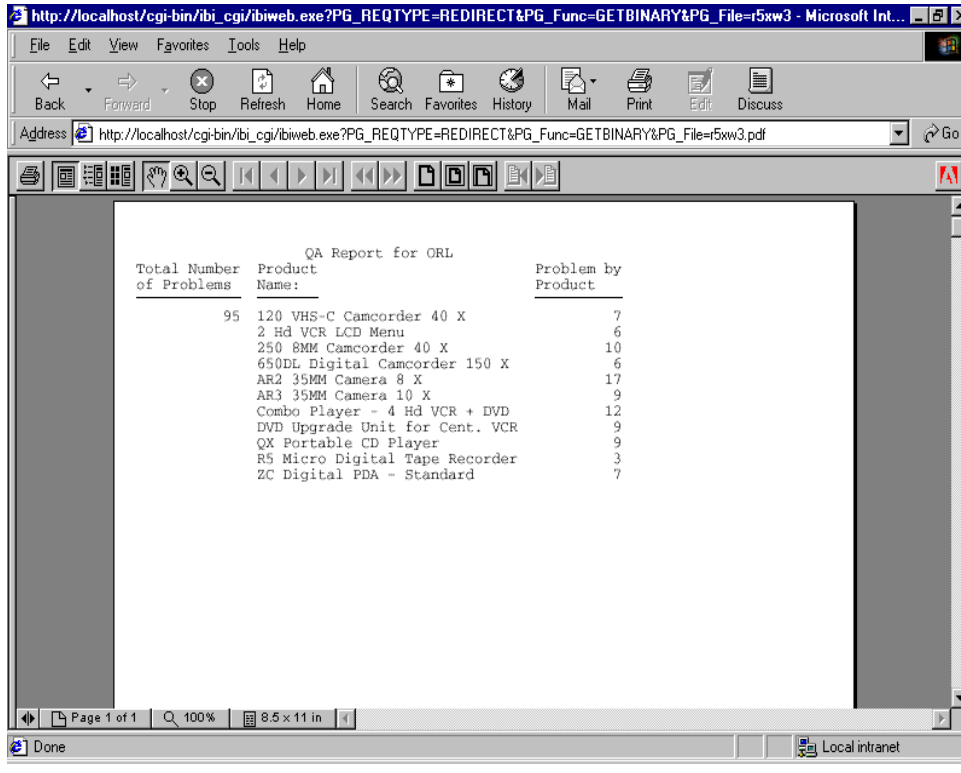
**3.** Deploy the procedure and launch page using the Deploy Wizard. See *Step 2. Create a Deployment Scenario* in Chapter 12, *Partitioning and Deploying Project Files*, for instructions.

**4.** Run the launch page:


Multiple Display Formats - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://localhost/ibi_html/formats.htm
Go
Welcome to the QA Database
Select a plant to generate a report on product problems:
BOS
Select a display format: HTML PDF Excel 2000
Submit Reset
Done
Local intranet

**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:




# Controlling Multiple Reports

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.

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 *Controlling the Execution of a Procedure* in Chapter 10, *Managing Flow of Control in an Application*, 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 Report Layout Painter. 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}
```

where:

`OFF`

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

`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

```
IBIWF_mprefix = {Report|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

```
IBIWF_morder = {FORWARD|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 MREPORT*n*, which may conflict with other HTML code.

The syntax is

```
IBIWF_mframename = {MREPORT|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

```
IBIWF_index = {ON|OFF}
```

where:

ON

Appends a sequence number of 1 (for the first report generated) to *n* (for the last reported generated). This value is the default.

OFF

Omits a sequence number. Only the text specified by IBIWF_mprefix applies.

## *Syntax* How to Control Multiple Reports From a Hyperlink

```
<A HREF="/alias/ibiweb.exe?IBIF_ex=procedure[&var=value[&var=value]...]">
text</A>
```

```
<A HREF="/alias/webapi.dll?IBIF_ex=procedure[&var=value[&var=value]...]">
text</A>
```

```
<A 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:

```
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 9-9 for a list of variables and valid values.

*text*

Is the text on the launch page that serves as the hyperlink that runs the procedure.

## *Example* 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 CGI 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>
<BODY>
<A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?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* in Chapter 12, *Partitioning and Deploying Project Files*, for instructions.

**4.** Run the launch page in the browser, and click *Run report* to receive the report on total dollar sales:

Sales Analysis Report 1
Sales Analysis Report 2

| Product | Dollar Sales |
|---|---|
| Biscotti | 5263317 |
| Capuccino | 2381590 |
| Coffee Grinder | 2337567 |
| Coffee Pot | 2449585 |
| Croissant | 7749902 |
| Espresso | 3906243 |
| Latte | 10943622 |
| Mug | 4522521 |
| Scone | 4216114 |
| Thermos | 2385829 |

Click *Sales Analysis Report 2* for the report on total unit and dollar sales:

Sales Analysis Report 1
Sales Analysis Report 2

| Product | Unit Sales | Dollar Sales |
|---|---|---|
| Biscotti | 421377 | 5263317 |
| Capuccino | 189217 | 2381590 |
| Coffee Grinder | 186534 | 2337567 |
| Coffee Pot | 190695 | 2449585 |
| Croissant | 630054 | 7749902 |
| Espresso | 308986 | 3906243 |
| Latte | 878063 | 10943622 |
| Mug | 360570 | 4522521 |
| Scone | 333414 | 4216114 |
| Thermos | 190081 | 2385829 |

## *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 9-9 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

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.

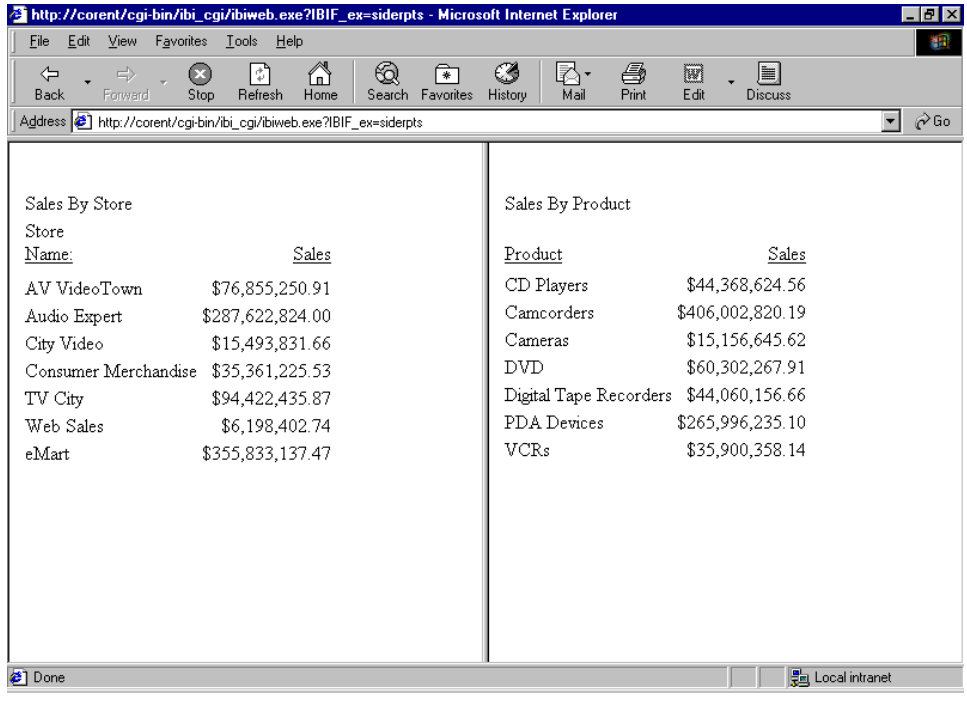
   -TYPE commands create a two-frame page on which the reports display side-by-side.

   **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* in Chapter 12, *Partitioning and Deploying Project Files* for instructions.
4. Access the launch page in the browser, and run the reports:



## *Example* 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.

   **Procedure:**

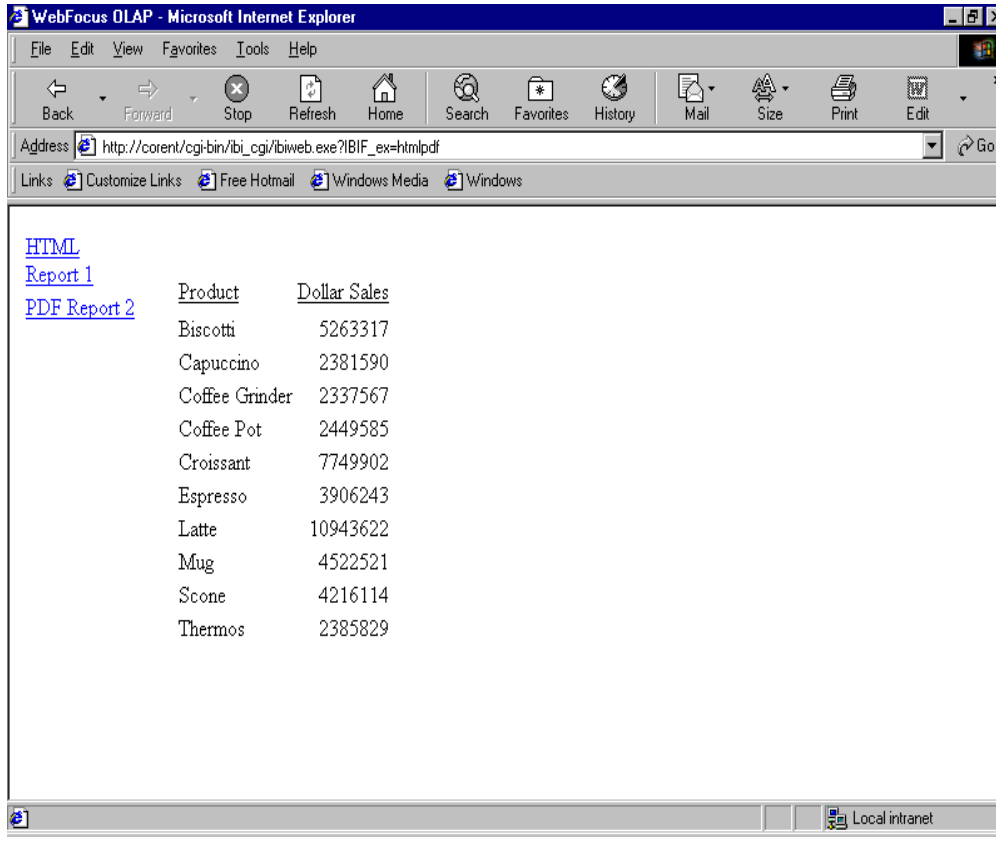
   ```
   -TYPE WEBFOCUS CGIVAR IBIWF_mreports=INDEX
   -TYPE WEBFOCUS CGIVAR IBIWF_mprefix=HTML Report
   TABLE FILE GGSALES
   SUM DOLLARS BY PRODUCT
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF,$
   ENDSTYLE
   END
   -RUN
   ```

   ```
   TABLE FILE GGSALES
   -TYPE WEBFOCUS CGIVAR IBIWF_mprefix=PDF Report
   SUM DOLLARS BY PRODUCT
   ON TABLE PCHOLD FORMAT PDF
   END
   ```

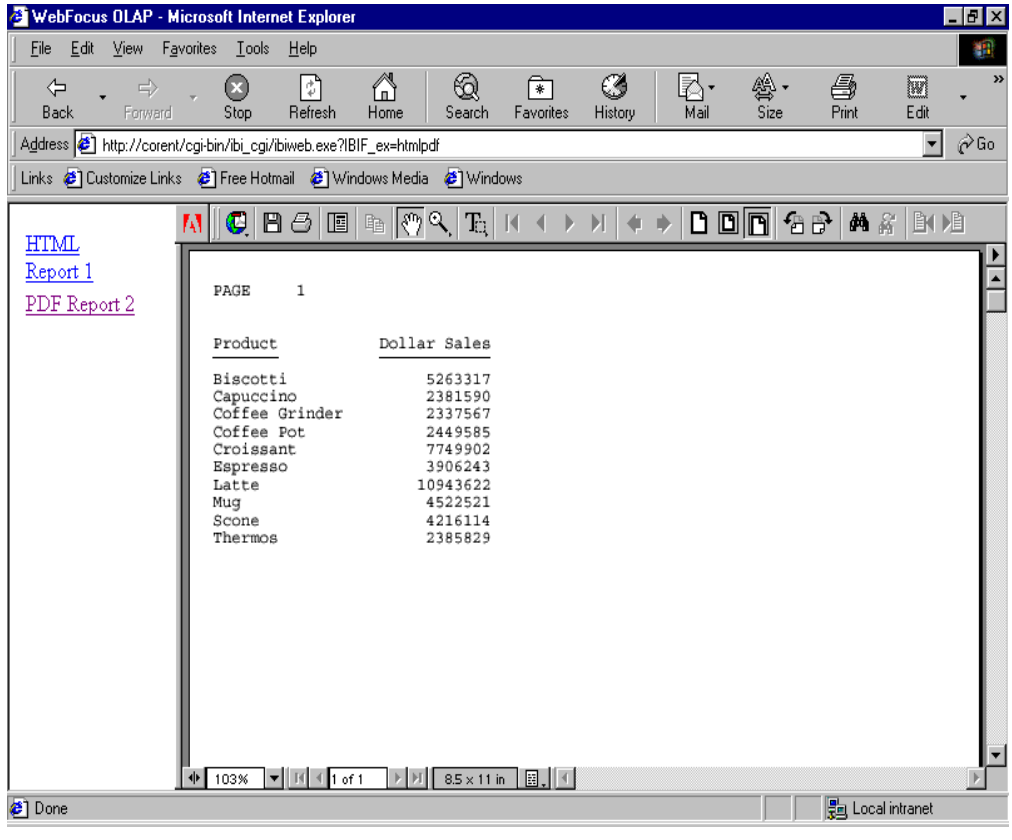
2. Create a launch page that runs the procedure.

3. Deploy the procedure and launch page using the Deploy Wizard. See *Step 2. Create a Deployment Scenario* in Chapter 12, *Partitioning and Deploying Project Files* for instructions.

**4.** Run the launch page in the browser. The report in HTML format displays:



**5.** Click *PDF Report 2* for the report in PDF using the Acrobat Reader:



## *Example* 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:**

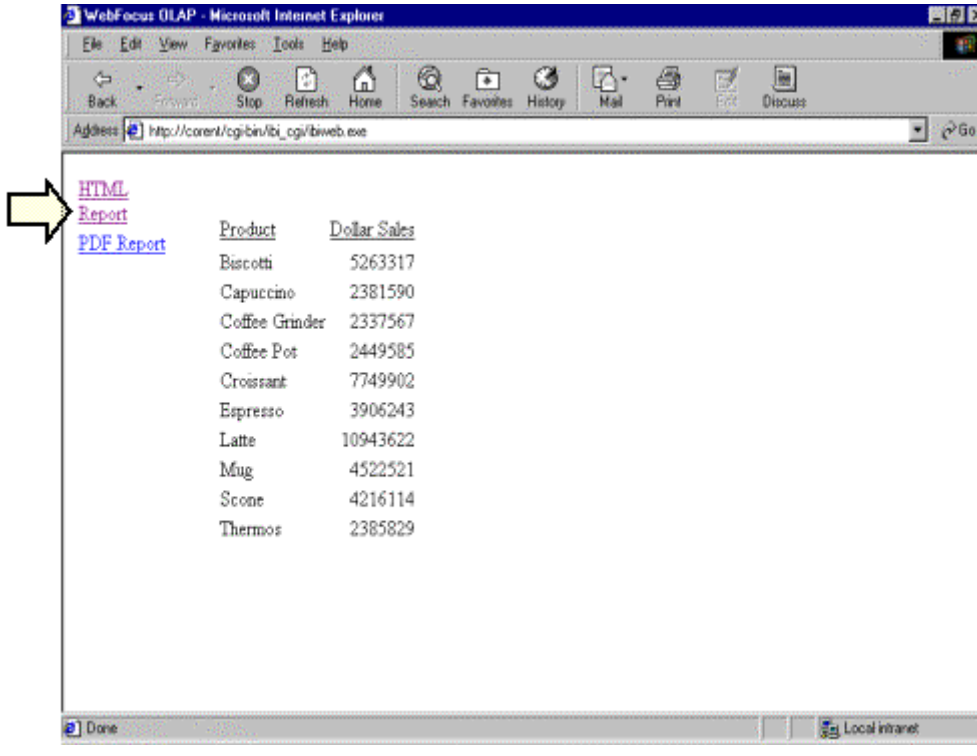
   ```
   -TYPE WEBFOCUS CGIVAR IBIWF_mreports=INDEX
   -TYPE WEBFOCUS CGIVAR IBIWF_index=OFF
   -TYPE WEBFOCUS CGIVAR IBIWF_mprefix=HTML Report
   TABLE FILE GGSALES
   SUM DOLLARS BY PRODUCT
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF,$
   ENDSTYLE
   END
   -RUN
   ```

   ```
   -TYPE WEBFOCUS CGIVAR IBIWF_mprefix=PDF Report
   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* in Chapter 12, *Partitioning and Deploying Project Files*for instructions.

**4.** Run the launch page in the browser. The reports have descriptive names without sequence numbers.

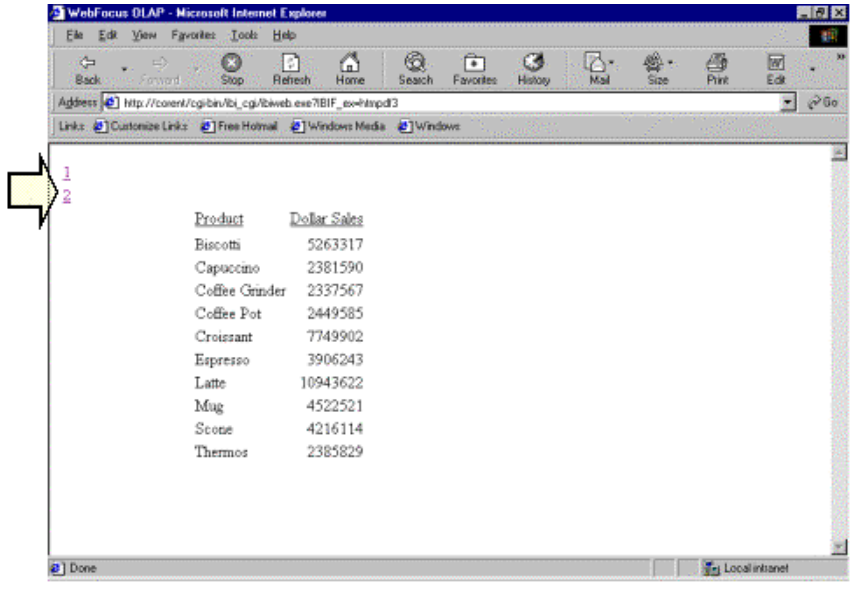


## *Example* Displaying Two Reports With Sequence Numbers Only

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,$
ENDSTYLE
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* in Chapter 12, *Partitioning and Deploying Project Files* for instructions.
4. Run the launch page in the browser. The reports have sequence numbers without descriptive names:



# Adding JavaScript for Drill-Down Reporting

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*** **How to Specify a Hyperlink to a JavaScript Function**

```
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=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'.

## *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.

```
   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>
```

**f.**
```
<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);
```

```
g.  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[11]=a11;
    arraygo[12]=a12;
    minval=100000;
    maxval=0;
    mnmmax=0;
    mnmmin=0;
    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[11]="November";
    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]; }
    </SCRIPT>
    <BODY>
    <FORM name="form1">
h.  <!--WEBFOCUS TABLE JAVATEMP>
    <INPUT TYPE=text NAME="text1" SIZE="10"> Units sold for store
    <INPUT TYPE=text NAME="store" SIZE="10"> Monthly average of
    <INPUT TYPE=text NAME="text2" SIZE="10"><BR><BR> Top selling
          month is
    <INPUT TYPE=text NAME="mnmmax" SIZE="10"> with
    <INPUT TYPE=text NAME="themax" SIZE="10">units<BR> Slowest
          month was
    <INPUT TYPE=text NAME="mnmmin" SIZE="10"> with
    <INPUT TYPE=text NAME="themin" SIZE="10">units<BR> Range
          between best and slowest months
    <INPUT TYPE=text NAME="range" SIZE="10">units </FORM>
    </BODY>
    </HTML>
i. -HTMLFORM END
   -EXIT
```

**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*

| Store Code | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R1019 | 3961 | 3119 | 3951 | 3983 | 4037 | 2888 | 4996 | 3943 | 4526 | 3962 | 3317 | 5305 |
| R1020 | 4152 | 3944 | 3927 | 4866 | 3697 | 3454 | 4449 | 4785 | 4461 | 4588 | 4454 | 4139 |
| R1040 | 4189 | 3766 | 4342 | 3741 | 4681 | 3953 | 3775 | 4655 | 3731 | 4127 | 3021 | 3794 |
| R1041 | 4650 | 4818 | 5061 | 3000 | 4528 | 4751 | 5396 | 4918 | 3443 | 4995 | 3873 | 5611 |
| R1044 | 3836 | 3599 | 3008 | 4356 | 4126 | 4414 | 4413 | 3038 | 4044 | 4109 | 4010 | 3493 |
| R1088 | 4039 | 5224 | 3700 | 3586 | 3435 | 3947 | 4083 | 3311 | 4889 | 3710 | 4382 | 4256 |
| R1100 | 4190 | 3652 | 4244 | 5146 | 4068 | 3848 | 4348 | 3760 | 4312 | 4151 | 3987 | 3963 |
| R1109 | 3716 | 4031 | 4350 | 3894 | 4396 | 4225 | 4221 | 4661 | 4421 | 3447 | 4161 | 4175 |
| R1200 | 3849 | 4604 | 3559 | 3657 | 4237 | 4688 | 4493 | 4663 | 3902 | 2995 | 3701 | 4303 |
| R1244 | 4169 | 3903 | 4014 | 4027 | 4049 | 4028 | 4111 | 4299 | 4061 | 4913 | 4730 | 3966 |
| R1248 | 3961 | 3283 | 4503 | 3854 | 5088 | 5324 | 4411 | 3658 | 3596 | 4586 | 4374 | 4358 |
| R1250 | 3195 | 4102 | 4492 | 4270 | 3579 | 4486 | 3781 | 3488 | 4233 | 4309 | 3759 | 3946 |

*Please click on the store code to summarize the store's data.*

Units sold for store Monthly average of

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:

47988 Units sold for store R1019 Monthly average of 3999

Top selling month is December with 5305 units
Slowest month was June with 2888 units
Range between best and slowest months 2417 units

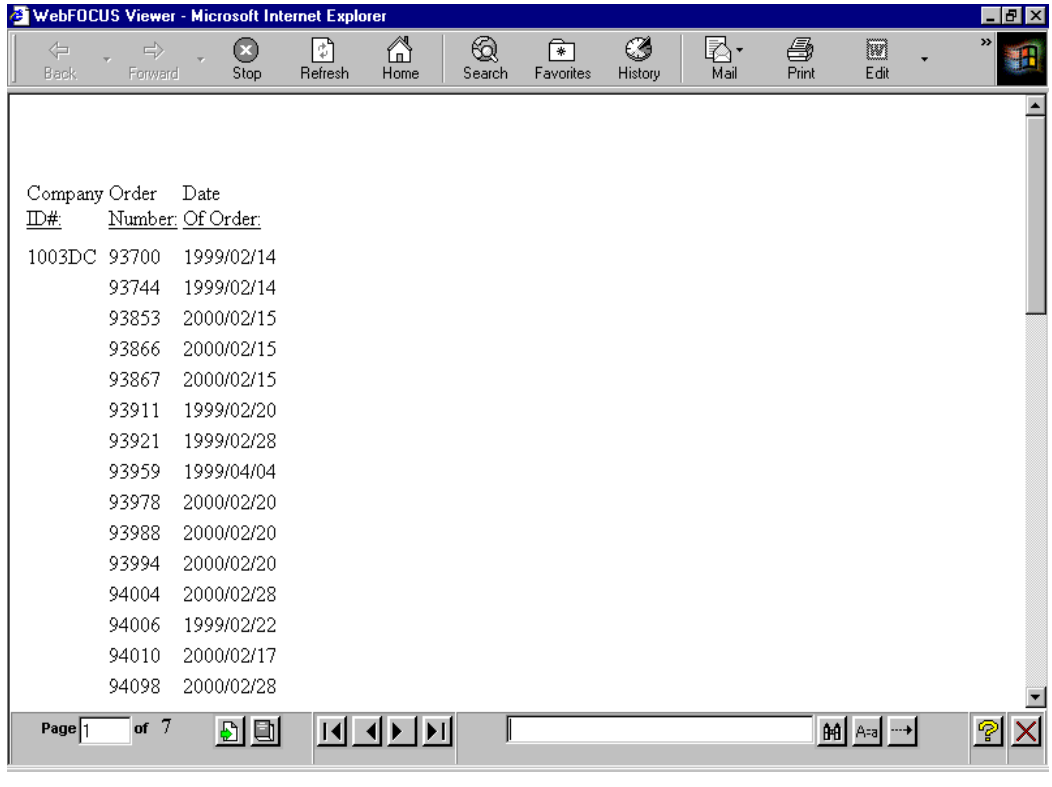
# Facilitating Report Manipulation

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:



## Using 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

```
SET WEBVIEWER = {OFF|ON}
```

or

```
ON TABLE SET WEBVIEWER {OFF|ON}
```

where:

OFF

Disables on-demand paging; WebFOCUS downloads the entire report to a standard browser window. This value is the default.

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.

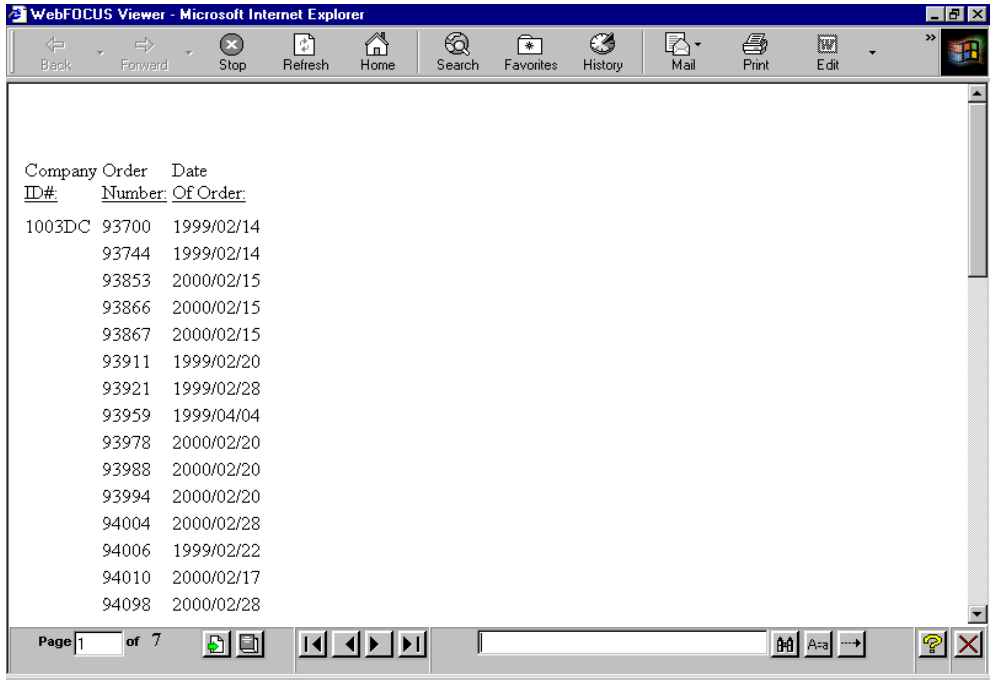
1. Create a procedure named ONDEMAND, which displays an order report for a store in the WebFOCUS Viewer.

   **Procedure:**

   ```
   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,$
   ENDSTYLE
   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

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 value 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. This value is the default.

### *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

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

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

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, $
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, $
b.  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 *Storing and Searching for Application Files* in Chapter 2, *WebFOCUS Application Logic*, for details on search paths.

   In Windows NT/2000/XP 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, which you create in step 3.

   **HTML Display Page: SCROLLPG.HTM**

```
<!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;
a.      top: 30;
        left: 60;
        width: 200;
        height: 200;
        border: thin outset;
b.      overflow: auto;
    }
    /* the following defines position and properties of top right
       graph */
    .graph1 { position: absolute;
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 {
d.    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 {
e.    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.

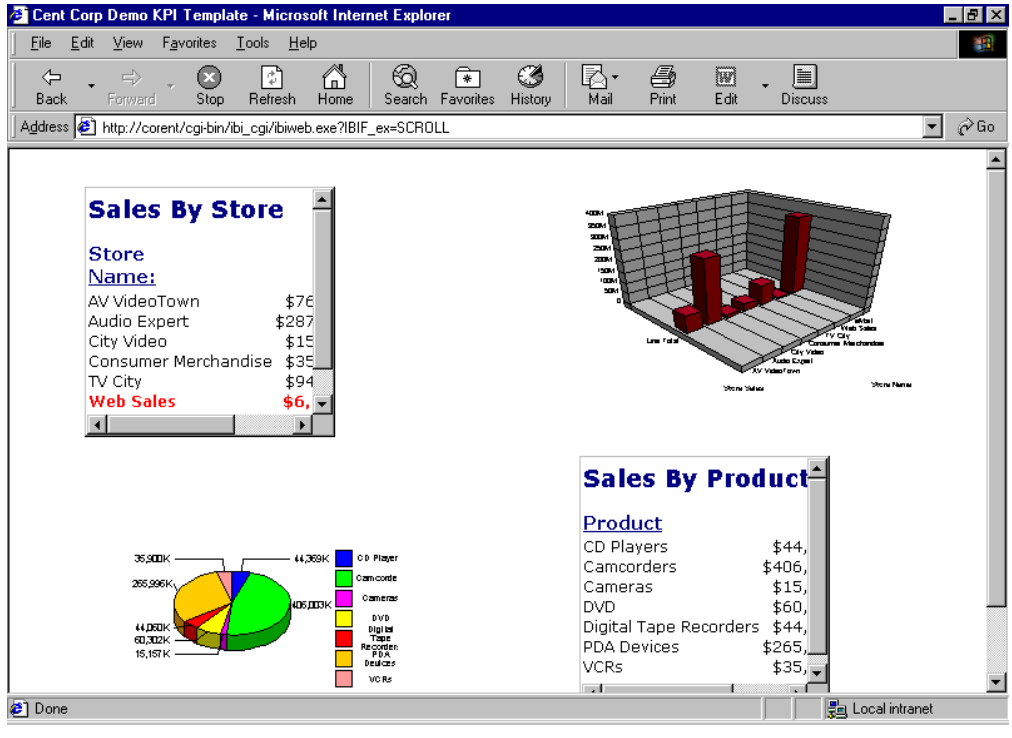
**b.** Create scroll bars if the report cannot be entirely displayed at one time in a frame.

**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:



# Displaying a Previously Run Report

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 NT/2000/XP:** *install_ drive*:\ibi\client52\conf\etc

  **UNIX:** /ibi/client52/conf/etc

  **OS/390:** /ibi/client52/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 example in this topic runs on WebFOCUS for Windows NT.

## *Procedure* How to Customize a Menu

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 NT file locations used in this example.

| Step | | File Name | File Location |
|---|---|---|---|
| **1.** | Create an HTML logon page. A cookie is created after successful logon, containing the user ID (IBIC_user) entered on the page. | SIGNON.HTM | WebFOCUS52\ibi_html |
| **2.** | Create a home page with a frameset that displays custom menu options and a report when run. | HOME.HTM | WebFOCUS52\ibi_html |
| **3.** | Create the frame pages. | MENU.HTM<br>LOADMEM.HTM<br>WELCOME.HTM | WebFOCUS52\ibi_html |
| **4.** | Modify the file CGIVARS.WFS to assign the value of the user ID to a Dialogue Manager amper variable. | CGIVARS.WFS | ibi\client52\conf\etc |
| **5.** | Create a user data source that contains user names and their allowed menu options. | USERLST.MAS<br>USERLST.DAT | srv52\ggdemo |

| Step | | File Name | File Location |
|---|---|---|---|
| **6.** | Create a menu data source that identifies and supplies a label for all available menu options, the location of the file accessed by an option, and the target frame for the output of an option. You can customize the contents of this file for your application. | MENULST.MAS<br>MENULST.DAT | srv52\ggdemo |
| **7.** | Create a procedure that generates a custom menu for a user. | FHSUB.FEX | srv52\ggdemo |
| **8.** | Create an HTML display page on which the custom menu is displayed. | SUBMENU.HTM | srv52\ggdemo |
| **9.** | Create the forms that call the report selections on the menu. | 1MYGRAPH.THM<br>2MYGRAP.HTM | WebFOCUS52\ibi_html |
| **10.** | Create the procedures that generate the user-specific reports. | FHSUB.FEX | srv52\ggdemo |

**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.

## *Example* Step 1: Creating the HTML Logon Page

1. The following file is named SIGNON.HTM. Modify the lines in bold to apply to the Web server at your site.

   **SIGNON.HTM:**

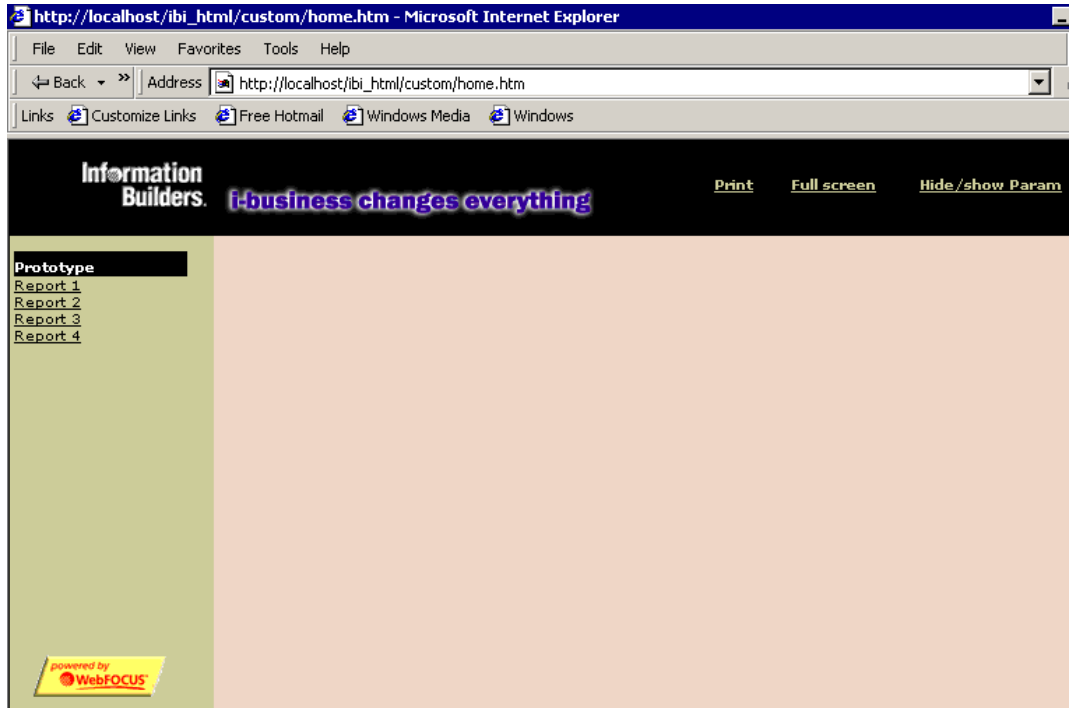
```
<html>
<head>
<meta http-equiv="Content-Language" content="nl-be">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Please Identify yourself!</title>
</head>
<body stylesrc="submenu.htm" bgcolor="#CECF9C">
<p> </p>
<form method="POST" action="/cgi-bin/ibi_cgi/ibiweb.exe"
    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>
        <p align="center"> </td>
    </tr>
    <tr>
      <td width="66%" height="19" align="right" colspan="2">
       <p align="center"><b><u><font face="Verdana" size="3">Log on to
          WebFOCUS reporting</font></u></b></p>
        <p align="center"> </td>
    </tr>
    <tr>
      <td width="31%" height="19" align="right"><font face="Verdana"
        size="2"><b>User ID</b></font></td>
      <td width="35%" height="19"><input type="text" name="IBIC_user"
        size="20"></td>
    </tr>
    <tr>
      <td width="31%" height="18" align="right"><font face="Verdana"
        size="2"><b>Password</b></font></td>
      <td width="35%" height="18"><input type="password"
        name="IBIC_pass" size="20"></td>
    </tr>
    <tr>
```

```
      <td width="31%" height="57">
        <p align="right"><input type="reset" value="Reset"
          name="B2"></td>
      <td width="35%" height="57"><input type="submit" value="Submit"
        name="B1"></td>
    </tr>
  </table>

  <p> </p>
  <p> </p>
  <p> </p>
  <p> </p>
  <p> </p>
  <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">
 </form>
<p> </p>
</body>
</html>
```

**2.** The following is the SIGNON.HTM page when accessed in the Web browser:

## *Example* Step 2: Creating the Home Page

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).



1. The following file is named HOME.HTM. Modify the lines in bold to apply to the Web server at your site.

   **HOME.HTM:**

```
<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>
 <noframes>
  <body>
  <p>This page uses frames, but your browser doesn't support them.</p>
  </body>
 </noframes>
</frameset>
</html>
```

## *Example* Step 3: Creating the Frame Pages

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.

   **MENU.HTM:**

```
<HTML>
<HEAD>
<meta base href="http://localhost/ibi_html" target="contents">
<TITLE>The JavaScript Menu Object</TITLE>
<script language="Javascript"><!---
function doprint()
{
 parent.document.frames.bodypart.all("Reportframe").focus();
 window.print();
}
function Show_query()
{
 if (parent.frames('QueryFrame').document.title == 'Query')
 {
  x=parent.frames('QueryFrame').document.all('Query').offsetHeight-10;
  parent.document.all("header").all("bodypart").all("QueryPart")
  .rows=x+",*";
 }
}

function Hide_A_query()
{
 if (parent.document.all("header").all("bodypart").all("QueryPart")
  .rows =='0,*')
 {
  resize()
 }
 else
 {
  parent.document.all("header").all("bodypart").all("QueryPart")
  .rows='0,*';
 }
}

function resize(){
var
x=parent.frames('QueryFrame').document.all('Query').offsetHeight-10;
parent.document.all("header").all("bodypart").all("QueryPart")
.rows=x+",*";
}
```

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])
    {
    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 = /^(\d{1,2})(\/|-)(\d{1,2})\2(\d{2}|\d{4})$/;
    // To require a 4 digit year entry, use this line instead:
    // var datePat = /^(\d{1,2})(\/|-)(\d{1,2})\2(\d{4})$/;

    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;
     }
    }
      }
    }
  }
```

```
 }
  return boodschap
}

function breakout() {
  mywindow=window.open("","Titel","scrollbars=yes,status=no");
  mywindow.location.href=parent.frames('ReportFrame').location.href;
}
//-->
</script>
<style fprolloverstyle>A:hover {color: #FFFF00; font-family: Verdana;
 font-size:8pt; font-weight: bold}
</style>
<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> </td>

    <td width="57" align="center" height="1">
      <p align="center"><font color="#CECF9C" size="1">
        <b>
        <a href='javascript:void 0' onclick='doprint();'>Print</a>
        </b></font>
    </td>

    <td width="86" height="1" align="center">
      <p align="center"><font color="#CECF9C" size="1">
      <b><a href='javascript:void 0' onclick='breakout();'>
         Full screen</a></b></font>
    </td>

    <td width="138" align="center" height="1">
      <p align="center"><font color="#CECF9C" size="1">
      <b><a href='javascript:void 0'onclick='Hide_A_query();'>
           Hide/show Param</a></b></font>
    </td>

</tr>
</table>
</BODY>
```

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>
<head>
<title>Welcome text</title>
<SCRIPT LANGUAGE="JavaScript">
<!--
function loadmenu() {
IBI_random=Math.random()*Math.random()*Math.random()*100 ;
loadurl="http://localhost/cgi-bin/ibi_cgi/
ibiweb.exe?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 ..</font></p>
</SCRIPT>
</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>
```

## *Example* Step 4: Modifying the SITE.WFS Files

1. Open the file SITE.WFS, located by default as follows:

   **Windows NT/2000/XP:** *drive:*\ibi\client52\conf\etc

   **UNIX:** /ibi/client52/conf/etc

   **OS/390:** /ibi/client52/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
   <endsendvar>
   ```

   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.

## *Example* Step 5: Creating the User Data Source

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 *Storing and Searching for Application Files* in Chapter 2, *WebFOCUS Application Logic*.

```
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.

## *Example* Step 6: Creating the Menu Data Source

The following Master File for the menu data source is named MENULST.MAS:

```
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
```

## *Example* Step 7: Creating the Procedure

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
-* &LSTDIR  : Location of rlist.htm
-*********************************************************************
-*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
```

## *Example* Step 8: Creating the HTML Display Page

The HTML display page is called by the procedure, and must be accessible by the WebFOCUS Reporting Server. In Windows NT/2000/XP and UNIX, the extension is always .HTM.

1. 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>
<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()">

<table border="0" width="100%" cellspacing="0" cellpadding="0">
  <tr>
    <td width="100%" bgcolor="#000000">
      <p align="left"><b><font size="1" color="#FFFFFF">Prototype
        </font></b>
     </td>
  </tr>
!IBI.FIL.RLIST;
  <tr>
    <td width="100%"><font size="1" color="#CECF9C">.</font></td>
  </tr>
</table>


<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p align="center"><img border="0" src="http://localhost/ibi_html/
  template/powered.gif" align="baseline"></p>
</body>
</html>
```

## *Example* Step 9: Creating the Forms That Call the Report Selections on the Menu

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.

   **1MYQRAP.HTM:**

```
<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(){
  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="/cgi-bin/ibi_cgi/ibiweb.exe"
  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"><u><b>
        Please provide Parameters for Report 1 </b></u></font>
      </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">
        <font face="Verdana" size="1">
        <select size="1" name="AMP_VAR1" style="font-family:Verdana;
           font-size: 8pt">
```

```
          <option value="ACTION">ACTION</option>
          <option value="MYSTERY">MYSTERY</option>
          <option value="CLASSIC">CLASSIC</option>
          <option value="CHILDREN">CHILDREN</option>
          <option value="MUSICALS">MUSICALS</option>
          </select></font></td>
      <tr>
        <td width="70" align="right" bgcolor="#D8D9B0" height="1">

        <p></p>
        </td>
        <td width="104" bgcolor="#D8D9B0" height="27">
           <font face="Verdana" size="1">
           <input type="submit" value="Run" name="B1"></font>
        </td>
      </tr>
    </table>
  </form>
  </DIV>
  </body>
  </html>
```

**2.** The following file is named 2MYQRAP.HTM. No modification is required.

**2MYQRAP.HTM:**

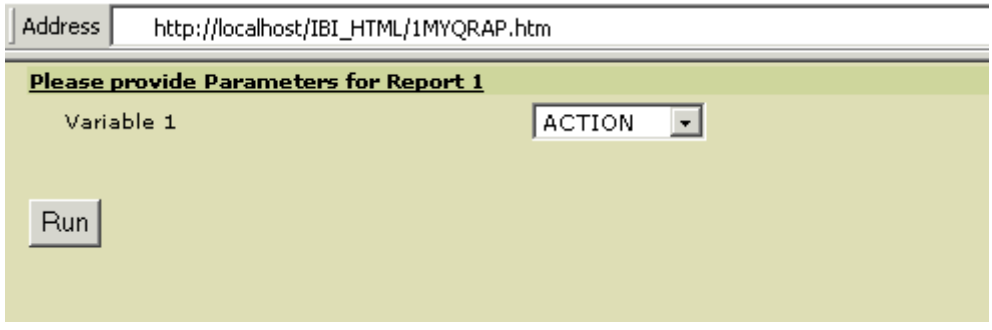
```
<html>
<head>
<title>Query</title>
<meta name="Microsoft Border" content>
</head>

<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()
{
  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));
```

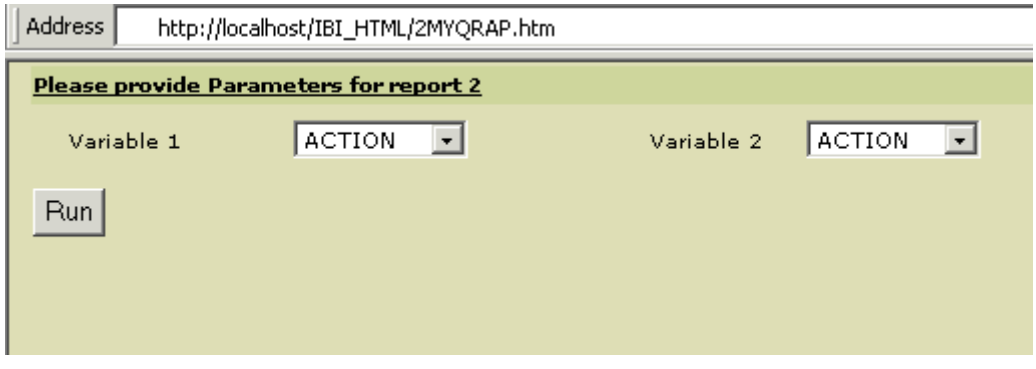
```
}
//--></script>
<form method="GET" action="/cgi-bin/ibi_cgi/ibiweb.exe"
  style="font-weight: bold" name="frminput"
  onsubmit="return Form1_Validator(this)" target="ReportFrame">
  <input type="hidden" name="IBIF_ex" value="REPORT2">
  <input type="hidden" name="IBI_random" value="">

<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>
      </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">
         <font face="Verdana" size="1">
         <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></font></td>

      <td width="82" align="right" bgcolor="#D8D9B0" height="1">
      <font face="Verdana" size="1">Variable 2</font>
      <td width="212" bgcolor="#D8D9B0" height="25">
          <font face="Verdana" size="1">
      <select size="1" name="AMP_VAR2" 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></font></td>
    </tr>

  <tr>
```

```
        <td width="70" align="right" bgcolor="#D8D9B0" height="1">
           <p></p>
         </td>
         <td width="104" bgcolor="#D8D9B0" height="27">
             <font face="Verdana" size="1">
           <input type="submit" value="Run" name="B1"></font></td>
     </tr>
    </table>
  </form>
  </DIV>
  </body>
  </html>
```

The following is the 1MYQRAP.HTM page when accessed in the Web browser:




The following is the 2MYQRAP.HTM page when accessed in the Web browser:



## *Example* Step 10: Creating the Procedures

1. 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>
<UL>
<LI> CATEGORY: &AMP_VAR1
<LI> CURRENT DATE : &DATEDMYY
<LI> CURRENT TIME : &TOD
<LI> CURRENT USER : &RUSER
</UL>
</BODY>
</HTML>
-HTMLFORM END
```

CHAPTER 10

# Managing Flow of Control in an Application

**Topics:**

- Uses for Dialogue Manager
- Dialogue Manager Processing
- Creating a Dialogue Manager Procedure
- Customizing a Procedure With Variables
- Creating and Working With Local and Global Variables
- Controlling the Execution of a Procedure
- Navigating a Procedure
- Enhancing an HTML Web Page With a Procedure
- Issuing Operating System Commands
- Controlling Passwords With Dialogue Manager
- Sending a Message to the Application
- Testing and Debugging a Dialogue Manager Procedure
- Dialogue Manager Syntax Reference

Dialogue Manager is the part of the WebFOCUS language that controls the execution of your application's components. You can dynamically control the execution of procedures, giving you flexibility in application design. Dialogue Manager also enables you to use variables in your procedures and supply values for them at run time.

This topic contains information for the following operating systems: Windows NT/2000/XP, OS/390, and UNIX. References to Windows NT/2000/XP apply to Developer Studio and WebFOCUS. References to all other operating systems apply only to WebFOCUS.

When you open a previously written procedure in Developer Studio, Dialogue Manager commands *within* TABLE commands will be eliminated. However, Dialogue Manager commands used before or after TABLE and END commands will be retained.

# Uses for Dialogue Manager

Dialogue Manager is a part of the WebFOCUS language that allows you to control the flow of your application with the use of variables. The following are some of the ways you can use Dialogue Manager:

- **Customizing a procedure with variables.** You can dynamically change a procedure's execution by including variables, or retrieve information from the system. For details, see *Customizing a Procedure With Variables* on page 10-9. For information on creating the variables used in a procedure, see *Creating and Working With Local and Global Variables* on page 10-31.
- **Controlling the execution of a procedure.** You can control a procedure by executing stacked commands, closing external files, and exiting or canceling a procedure. For details, see *Controlling the Execution of a Procedure* on page 10-41.
- **Navigating a procedure.** You can conditionally execute requests, repeat execution with program loops, or call another procedure. For details, see *Navigating a Procedure* on page 10-44.
- **Enhance an HTML Web page.** You can increase the functionality of your Web page by including HTML commands in your procedure. For details, see *Enhancing an HTML Web Page With a Procedure* on page 10-58.
- **Issuing operating system commands**. You can issue an operating system command to set up an environment in which a request must run. For details on issuing operating system commands, see *Issuing Operating System Commands* on page 10-66.
- **Controlling passwords.** You can directly assign and change passwords. For details, see *Controlling Passwords With Dialogue Manager* on page 10-67.
- **Sending a message to the application.** You can send a message to an application while a procedure is processing to explain the purpose of the procedure, display results, or present useful information. For details, see *Sending a Message to the Application* on page 10-68.

## *Reference* Summary of Dialogue Manager Commands

The following table summarizes commands available in Dialogue Manager. For detailed information about these commands, see the individual topics for each command.

| Command | Description |
|---|---|
| `-*` | Signals a comment. |
| `-?` | Displays the value of local variables. |
| `-CLOSE` | Closes an external file on OS/390 opened for reading or writing. An external file is a sequential file in the operating system's file system. |
| `-DEFAULT` | Sets a variable to an initial value. |
| `-DOS` | Executes a DOS operating system command. |
| `-EXIT` | Executes stacked commands and terminates the procedure. |
| `-GOTO` | Transfers control to a label. |
| `-HTMLFORM` | Sends one or more reports to the browser. |
| `-IF` | Determines the execution flow of a procedure based on the evaluation of an expression. |
| `-INCLUDE` | Calls another Dialogue Manager procedure. |
| `-label` | Is the target of a -GOTO or -IF. |
| `-MVS` | In WebFOCUS, executes an OS/390 operating system command. |
| `-PASS` | Issues and controls passwords. |
| `-QUIT` | Terminates the procedure without executing stacked commands. |
| `-QUIT FOCUS` | In Developer Studio, terminates a procedure and exits FOCUS. |
| `-READ` | Reads data from an external file. |
| `-REPEAT` | Executes a loop. |
| `-RUN` | Executes stacked commands. |
| `-SET` | Sets a variable to a literal value or to a value computed in an expression. |
| `-TSO` | In WebFOCUS, executes a TSO operating system command. |

| Command | Description |
|---|---|
| -TYPE | Sends a message to a client application. |
| -UNIX | In WebFOCUS, executes a UNIX operating system command. |
| -VMS | In WebFOCUS, executes a VMS operating system command. |
| -WINNT | Executes a Windows NT/2000/XP operating system command. |
| -WRITE | Writes data to an external file. |

# Dialogue Manager Processing

You can modify your application at run time with user input and environment conditions by using Dialogue Manager procedures, which include commands and variables. It is important to understand how Dialogue Manager processes an application's commands and variables, using a procedure.

Dialogue Manager executes a procedure as follows:

1. Dialogue Manager reads each line of the procedure one by one, and substitutes values for variables when encountered.

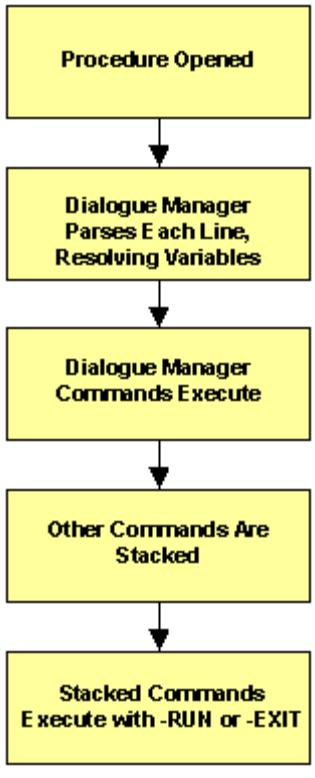
   You must make sure to assign values or identify defaults for all variables. Otherwise, you will receive an error message indicating a missing value. In Developer Studio, you are prompted for values not already assigned.

2. All Dialogue Manager commands execute as soon as Dialogue Manager reads them.

   Other WebFOCUS commands are temporarily stored for subsequent execution; they are called *stacked* commands.

3. The -RUN and -EXIT commands execute any stacked commands.

The following diagram illustrates how a Dialogue Manager procedure is processed:



## *Example* A Procedure and Its Execution Process

The following is an example of a procedure, with an explanation of the way it processes.

The procedure contains the variable &PLANT. Assume that when the procedure executes it includes the variable value of BOS for &PLANT. The numbers on the left correspond to the notes explaining the code.

```
1. -IF &PLANT EQ 'DONE' THEN GOTO GETOUT;

2. TABLE FILE CENTHR
   HEADING
    "SALARY REPORT FOR LOCATION: &PLANT"
   PRINT SALARY POSITION
   BY LNAME
   WHERE PLANT IS '&PLANT'
   END

3. -RUN

4. -EXIT

   -GETOUT
   -TYPE NO PROCESSING DONE: EXITING
```

The procedure processes as follows:

1. The values for &PLANT is passed to the procedure before the first line executes. This can be done by entering values in the HTML interface, or by setting the values previously with other Dialogue Manager commands. Dialogue Manager substitutes the value BOS for the variable &PLANT in the first line, and tests for the value DONE. The test fails, so Dialogue Manager proceeds to the next line.

   If the value were DONE instead of BOS, control would pass to the label -GETOUT, and the message NO PROCESSING DONE: EXITING would be generated. Dialogue Manager would skip the report request.

2. The next six lines are the report request. Dialogue Manager scans them for the presence of variables, substituting BOS for &PLANT. As each line is processed, it is placed on a stack to be executed later.

After Dialogue Manager processes the END command, the stacked commands look like this:

```
TABLE FILE CENTHR
HEADING
"SALARY REPORT FOR LOCATION: BOS"
PRINT SALARY POSITION
BY LNAME
WHERE PLANT IS 'BOS'
END
```

The next line is then processed by Dialogue Manager.

To display populated variable values like those contained in this step, include the &ECHO variable in your procedure. For information on the &ECHO variable, see *Testing and Debugging a Dialogue Manager Procedure* on page 10-69.

3. -RUN causes the stacked commands to be executed. They are sent to be processed.
4. -EXIT terminates the procedure.

For complete information on using variables in procedures, see *Customizing a Procedure With Variables* on page 10-9.

## Creating a Dialogue Manager Procedure

You can create a procedure with a system editor. When creating a procedure, you must follow these rules:

- Dialogue Manager commands begin with a hyphen. Other commands do not.
- At least one space must be inserted between the Dialogue Manager command and other text.

If a Dialogue Manager command exceeds one line, the following line must begin with a hyphen (-) in the first position. The continuation line can begin immediately after the hyphen, or you may insert a space between the hyphen and the rest of the line.

## Commenting a Procedure

It is good practice to include comments in a procedure for the benefit of others who may use it. It is particularly recommended that you use comments in a procedure heading to supply the date, the version, and other relevant information. A hyphen and an asterisk (-*) mark the beginning of a comment.

### *Reference* Adding a Comment in a Procedure

You can place a comment in a procedure in the following ways:

- Include text after the -*, optionally with a space before the text.
- Place a comment at the beginning or end of a procedure, or in between commands. A comment cannot be on the same line as a command. The following is invalid:

  ```
  -RUN -*Version 2 06/10/00
  ```

- In Developer Studio, if you are creating a procedure with graphical tools such as Report Painter, Report Assistant, or Graph Assistant, use the *Other* tool to enter comments. Comments added directly in a report or graph procedure using the text editor will be lost when you open the procedure in the graphical tool if the comments occur between the TABLE and END commands. See Chapter 18, *Editing Application Components as Text in Developer Studio* for details on the Other tool.

### *Example* Placing a Comment in a Procedure

The following example places comments at the beginning of a procedure.

```
-* Version 1  08/26/02  HRINFO  Procedure
TABLE FILE CENTHR
    .
    .
    .
```

# Customizing a Procedure With Variables

You can customize a procedure and control its execution with variables. This is done dynamically by supplying a variable's value at run time. You can supply a variable's value in the following ways:

- Directly in a procedure. The commands -DEFAULT, -SET, and -READ enable you to supply values directly in a procedure.
- When calling a procedure from another procedure. You can include the variable names and their corresponding values as parameters in an EXEC command that calls one procedure from another.
- When calling a procedure from a Web page. The Web page calls the WebFOCUS script from either a form or a hyperlink. If you use a form, the end user enters the values in the form. If you use a hyperlink, you include the name of the procedure and the variable values directly in the HTML <A HREF code. The value for each variable is included in the call to the WebFOCUS script.
- In a drill-down report. You can use a WebFOCUS style sheet to pass parameters.
- From an external file. You can supply values from an external file with the -READ command.

For information on the types of variables and how to use them, see *Creating and Working With Local and Global Variables* on page 10-31.

*Reference*

## Rules for Supplying Variable Values

The following rules apply to values for variables:

- The maximum length of a variable value is 32,000 characters when using the -TYPE or -WRITE command. When used with other Dialogue Manager commands, a variable cannot be longer than 4,096 characters.
- If a value contains an embedded comma, equal sign, or blank, you must enclose the variable name in single quotation marks when you use it in an expression. For example, if the value for &CITY is NY, NY, you must refer to the variable as '&CITY' in any expression.
- Once a value is supplied for a local or global variable, it is used throughout the procedure unless it is changed with a command such as -SET or -READ.
- In Developer Studio, you can supply a variable by selecting *Run...* from the *File* menu.
- You can populate a global variable each time a procedure is executed by including the variable in a WebFOCUS Reporting Server profile, such as EDASPROF.PRF.

## *Example* Customizing a Procedure With Variable Values

This example illustrates the use of the -DEFAULT and -SET commands to supply values for variables. The end user supplies the value B10 for &CODE1, B20 for &CODE2, and SMITH for &REGIONMGR as prompted by an HTML form.

**Note:** In this example, the output was formatted to eliminate page numbering and grids. For information on formatting your output, see the *Creating Reports With WebFOCUS Language* manual.

The numbers to the left of the example apply to the notes that follow:

```
1. -DEFAULT &VERB=SUM
2. -SET &CITY=IF &CODE1 GT 'B09' THEN 'STAMFORD' ELSE 'UNIONDALE';
3. -TYPE REGIONAL MANAGER FOR &CITY
   SET PAGE=OFF
4.     TABLE FILE SALES
       HEADING CENTER
       "MONTHLY REPORT FOR &CITY"
       "PRODUCT CODES FROM &CODE1 TO &CODE2"
       " "
       &VERB UNIT_SOLD AND RETURNS AND COMPUTE
       RATIO/D5.1 = 100 * (RETURNS/UNIT_SOLD);
       BY PROD_CODE
       IF PROD_CODE IS-FROM &CODE1 TO &CODE2
       FOOTING CENTER
5.     "REGION MANAGER: &REGIONMGR"
       "CALCULATED AS OF &DATE"
       ON TABLE SET STYLE *
       TYPE=REPORT, GRID=OFF,$
       END
6. -RUN
```

The procedure executes as follows:

1. The -DEFAULT command sets the value of &VERB to SUM.
2. The -SET command supplies the value for &CITY depending on the value the end user entered in the form for &CODE1. Because the end user entered B10 as the value for &CODE1, &CITY becomes STAMFORD.
3. When the end user runs the report, WebFOCUS writes a message that incorporates the value for &CITY:

   ```
   REGIONAL MANAGER FOR STAMFORD
   ```

   The output from -TYPE is not displayed on the HTML page; instead, it is written to the HTML source file. To display the message, the end user must view the document's source file.

**4.** The stack contains the following lines:

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR STAMFORD"
"PRODUCT CODES FROM B10 TO B20"
" "
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.1 = 100 * (RETURNS/UNIT_SOLD);
BY PROD_CODE
IF PROD_CODE IS-FROM B10 TO B20
FOOTING CENTER
"REGION MANAGER: SMITH"
"CALCULATED AS OF 07/08/93"
END
```

**5.** The end user supplied the value for &REGIONMGR on the form. WebFOCUS supplies the current date at run time.

**6.** The -RUN command causes execution of all commands in the stack. The output from the report request is as follows:

MONTHLY REPORT FOR STAMFORD

PRODUCT CODES FROM B10 TO B20

| PROD_CODE | UNIT_SOLD | RETURNS | RATIO |
|---|---|---|---|
| B10 | 103 | 13 | 12.6 |
| B12 | 69 | 4 | 5.8 |
| B17 | 49 | 4 | 8.2 |
| B20 | 40 | 1 | 2.5 |

REGION MANAGER: SMITH

CALCULATED AS OF 07/08/93

## Supplying a Default Variable Value

You can supply a default variable with the -DEFAULT command. It ensures that a value is passed to a variable.

You can override default values by supplying new values when you call the procedure, or by issuing -SET after the -DEFAULT command.

### *Syntax* How to Supply a Default Value

```
-DEFAULT &[&]name=value [&[&]name=value]
```

where:

*&name*

Is the name of the variable.

*value*

Is the default value assigned to the variable.

### *Example* Supplying a Default Value

In the following example, -DEFAULT sets a default value for &PLANT:

```
-DEFAULT &PLANT=BOS
TABLE FILE CENTHR
      .
      .
     .
```

## Supplying Variable Values in an Expression

You can assign a variable's value by computing the value in an expression or assigning a literal value to a variable with the -SET command.

### *Syntax* How to Assign a Value in an Expression

```
-SET &[&]name= {expression|value};
```

where:

*&name*

Is the name of the variable.

*expression*

Is a valid expression. Expressions can occupy several lines, so you should end the command with a semicolon.

*value*

Is a literal value, or arithmetic or logical expression assigned to the variable. If the literal value contains commas or embedded blanks, you must enclose the value in single quotation marks. If the value contains a single quote, place two single quotes where you want one to appear.

### *Example* Setting a Variable Value in an Expression

In the following example, -SET assigns the value 14Z or 14B to the variable &STORECODE, as determined by the logical IF expression. The value of &CODE is supplied by the end user.

```
-SET &STORECODE = IF &CODE GT C2 THEN '14Z' ELSE '14B';
   TABLE FILE SALES
   SUM UNIT_SOLD AND RETURNS
   BY PROD_CODE
   IF PROD_CODE GE &CODE
   BY STORE_CODE
   IF STORE_CODE IS &STORECODE
   END
```

## Supplying Variable Values From Another Procedure

You can supply values for variables in an EXEC command used in a procedure to call another procedure. In Developer Studio, this can also be done with the Execute component. See *Calling Another Procedure With EXEC* on page 10-57 for more information on the use of this technique to control application flow.

Values are assigned using a name/value pair, and multiple pairs must be separated by commas. If the list of pairs exceeds the maximum width of the line, insert a comma as the last character on the line and continue the list on the following line. You do not have to enter the pairs in the order in which they are encountered in the procedure.

With EXEC you can supply values for some, but not all, variables used in the procedure. In that case, values not provided in the EXEC command must be supplied using another Dialogue Manager command. For an example, see *Supplying Values With Another Procedure* on page 10-15.

If the variable is positional (that is, if it is a numbered variable), you do not need to specify the variable name in the EXEC command. WebFOCUS matches the EXEC values to the positional variables as they are encountered in the procedure. Therefore, it is critical to enter the appropriate value for each variable in order. For an example on supplying positional variable values, see *Using Positional Variables* on page 10-16.

### *Reference* Rules for Using Named and Positional Variables With EXEC

You can mix named and positional variables freely in the EXEC command by following these rules:

- Names must be associated with values for named variables.
- Values for positional variables must be supplied in the order that those variables are numbered within the procedure.

The following command is valid for named variables and positional variables:

```
EX HRINFO SALARY=72000, 5, BOS, STATUS=EMPLOYED
```

## *Syntax* How to Supply a Variable Value With Another Procedure

```
EX[EC] procedure name=value
```

where:

*procedure*

Is the name of the procedure that will contain the name/value values.

*name*

Is the variable name.

*value*

Is the value you are giving to the variable.

## *Example* Supplying Values With Another Procedure

Consider the following procedure named SLRPT:

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &CITY"
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY PROD_CODE
IF PROD_CODE IS-FROM &CODE1 TO &CODE2
BY CITY
IF CITY EQ &CITY
END
```

You can call this procedure from another procedure and supply values for the variables as parameters using the EX command as follows:

```
EX SLRPT CITY=STAMFORD, CODE1=B10, CODE2=B20
```

### *Example* Using Positional Variables

Consider the following example:

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &1"
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY PROD_CODE
IF PROD_CODE IS-FROM &2 TO &3
BY CITY
IF CITY EQ &1
END
```

The EX command that calls the procedure is as follows:

```
EX SLRPT STAMFORD, B10, B20
```

This command will substitute STAMFORD for the first positional variable, B10 for the second, and B20 for the third.

## Reading Variable Values From and Writing Variable Values to an External File

You can read variable values from an external file, or write variable values to an external file with the -READ and -WRITE commands.

- You can supply variable values with the -READ command. For example, an external file may contain the start and end dates of a reporting period. Dialogue manager can read these values from an external file and use them a variable in a WHERE command that limits the range of data selected in a report request.
- You can save variable values in an external file with the -WRITE command. For example, a request can store the summed total of sales for the day in an external file so that it can be compared to the following day's total sales.

The external file can be a fixed-format file (in which the data is in fixed columns) or a free-format file (in which the data is comma delimited).

When using a -READ or -WRITE command, the external file must be included in a FILEDEF command in your procedure. A -RUN command must then separate the FILEDEF command and the -READ or -WRITE command.

## *Syntax* How to Retrieve a Variable Value From an External File

```
-READ filename[,] [NOCLOSE] &name[.format.][,][&name][.format.]
```

where:

*filename[,]*

Is the name of the external file, which must be defined to the operating system. A space after *filename* denotes a fixed-format file, while a comma denotes a free-format file.

- On UNIX and Windows NT/2000/XP platforms, a FILEDEF for the external file is required.
- On OS/390, the external file must be allocated in the JCL or dynamically allocated by WebFOCUS with the ALLOCATE command.

NOCLOSE

Keeps the external file open until the -READ operation is complete. Files kept open with NOCLOSE can be closed using the command -CLOSE *filename* or a subsequent -WRITE command. The option NOCLOSE is available only in OS/390.

*&name[,]*

Is the variable name. For free-format files, you may separate the variable names with commas. If the list of variables is longer than one line, end the first line with a comma and begin the next line with a dash followed by a blank. Use a dash and a comma for comma-delimited files.

**Free-format**

```
-READ EXTFILE, &CITY, &CODE1,
- &CODE2
```

**Fixed-format**

```
-READ EXTFILE &CITY.A8. &CODE1.A3.,
-, &CODE2.A3.
```

*.format.*

Is the format of the variable. For a free-format file, you do not have to specify this value, but you may. For a fixed-format file, format is the length or the length and type of the variable (A is the default type). The value of format must be delimited by periods. The format is ignored for comma-delimited files.

**Note:** Instead of using *.format.*, you can specify the length of a variable using -SET and enclosing the corresponding number of blanks in single quotes. For example:

```
-SET &CITY='         ';
-SET &CODE1='    ';
-SET &CODE2='    ';
```

## Syntax How to Write a Variable Value to an External File

```
-WRITE filename [NOCLOSE] text
```

where:

*filename*

Is the name of the file being written to or read from. On UNIX and Windows NT/2000/XP platforms, *filename* must specify the absolute path and file name for the called procedure.

NOCLOSE

Keeps the external file open until the -WRITE operation is complete. A file kept open with NOCLOSE can be closed using the command -CLOSE *filename*. The option NOCLOSE is available only in OS/390.

*text*

Is any combination of variables and text.

## Example Reading a Value From an External File

Assume that EXTFILE is a fixed-format file containing the following data:

```
STAMFORDB10B20
```

To detect the end of a file, the following code tests the system variable &IORETURN. When no records remain to be read, a value equal to zero will not be found.

```
-READ EXTFILE &CITY.A8. &CODE1.A3. &CODE2.A3.
-IF &IORETURN NE 0 GOTO RESUME;
   TABLE FILE SALES
   SUM UNIT_SOLD
   BY CITY
   IF CITY IS &CITY
   BY PROD_CODE
   IF PROD_CODE IS-FROM &CODE1 TO &CODE2
    END
-RESUME
 .
 .
 .
```

## *Example* Reading From and Writing to an External File

The following example reads from and writes to text files and uses FILEDEF commands in a procedure; the numbers on the left apply to the notes that follow. In the example, the user supplies a value for &CITY in an HTML form:

```
1. -TOP
2.    FILEDEF PASS DISK D:PASS.DAT
   -RUN
3. -WRITE PASS &CITY
      TABLE FILE SALES
      HEADING CENTER
      "LOWEST MONTHLY SALES FOR &CITY"
      " "
      PRINT DATE PROD_CODE
      BY LOWEST 1 UNIT_SOLD
      BY STORE_CODE
      BY CITY
      WHERE CITY EQ '&CITY'
      FOOTING CENTER
      "CALCULATED AS OF &DATE"
      ON TABLE SAVE AS INFO
      END
4. -RUN
5.  FILEDEF LOG DISK D:LOG.DAT
   -RUN
      MODIFY FILE SALES
      COMPUTE
      TODAY/I6=&YMD;
      CITY='&CITY';
      FIXFORM X5 STORE_CODE/A3 X15 DATE/A4 PROD_CODE/A3
      MATCH STORE_CODE DATE PROD_CODE
      ON MATCH TYPE ON LOG
      "<STORE_CODE><DATE><PROD_CODE><TODAY>"
      ON MATCH DELETE
      ON NOMATCH REJECT
      DATA ON INFO
      END
6. -RUN
      EX SLRPT3
7. -RUN
8. -GOTO TOP
9. -QUIT
```

The procedure SLRPT3, which is invoked from the calling procedure, contains the following lines:

```
10. -READ PASS &CITY.A8.
      TABLE FILE SALES
      HEADING CENTER
      "MONTHLY REPORT FOR &CITY"
      "LOWEST SALES DELETED"
      " "
      PRINT PROD_CODE UNIT_SOLD RETURNS DAMAGED
      BY STORE_CODE
      BY CITY
      WHERE CITY EQ '&CITY'
      FOOTING CENTER
      "CALCULATED AS OF &DATE"
      END
11. -RUN
```

The procedure processes as follows:

1. -TOP marks the beginning of the procedure.
2. The FILEDEF command defines a file named PASS.
3. The -WRITE command writes the value of &CITY to the text file named PASS. In this case, assume that the value written is STAMFORD.
4. The -RUN command executes the stacked report request. In this case, a text file named INFO is created with the SAVE command. This is a sequential file, containing the result of the report request.
5. The FILEDEF command defines a log file for the subsequent MODIFY request.
6. The -RUN statement executes the stacked MODIFY request. The data comes directly from the INFO file created in the prior report request and is entered using the FIXFORM statement. The product with the lowest value in UNIT_SOLD is deleted from the data source and logged to a log file.
7. The next -RUN command executes the procedure called SLRPT3.
8. The -GOTO TOP statement passes control to the start of the procedure.
9. -QUIT ends processing.
10. The -READ command reads the value for &CITY from the text file PASS. In this case, the value passed is STAMFORD.
11. The -RUN command executes the report request and control is returned to the calling procedure.

### Closing an External File

The -CLOSE command closes an external file opened with the -READ or -WRITE command with the NOCLOSE option. The NOCLOSE option keeps a file open until the -READ or -WRITE operation is complete.

*Syntax* **How to Close an External File**

```
-CLOSE filename
```

where:

*filename*

Is a symbolic name associated with a physical file known to the operating system.

## Supplying Variable Values From Report Painter in Developer Studio

In Developer Studio, in place of Dialogue Manager, the Report Painter provides an intuitive means of embedding variables into your report.

Dialogue Manager commands *within* TABLE commands will be eliminated. However, Dialogue Manager commands used before or after TABLE and END commands will be retained.

You can also use the Report Layout Painter to create forms in which users can supply variable values for your procedure. For details on the Report Layout Painter, see Chapter 6, *Designing a User Interface for a Web Application With the Report Layout Painter*.

*Example* **Supplying Variable Values From the Report Painter**

The following example illustrates how to create procedures containing values from the Report Painter:

1. Create a new procedure.
2. Click and hold a component connector (yellow diamond), then click the *Report* button on the Component Connector toolbox.

   The File Description List For Report window displays.
3. Choose *EMPLOYEE* from the list, and click *Paint*.

   The Report Painter opens.
4. Double-click *PCT_INC*, then *SALARY* in the Employee window.

   The fields display in the Report Painter.
5. Click *By* from the toolbar.
6. Double-click *DEPARTMENT* in the list of fields in the Employee window.

7. Click the *Select* menu, then select *Where/If*.

   The Where/If window displays.

8. Click *Assist*.

   The Expression Builder displays.

9. Select *SALARY* in the Name field, *Is LESS THAN* in the Relations field, and *Parameter (& Variable)* in the Compare Type section.

   The Parameter field displays in the Expression Builder.

10. Enter *SALPROMPT*, and click *OK*.

    The Expressions Builder displays with *SALARY LT &SALPROMT* in the Expressions field.

11. Click *OK*.

    The Where/If window is active.

12. Click *OK*.

13. Click the Run icon to run your request.

    A pop-up window displays with a field for the value of SALPROMPT.

14. Enter *100000*, and click *OK*.

The report output displays in the browser.

## Supplying Dates as Variable Values

You can use variables to supply dates to Dialogue Manager. A date supplied to Dialogue Manager cannot be more than 20 characters long, including spaces. Dialogue Manager variables only accept full-format dates (that is, MDY or MDYY, in any order).

*Example* **Setting the Difference Between Two Dates**

In the following example, the variable &DELAY is set to the difference in days between &LATER and &NOW and the result is returned to your browser.T

```
-SET &NOW = 'JUN 30 2002';
-SET &LATER = '2002 25 AUG';
-SET &DELAY = &LATER - &NOW;
-TYPE &DELAY
```

### Using Variables With Cross-Century Dates

If you are working with cross-century dates that do not include a four-digit year, variables are available which allow you to identify the century. For details, see the legacy repository.

## Screening a Variable Value

To ensure that a supplied value is valid and being used properly in a procedure, you can test it for presence, type, and length. For example, you would not want to perform a numerical computation on a variable for which alphanumeric data has been supplied.

You screen a value by adding a suffix to the variable value:

- The .EXIST suffix tests the presence of a value.
- The .EVAL suffix tests the value of a variable.
- The .LENGTH suffix tests the length of a value.
- The .TYPE suffix tests the type of a value.

### *Syntax* How to Screen a Variable Value

```
-IF &name{suffix} expression GOTO label...;
```

where:

*&name*

Is a user-supplied variable.

suffix

Is one of the following:

.EXIST tests for the presence of a value. If a value is not present, a zero (0) is passed to the expression. Otherwise, a non-zero value is passed.

.LENGTH tests for the length of a value. If a value is not present, a zero (0) is passed to the expression. Otherwise, the number of characters in the value is passed.

.TYPE tests for the type of a value. The letter N (numeric) is passed to the expression if the value can be interpreted as a number up to $10^{9-}1$ and stored in four bytes as a floating point format. In Dialogue Manager, the result of an arithmetic operation with numeric fields is truncated to an integer after the whole result of an expression is calculated. If the value could not be interpreted as numeric, the letter A (alphanumeric) is passed to the expression.

.EVAL tests the value of a variable.

*expression*

Is a valid expression that uses *&name* with the specified suffix as an variable.

GOTO *label*

Specifies a label to branch to.

### *Example* Testing for the Presence of a Variable

If a value is not supplied in the following example, &OPTION.EXIST is equal to zero and control is passed to the label -CANTRUN. The -HTMLFORM displays a message that the procedure cannot run. If a value is supplied, control passes to the label -PRODSALES.

```
-IF &OPTION.EXIST GOTO HRINFO ELSE GOTO CANTRUN;
     .
     .
     .
-HRINFO
   TABLE FILE CENTHR
     .
     .
     .
   END
-EXIT
-CANTRUN
-HTMLFORM BEGIN
<HTML>
<BODY>
TOTAL REPORT CAN'T BE RUN WITHOUT AN OPTION.
</BODY>
</HTML>
-HTMLFORM END
-EXIT
```

### *Example* Testing for the Length of a Variable With an HTML Format Message

If the length of the value supplied for &OPTION is more than one character, control passes to the label -FORMAT. Control then continues to the command -HTMLFORM, which displays a message, informing the user that only a single character can be entered.

```
-IF &OPTION.LENGTH GT 1 GOTO FORMAT ELSE
-GOTO PRODSALES;
      .
      .
      .
-PRODSALES
   TABLE FILE SALES
      .
      .
      .
   END
-EXIT
-FORMAT
-HTMLFORM BEGIN
<HTML>
<BODY>
ONLY A SINGLE CHARACTER IS ALLOWED.
</BODY>
</HTML>
-HTMLFORM END
```

## *Example* Testing for the Length of a Variable

If the length of &OPTION is greater than one character, control passes to the label -FORMAT. The -HTMLFORM command then displays a message informing the user that only a single character can be entered.

```
-TOP
-IF &OPTION.LENGTH GT 1 GOTO FORMAT ELSE
-GOTO PRODSALES;
      .
      .
      .
-PRODSALES
   TABLE FILE SALES
      .
      .
      .
   END
-EXIT
-FORMAT
-HTMLFORM BEGIN
<HTML>
<BODY>
PLEASE ENTER ONLY ONE CHARACTER.
</BODY>
</HTML>
-HTMLFORM
END
```

### *Example* Testing for the Type of a Value

If &OPTION is not alphanumeric, control passes to the label -NOALPHA. -HTMLFORM displays an HTML page that informs the user that only alphanumeric characters are allowed.

```
-IF &OPTION.TYPE NE A GOTO NOALPHA ELSE
- GOTO HRINFO;
      .
      .
      .
-HRINFO
   TABLE FILE CENTHR
      .
      .
      .
   END
-EXIT
-NOALPHA
-HTMLFORM BEGIN
<HTML>
<BODY>
ENTER A LETTER ONLY.
</BODY>
</HTML>
-HTMLFORM
```

## Evaluating a Variable Immediately

The .EVAL operator enables you to evaluate a variable's value immediately, making it possible to change a procedure dynamically. The .EVAL operator is particularly useful in modifying code at run time.

### *Syntax* How to Evaluate a Variable Immediately

```
[&]&variable.EVAL
```

where:

*variable*

Is a local or global variable.

When the command procedure is executed, the expression is replaced with the value of the specified variable before any other action is performed. Without the .EVAL operator, a variable cannot be used in place of some commands.

### *Example* Using .EVAL to Allow WebFOCUS to Interpret a Variable

The code

```
-SET &A='-TYPE';
&A HELLO
```

produces an error message which shows that WebFOCUS does not recognize the value of &A:

```
(FOC1517) UNRECOGNIZED COMMAND -type HELLO
```

Appending the .EVAL operator to the &A variable makes it possible for WebFOCUS to interpret the variable correctly. Adding the .EVAL operator as follows

```
-SET &A='-TYPE';
&A.EVAL HELLO
```

produces the following output:

```
HELLO
```

### *Example* Evaluating a Variable Immediately

The following example illustrates how to use the .EVAL operator in a record selection expression. The numbers to the left apply to the notes that follow the procedure:

```
1. -SET &R='IF SALARY GT 60000';
2. -IF &Y EQ 'YES' THEN GOTO START;
3. -SET &R = '-*';
   -START
4.   TABLE FILE CENTHR
     PRINT SALARY
     BY PLANT BY LNAME
5.   &R.EVAL
     END
```

The procedure executes as follows:

1. The procedure sets the value of &R to 'IF SALARY GT 60000'.
2. If &Y is YES, the procedure branches to the START label, bypassing the second -SET command.
3. If &Y is NO, the procedure continues to the second -SET command, which sets &R to '-*', which is a comment.

   The report request is stacked.
4. The procedure evaluates the value of &R. If the end user wanted a record selection test, the value of &R is 'IF SALARY GT 60000' and this line is stacked.
5. If the end user did not want a record selection test, the value of &R is '-*' and this line is ignored.

## Counting With an Indexed Variable

You can append the value of one variable to the value of another variable, creating an *indexed variable*. This feature applies to both local and global variables.

If the indexed value is numeric, the effect is similar to that of an array in traditional computer programming languages. For example, if the value of index &K varies from 1 to 10, the variable &AMOUNT.&K refers to one of ten variables, from &AMOUNT1 to &AMOUNT10.

A numeric index can be used as a counter; it can be set, incremented, and tested in a procedure.

### *Syntax* How to Create an Indexed Variable

```
-SET &name.&index[.&index...] = expression;
```

where:

*&name*

Is a variable.

*.&index*

Is a numeric or alphanumeric variable whose value is appended to *&name*. The period is required.

When more than one index is used, all index values are concatenated and the string appends to the name of the variable.

For example, &V.&I.&J.&K is equivalent to &V1120 when &I=1, &J=12, and &K=0.

*expression*

Is a valid expression. For information on the kinds of expressions you can write, see the *Creating Reports With WebFOCUS Language* manual.

### *Example* Using an Indexed Variable in a Loop

An indexed variable can be used in a loop. The following example creates the equivalent of a DO loop used in traditional programming languages:

```
-SET &N = 0;
-LOOP
-SET &N = &N+1;
-IF &N GT 12 GOTO OUT;
-SET &MONTH.&N=&N;
-TYPE &MONTH.&N
-GOTO LOOP
-OUT
```

In this example, &MONTH is the indexed variable and &N is the index. The value of the index is supplied through the command -SET; the first -SET initializes the index to 0, and the second -SET increments the index each time the procedure goes through the loop.

If the value of an index is not defined prior to reference, a blank value is assumed. As a result, the name and value of the indexed variable will not change.

Indexed variables are included in the system limit of 384 variables.

## Creating and Working With Variables

You can use variables to customize a procedure, as described in *Customizing a Procedure With Variables* on page 10-9. Variables fall into two categories:

- Local and global variables, whose user-defined values must be supplied at run time. For details, see *Creating and Working With Local and Global Variables* on page 10-31.
- System and statistical variable values are predefined and automatically supplied by the system when a procedure references them. For details, see *Naming Conventions for Local and Global Variables* on page 10-32.

### *Reference* Variable Characteristics

The following features apply to all variables:

- A variable stores numbers or a string of text and can be placed anywhere in a procedure.
- A variable can refer to a command, a data source field, a verb, or a phrase.
- The maximum number of variables allowed in a procedure is 1,024. Because approximately 40 are reserved for server use, the maximum number of *user-named* variables allowed in a procedure is 984.
- The total length of all variable values cannot exceed 32K.
- A variable can be supplied directly in the command, when you execute the procedure, or through the -DEFAULT, -SET, or -READ commands in the procedure.

## Creating and Working With Local and Global Variables

Local and global variables are variables whose user-defined values must be supplied at run time.

- A local variable retains its values during the execution of one procedure; values are lost after the procedure finishes processing. Values are *not* passed to other procedures that contain the same variable name.
- A local variable is identified by a single ampersand followed by the variable name.
- A global variable retains its value for the duration of your connection to the WebFOCUS Reporting Server and are passed from the execution of one procedure to the next.

  Because WebFOCUS creates a new WebFOCUS session on the WebFOCUS Reporting Server each time it submits a request, values for global variables are not retained between report requests. This means that you can use the same global variable in more than one procedure as long as these procedures are called in the same request.

  If you want a global variable's value to be in effect for every procedure, add the variable to a WebFOCUS Reporting Server profile, such as EDASPROF.PRF.

  A global variable is identified by a double ampersand followed by the variable name.

## *Reference* Naming Conventions for Local and Global Variables

Local and global variable names are user-defined, while system and statistical variables have predefined names. The following rules apply to the naming of local and global variables:

- A local variable name is always preceded by an ampersand (&).
- A global variable name is always preceded by a double ampersand (&&).
- Excluding the ampersand, the maximum number of characters permitted in a name is 12.
- Embedded blanks are not permitted in a variable name.
- If a value for a variable might contain an embedded blank, comma, or equal sign, enclose the variable in single quotation marks when you refer to it.
- A variable name may be any combination of the characters A through Z, 0 through 9, and the underscore. The first character of the name should be a letter.
- You can assign a number instead of a name to a variable to create a positional variable.
- The underscore may be included in a variable name, but the following special characters are not permitted: plus sign, minus sign, asterisk, slash, period, ampersand, and semicolon.

## *Syntax* How to Specify a Variable Name

`&[&]name`

where:

*name*

Is the variable name. A single ampersand (&) denotes a local variable, while a double ampersand (&&) denotes a global variable. A single ampersand followed by a numeric string denotes a positional variable.

The name you assign must follow the rules outlined in *Naming Conventions for Local and Global Variables* on page 10-32.

## *Example* Variable Names

The following variables are properly named:

```
&WHICHPRODUCT
&WHICH_CITY
'&CITY'
&&CITY
```

The following variables are improperly named for the reason given:

`&CORPORATECITY`

Too long (exceeds 12 characters).

`&WHICH CITY`

Contains an embedded blank.

`&WHICH -CITY`

Contains a hyphen.

`WHICHCITY`

Leading ampersand(s) missing.

## *Example* Using Local Variables

Consider the following procedure, SALESREPORT, in which &CITY, &CODE1, and &CODE2 are local variables.

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &CITY"
"PRODUCT CODES FROM &CODE1 TO &CODE2"
" "
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ &CITY
BY PROD_CODE
IF PROD_CODE IS-FROM &CODE1 TO &CODE2
END
```

Assume you supply the following values when you call the procedure:

```
EX SALESREPORT CITY=STAMFORD, CODE1=B10, CODE2=B20
```

Dialogue Manager substitutes the values for the variables as follows:

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR STAMFORD"
"PRODUCT CODES FROM B10 TO B20"
" "
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ STAMFORD
BY PROD_CODE
IF PROD_CODE IS-FROM B10 TO B20
END
```

After the procedure executes and terminates, the values STAMFORD, B10, and B20 are lost.

## *Example* Using Global Variables

The following example illustrates the use of three global variables: &&CITY, &&CODE1, &&CODE2. The values are substituted in the first procedure, PROC1, and the values are retained and passed to the second procedure, PROC2.

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &&CITY"
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ &&CITY
BY PROD_CODE
IF PROD_CODE IS-FROM &&CODE1 TO &&CODE2
END
EX PROC2
```

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &&CITY AND PRODUCT &&CODE1"
PRINT UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ &&CITY
IF PROD_CODE EQ &&CODE1
END
```

## Concatenating Variables

You can append a variable to a character string, or you can combine two or more variables and/or literals to concatenate a variable. See the *Creating Reports With WebFOCUS Language* manual for details on concatenation. When using variables, it is important to separate each variable from the concatenation symbol with a space.

## *Syntax* How to Concatenate a Variable

```
-SET &name3 = &name1 || &name2;
```

where:

*&name3*

Is the name of the concatenated variable.

*&name1 || &name2*

Are the variables, separated by the concatenation symbol and a space on both sides.

## Displaying the Value of a Variable

You can display the value of a variable by issuing a query.

### *Syntax* How to Display the Value of a Local Variable

```
-? &[string]
```

where:

*string*

Is a variable name containing up to 12 characters. If this parameter is not specified, the current values of all local, global, system, and statistical variables are displayed.

### *Syntax* How to Display the Value of Global Variables

```
? &&
```

## Assigning the Value of a Variable to a Parameter

You can capture a the value of a SET parameter value in a local variable.

In Developer Studio, the output appears in the Command Console. In WebFOCUS, the result is returned in your browser window, or as a comment in the HTML file if there is other HTML output from the request. Use the applicable Web browser functions to view the HTML comments (for example, View Source in Microsoft Internet Explorer).

### *Syntax* How to Assign the Value of a Parameter to a Variable

```
-? SET parameter variable
```

where:

*parameter*

Is a SET parameter.

*variable*

Is the name of the variable where the value is to be stored.

### *Example* Assigning the Value of a Variable to a Parameter

Entering

```
-? SET ASNAMES &ABC
```

stores the value of ASNAMES as the value for &ABC. If you omit &ABC from the command, then a variable called &ASNAMES is created that contains the value of ASNAMES.

## Working With System and Statistical Variables

System and statistical variable values are predefined and automatically supplied by the system when a procedure references them. System and statistical variables have names that begin with a single ampersand (&). Examples of these variables are &LINES, which indicates how many lines of output were produced, and &DATE, which indicates the current date.

- System values can be used to control processing. For example, if no records are read for a report request because no records fit the selection criteria, processing may halt or the selection criteria may be changed.

  For a list of system variables, see *WebFOCUS System Variables* on page 10-36.

- A statistical variable can be used in a procedure to derive information about the last request run. Dialogue Manager automatically supplies values for statistical variables.

  For a list of statistical variables, see *WebFOCUS System Variables* on page 10-36.

### *Reference* WebFOCUS System Variables

The following table lists the system variables available in WebFOCUS. You can override system-supplied values by assigning a value to the variable explicitly with a -SET or -DEFAULT command. However, we recommend that you do not override these values.

| System Variable | Description | Format or Value |
|---|---|---|
| `&DATE` | Returns the current date. | `MM/DD/YY` |
| `&MDY` | Returns the current date. Useful for numerical comparisons. | `MMDDYY` |
| `&DMY` | Returns the current date. | `DDMMYY` |
| `&FOCCPU` | Calculates the operating system CPU time. | `milliseconds` |
| `&FOCFIELDNAME` | Returns a string indicating whether long and qualified field names are supported. A value of OLD means that they are not; NEW means that they are; and NOTRUNC means that they are supported, but unique truncations of field names cannot be used. | `NEW`<br>`OLD`<br>`NOTRUNC` |

| System Variable | Description | Format or Value |
|---|---|---|
| &FOCFOCEXEC | Manages reporting operations involving many similarly named requests that are executed using the EX command. &FOCFOCEXEC enables you to easily determine which procedure is running by returning the fully qualified path for the procedure. &FOCFOCEXEC can be specified within a request or a Dialogue Manager command to display the name of the current procedure. | |
| &FOCINCLUDE | Manages reporting operations involving many similarly named requests that are included using the -INCLUDE command. &FOCINCLUDE can be specified within a request or a Dialogue Manager command to display the name of the current procedure. | |
| &FOCMODE | Returns the operating environment. | |
| &FOCQUALCHAR | Returns the character used to separate the components of qualified field names. | .<br>:<br>!<br>%<br>\|<br>\ |
| &FOCREL | Returns the release number. | |
| &FOCTMPDSK | Identifies the disk where FOCUS places temporary work files (for example, HOLD files). | A...Z |
| &FOCTTIME | Calculates total CPU time. | milliseconds |
| &FOCVTIME | Calculates virtual CPU time. | milliseconds |
| &IORETURN | Is the value returned after the last -READ or -WRITE operation, which indicates the success of the operation. A value of 0 means the it was successful; 1 means it failed. | 0<br>1 |

| System Variable | Description | Format or Value |
|---|---|---|
| &RETCODE | Is the value returned after a server command is executed.<br>&RETCODE executes all stacked commands, like the command -RUN. | Any value defined by the server command. |
| &TOD | Returns the current time. When you enter FOCUS, this variable is updated to the current system time only when you execute a MODIFY, SCAN, or FSCAN command. To obtain the exact time during any process, use the HHMMSS subroutine. | HH.MM.SS |
| &YMD | Returns the current date. | YYMMDD |

## *Example* Retrieving the Date With &DATE

The following example incorporates the system variable &DATE into a request. The footing uses the system variable &DATE to insert the current system date at the bottom of the report.

```
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE
FOOTING
"CALCULATED AS OF &DATE"
END
```

## *Example* Retrieving the Procedure Name With &FOCFOCEXEC

The following example for Developer Studio incorporates the system variable &FOCFOCEXEC in a report request to display the name of the current procedure.

```
SET PAGE=OFF
TABLE FILE EMPLOYEE
"REPORT: &FOCFOCEXEC -- EMPLOYEE SALARIES"
PRINT CURR_SAL BY EMP_ID
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=10, GRID=OFF,$
END
```

If the request is stored as a FOCEXEC called SALPRINT, the output is:

REPORT: SALPRINT -- EMPLOYEE SALARIES

| EMP_ID | CURR_SAL |
|---|---|
| 071382660 | $11,000.00 |
| 112847612 | $13,200.00 |
| 115360218 | $.00 |
| 117593129 | $18,480.00 |
| 119265415 | $9,500.00 |
| 119329144 | $29,700.00 |
| 121495681 | $.00 |
| 123764317 | $26,862.00 |
| 126724188 | $21,120.00 |
| 219984371 | $18,480.00 |
| 326179357 | $21,780.00 |
| 451123478 | $16,100.00 |
| 543729165 | $9,000.00 |
| 818692173 | $27,062.00 |

## *Reference* WebFOCUS Statistical Variables

The following statistical variables are available in WebFOCUS.

| Statistical Variable | Description |
|---|---|
| &BASEIO | Is the number of input/output operations. |
| &FOCDISORG | Is the percentage of disorganization for a FOCUS file. This variable can be displayed or tested even if the value is less than 30% (the level at which ? FILE displays the amount of disorganization). |
| &FOCERRNUM | Is the last error number displayed after the execution of a procedure. If no error occurs, the value of &FOCERRNUM is 0. |
| &LINES | Is the number of lines printed in last answer set. |

| Statistical Variable | Description |
|---|---|
| &READS | Is the number of physical reads from an external file. |
| &RECORDS | Is the number of records retrieved in last answer set. |

*Example* **Counting the Number of Printed Lines With &LINES**

In the following example, the system calculates the value of the statistical variable &LINES. If &LINES is 0, control passes to the TABLE FILE EMPLOYEE request identified by the label -RPT2. If the value is not 0, control passes to the label -REPTDONE, and processing is terminated.

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &CITY"
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ &CITY
BY PROD_CODE
IF PROD_CODE IS-FROM &CODE1 TO &CODE2
END
-RUN
-IF &LINES EQ 0 GOTO RPT2 ELSE GOTO REPTDONE;
-RPT2
TABLE FILE EMPLOYEE
   .
   .
   .
END
-RUN
-QUIT
-REPTDONE
-EXIT
```

# Controlling the Execution of a Procedure

You can use Dialogue Manager commands to manage the execution of a procedure. The commands used for this purpose are -RUN, -EXIT, -QUIT, and -QUIT FOCUS.

- -RUN causes immediate execution of all stacked commands, closes any external files, and continues the procedure. See *Executing Stacked Commands and Continuing the Procedure* on page 10-41 for more information.
- -EXIT forces the execution of stacked commands, and closes the procedure. For more information, see *Executing Stacked Commands and Exiting the Procedure* on page 10-42.
- -QUIT cancels execution of any stacked commands and causes an immediate exit from the procedure. For more information, see *Canceling the Execution of a Procedure* on page 10-43.
- -QUIT FOCUS terminates a procedure and exits FOCUS in Developer Studio. For more information, see *Canceling the Execution of a Procedure* on page 10-43.

## Executing Stacked Commands and Continuing the Procedure

You can execute stacked commands and continue the procedure with the -RUN command.

The -RUN command causes immediate execution of all stacked commands and closes any external files opened with -READ or -WRITE. Following execution of the stacked commands, processing of the procedure continues with the line that follows -RUN.

### *Example* Executing Stacked Commands and Continuing the Procedure

The following illustrates the use of -RUN to execute stacked code and then return to the procedure. The numbers to the left correspond to the notes explaining the code.

```
1. TABLE FILE SALES
    PRINT PROD_CODE UNIT_SOLD
    BY CITY
   END
2. -RUN
   TABLE FILE EMPLOYEE
    PRINT LAST_NAME FIRST_NAME
    BY DEPARTMENT
   END
```

The procedure processes as follows:

1. The first four lines are the report request. Each line is placed on a stack to be executed later.
2. -RUN causes the stacked commands to be executed. They are sent to the WebFOCUS Reporting Server for processing. Processing continues with the line following -RUN.

## Executing Stacked Commands and Exiting the Procedure

You can execute stacked commands and then exit a procedure with the -EXIT command. -EXIT forces the execution of stacked commands as soon as it is encountered.

- In WebFOCUS, -EXIT closes all external files and terminates the procedure. If the procedure was called by another procedure, control returns to the calling procedure.
- In Developer Studio, -EXIT closes the procedure and returns the user to the Application Manager. If the procedure was called by another procedure, control returns to the calling procedure.

### *Example* Executing Stacked Commands and Exiting the Procedure

In the following, the first report request or the second report request will execute, but not both.

```
1. -SET &PROC = 'SALES'
2. -IF &PROC EQ 'EMPLOYEE' GOTO EMPLOYEE;
   -SALES
3. TABLE FILE SALES
   SUM UNIT_SOLD
    BY PROD_CODE
   END
4. -EXIT
   -EMPLOYEE
   TABLE FILE EMPLOYEE
   PRINT LANT_NAME
    BY DEPARTMENT
   END
```

The procedure processes as follows:

1. Dialogue Manager passes SALES to &PROC.
2. An -IF test is done, and since the value for &PROC is not EMPLOYEE, the test fails and control is passed to the next line, -SALES

   If the value for &PROC had been EMPLOYEE, control would pass to -EMPLOYEE.
3. The FOCUS code is processed, and stacked to be executed later.
4. -EXIT executes the stacked commands. The output is sent to the WebFOCUS client application and the procedure is terminated.

   The request under the label -EMPLOYEE is not executed.

This example also illustrates an implicit exit. If the value of &PROC was EMPLOYEE, control would pass to the label -EMPLOYEE after the -IF test, and the procedure would never encounter -EXIT. The TABLE FILE EMPLOYEE request would execute and the procedure would automatically terminate.

## Canceling the Execution of a Procedure

You can cancel the execution of a procedure with the -QUIT command. -QUIT cancels execution of any stacked commands and causes an immediate exit from the procedure. Control returns directly to the WebFOCUS application regardless of whether the procedure was called by another procedure. This command is useful if tests or computations generate results that make additional processing unnecessary.

In Developer Studio, you can use the -QUIT FOCUS command. This command cancels the execution of a procedure and terminates the FOCUS session. It returns you to the operating system and sets a return code.

### *Syntax* How to Cancel the Execution of a Procedure

```
-QUIT
```

### *Syntax* How to Cancel the Execution of a Procedure and Exit FOCUS

```
-QUIT FOCUS [n]
```

where:

*n*

Is the operating system return code number. It can be a constant or variable. A variable should be an integer. If you do not supply a value or if you supply a non-integer value, a default return code is posted to the operating system, indicating abnormal termination.

A major function of user-controlled return codes is to detect processing problems. The return code value determines whether to continue or terminate processing. This is particularly useful for batch processing.

### *Example* Canceling the Execution of a Procedure

The following example illustrates the use of -QUIT to cancel execution based on the results of an -IF test:

```
1. -DEFAULT &CODE='B11';
2. -IF &CODE EQ '0' OR &CODE EQ 'DONE' GOTO QUIT;
3. TABLE FILE SALES
   SUM UNIT_SOLD
   WHERE PROD_CODE EQ &CODE
   END
4. -QUIT
```

The procedure processes as follows:

1. The -DEFAULT command sets the default value for &CODE to B11.
2. The value B11 is passed to &CODE.
3. The FOCUS code is processed, and stacked to be executed later.
4. -QUIT cancels the execution of stacked commands and exits the procedure.

## Navigating a Procedure

You can navigate a procedure in the following ways:

- **Unconditional branching** transfers control to a label. For details, see *Performing Unconditional Branching* on page 10-45.
- **Conditional branching** transfers control to a label depending on the outcome of a test. For details, see *Performing Conditional Branching* on page 10-46.
- **Looping** performs a function repeatedly in your procedure. For details, see *Looping in a Procedure* on page 10-50.
- **Calling another procedure** incorporates a whole or partial procedure into your procedure. For details, see *Calling Another Procedure With -INCLUDE* on page 10-53 and *Calling Another Procedure With EXEC* on page 10-57.

## Performing Unconditional Branching

You can perform unconditional branching which transfers control to a label with the -GOTO command. When Dialogue Manager encounters a -GOTO command, it searches forward through the procedure for the target label. If it reaches the end without finding the label, it continues the search from the beginning of the procedure. The first time through a procedure, Dialogue Manager notes the addresses of all the labels so that they can be found immediately if needed again. Dialogue Manager takes no action on labels that do not have a corresponding -GOTO.

If a -GOTO does not have a corresponding label, execution halts and an error message is displayed.

### *Syntax* How to Perform Unconditional Branching

```
-GOTO label
   .
   .
   .
-label [TYPE text]
```

where:

*-label*

Is a user-defined name of up to 12 characters. Do not use embedded blanks or the name of any other Dialogue Manager command except -QUIT or -EXIT. Do not use arithmetic or logical operations, or words that can be confused with functions.

The *label* text may precede or follow the -GOTO command in the procedure.

**Note:** When the *label* follows the -GOTO command, a dash does not precede it.

TYPE *text*

Sends a message to a client application.

### *Example* Performing Unconditional Branching

The following example “comments out” all the FOCUS code using an unconditional branch rather than -* in front of every line:

```
-GOTO DONE
TABLE FILE SALES
PRINT UNIT_SOLD RETURNS
BY PROD_CODE,CITY
END
-RUN
-DONE
```

## *Example* Performing Conditional Branching

Conditional branching performs a test of the values of variables and, based on the test, transfers control to a label in the procedure with the -IF... GOTO command. This helps control the execution of requests and builds a dynamic procedure by choosing to execute or not execute parts of a procedure.

For example, you can check whether an extract file was created from a production data source. If the extract file exists, the program runs a set of reports against the extract. If it does not exist, the program branches around the reports and writes a message to a log file.

**Note:** An -IF test does not require that each test specify a target label.

## *Syntax* How to Perform Conditional Branching

```
-IF expression [THEN] GOTO label1; [ELSE IF...;] [ELSE GOTO label2;]
```

where:

*expression*

Is a valid expression. Literals do not need to be enclosed in single quotation marks unless they contain embedded blanks or commas.

THEN

Is an optional command that increases readability of the command.

*label1*

Is a user-defined name of up to 12 characters to which to pass control if the -IF test is true. Do not use embedded blanks or the name of any other Dialogue Manager command except -QUIT or -EXIT. Do not use arithmetic or logical operations, or words that can be confused with functions.

The *label* text may precede or follow the -IF criteria in the procedure.

ELSE IF

Specifies a compound -IF test. The command -IF must end with a semicolon to signal that all logic has been specified. For more information see *Performing a Compound -IF Test* on page 10-49.

ELSE GOTO

Passes control to *label2* when the -IF test fails.

If a command spans more than one line, continuation lines must begin with a hyphen and one or more spaces.

### *Example* Performing Conditional Branching

The following example passes control to the label -PRODSALES if &OPTION is equal to S. Otherwise, control passes to the label -PRODRETURNS, the next line in the procedure.

```
-IF &OPTION EQ 'S' GOTO PRODSALES;
-PRODRETURNS
TABLE FILE SALES
PRINT PROD_CODE UNIT_SOLD
BY STORE_CODE
END
-EXIT
-PRODSALES
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE
END
-EXIT
```

The following command specifies both transfers explicitly:

```
-IF &OPTION EQ 'S' GOTO PRODSALES ELSE
                 - GOTO PRODRETURNS;
```

Notice that the continuation line begins with a hyphen and includes a space after the hyphen.

### *Example* Testing System and Statistical Variables

In the following example, if data (&LINES) is retrieved with the request, then the procedure branches to the label -PRODSALES; otherwise, it terminates.

```
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE BY CITY
WHERE TOTAL UNIT_SOLD GE 50
ON TABLE HOLD
END
-IF &LINES NE 0 GOTO PRODSALES;
-EXIT
-PRODSALES
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE ACROSS CITY
END
```

## *Example* Executing Stacked Commands and Exiting the Procedure

In the following example, the first report request or the second report request, but not both, will execute. Assume the report is launched by an HTML form that uses a text input variable named &PROC to prompt the user for the procedure to run. The user may enter SALES or EMPLOYEE. For more information on launching a report from an HTML form, see Chapter 5, *Understanding User Interface Basics*.

```
1. -IF &PROC EQ 'EMPLOYEE' GOTO EMPLOYEE;
2. -SALES
   TABLE FILE SALES
   SUM UNIT_SOLD
    BY PROD_CODE
   END
3. -EXIT
   -EMPLOYEE
   TABLE FILE EMPLOYEE
   PRINT LANT_NAME
    BY DEPARTMENT
   END
```

The procedure processes as follows:

1. Dialogue Manager passes SALES to &PROC. An -IF test is done, and since the value for &PROC is not EMPLOYEE, the test fails and control is passed to the next line, -SALES

   If the value for &PROC had been EMPLOYEE, control would pass to -EMPLOYEE.

2. The FOCUS code is processed, and stacked to be executed later.

3. -EXIT executes the stacked commands. The output is sent to the WebFOCUS client application, and the procedure is terminated.

   The request under the label -EMPLOYEE is not executed.

This example also illustrates an implicit exit. If the value of &PROC was EMPLOYEE, control would pass to the label -EMPLOYEE after the -IF test, and the procedure would never encounter -EXIT. The TABLE FILE EMPLOYEE request would execute and the procedure would automatically terminate.

*Example* **Canceling the Execution of a Procedure With Conditional Branching**

The following example illustrates the use of -QUIT to cancel execution based on the results of an -IF test. Assume the report is launched by an HTML form that uses a text input variable named &SELECT to prompt the user for the procedure to run.

```
-IF &SELECT EQ 'DONE' GOTO QUIT;
-REPORT
   TABLE FILE SALES
   SUM UNIT_SOLD
   BY PROD_CODE
   WHERE AREA EQ '&SELECT';
   END
-RUN
-QUIT
```

The user can run sequential reports by hitting the Back button in the browser and entering a new value for SELECT and resubmitting the form. If a product code is selected, the procedure continues to the next line and the request executes before encountering -QUIT. When the user enters DONE, control passes to -QUIT. The procedure is exited.

## Performing a Compound -IF Test

A compounded -IF test is a series of -IF tests nested within each other. You can use a compound -IF test if each test specifies a target label.

*Example* **Using a Compound -IF Test**

In the following example, if the value of &OPTION is neither R nor S, the procedure terminates (-GOTO QUIT). -QUIT serves both as a target label for the GOTO and as an executable command. Assume the report is launched by an HTML form that uses a text input variable named &OPTION to prompt the user for the procedure to run.

```
-IF &OPTION EQ 'R' THEN GOTO PRODRETURNS ELSE IF
- &OPTION EQ 'S' THEN GOTO PRODSALES ELSE
- GOTO QUIT;
-PRODRETURNS
TABLE FILE SALES
PRINT PROD_CODE UNIT_CODE
BY STORE_CODE
END
-EXIT
-PRODSALES
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE
END
-QUIT
```

## Looping in a Procedure

You can perform a function repeatedly by using looping in your procedure with the -REPEAT command. Looping can be used for many tasks. For example, files can be named and renamed by embedding operating system calls within a Dialogue Manager loop. Indexed variables can also be populated using a loop, or the output of a request can be used for a second request.

A process loop can be executed a designated number of times or until a condition is met. A loop ends when any of the following occurs:

- It is executed in its entirety.
- A -QUIT or -EXIT command is issued.
- A -GOTO is issued to a label outside of the loop.

  **Note:** If you issue another -GOTO later in the procedure to return to the loop, the loop proceeds from the point it left off.

You can also limit the repetition of a loop by incrementing a variable with the -SET command.

### *Syntax* How to Specify a Loop

```
-REPEAT label n TIMES
```

or

```
-REPEAT label WHILE condition
```

or

```
-REPEAT label FOR &variable [FROM fromval] [TO toval] [STEP s]
```

where:

*label*

Identifies the code to be repeated (the loop). A label can include another loop if the label for the second loop has a different name than the first.

*n* TIMES

Specifies the number of times to execute the loop. The value of *n* can be a local variable, a global variable, or a constant. If it is a variable, it is evaluated only once, so you cannot change the number of times to execute the loop. The loop can only be ended early using -QUIT or -EXIT.

WHILE *condition*

Specifies the condition under which to execute the loop. The condition is any logical expression that can be true or false. The loop executes if the condition is true.

*&variable*

Is a variable that is tested at the start of each execution of the loop. It is compared with the value of *fromval* and *toval*, if supplied. The loop is executed only if *&variable* is less than or equal to *toval* when *s* is positive, or greater than or equal to *toval* when *s* is negative.

*fromval*

Is the minimum value of &*variable* that will execute a loop. The default value is 1.

*toval*

Is the maximum value of &*variable* that will execute a loop. The default value is 1,000,000.

STEP *s*

Increments the value of &*variable* by a constant, *s*. It may be positive or negative. The default increment is 1.

**Note:** The parameters FROM, TO, and STEP can appear in any order.

## *Example* Repeating a Loop

These examples illustrate each syntactical element of -REPEAT.

```
-REPEAT label n TIMES
```

For example:

```
-REPEAT LAB1 2 TIMES
-TYPE INSIDE
-LAB1 TYPE OUTSIDE
```

The output is:

```
INSIDE
INSIDE
OUTSIDE
```

```
-REPEAT label WHILE condition
```

For example:

```
-SET &A = 1;
-REPEAT LABEL WHILE &A LE 2;
-TYPE &A
-SET &A = &A + 1;
-LABEL TYPE END: &A
```

The output is:

```
1
2
END: 3
```

```
-REPEAT label FOR &variable FROM fromval TO toval STEP s
```

For example:

```
-REPEAT LABEL FOR &A STEP 2 TO 4
-TYPE INSIDE &A
-LABEL TYPE OUTSIDE &A
```

The output is:

```
INSIDE 1
INSIDE 3
OUTSIDE 5
```

## *Example* Controlling Loops

The following example illustrates the use of -SET to control a loop:

```
1. -DEFAULT &N=0
2. -START
3. -SET &N=&N+1;
4.   EX SLRPT
   -RUN
5. -IF &N GT 5 GOTO NOMORE;
6. -GOTO START
5. -NOMORE TYPE EXCEEDING REPETITION LIMIT
   -EXIT
```

The procedure executes as follows:

1. The -DEFAULT command gives &N the initial value of 0.
2. -START begins the loop. This is also the target of an unconditional -GOTO.
3. The -SET command increments the value of &N by one each time the loop executes.
4. The FOCUS command EX SLRPT is stacked. -RUN then executes the stacked command.
5. The -IF command tests the current value of the variable &N. If the value is greater than 5, control passes to the label -NOMORE, which displays a message for the end user and forces an exit. If the value of &N is 5 or less, control goes to the next Dialogue Manager command.
6. -GOTO passes control to the -START label, and the loop continues.

## Calling Another Procedure With -INCLUDE

You can call a procedure from another procedure with the -INCLUDE command, which can incorporate a whole or partial procedure. A partial procedure might contain heading text, or code that should be included at run time based on a test in the calling procedure. It executes immediately when encountered.

A calling procedure cannot branch to a label in a called procedure, and vice versa. When a procedure is included using the -INCLUDE command, the procedure being included (called) has full access to variables defined in the calling procedure.

Procedures called using -INCLUDE must be in the WebFOCUS Reporting Server's search path. For details, see *Storing and Searching for Application Files* in Chapter 2, *WebFOCUS Application Logic*.

Procedures called using the -INCLUDE command must be in the WebFOCUS Reporting Server's search path. For information about the search path, see *Storing and Searching for Application Files* in Chapter 2, *WebFOCUS Application Logic*.

In Developer Studio, the -INCLUDE command can be inserted with the Include component. For information on using the Include component, see *How to Call a Procedure With the Include Component in Developer Studio* on page 10-54.

*Reference*

### Uses for -INCLUDE

The -INCLUDE command can be used for the following:

- Controlling the environment. For example, the called procedure may set variables such as server name or user name before the calling procedure continues execution.
- As a security mechanism. The included procedure can be encrypted and a direct password set.
- Shortening the code when there are several possible procedures that may be called. For example, the command -INCLUDE &NEWLINES could be used to determine the called procedure, reducing the number of GOTO commands.
- Continuing sections of code used throughout the application such as standard headings and footings. This enables changes made in a single module effect the entire application.

## *Procedure* How to Call a Procedure With the Include Component in Developer Studio

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.

## *Syntax* How to Call Another Procedure With -INCLUDE

```
-INCLUDE filename
```

where:

*filename*

Is the name of the called procedure:

- On UNIX and NT platforms, the procedure must have the extension .fex.
- On UNIX, *filename* must specify the absolute path and file name for the called procedure. For example, F:\\ibi\apps\sales\headings.fex.
- On OS/390, the included file must reside in the EDARPC library.

### *Example* Calling Another Procedure With -INCLUDE

In the following example, Dialogue Manager searches for a procedure named DATERPT as specified on the -INCLUDE command.

```
-IF &OPTION EQ 'S' GOTO PRODSALES
-ELSE GOTO PRODRETURNS;
    .
    .
    .
-PRODRETURNS
-INCLUDE DATERPT
-RUN
    .
    .
    .
```

Assume that DATERPT contains the following code, which Dialogue Manager incorporates into the original procedure. Dialogue Manager substitutes a value for the variable &PRODUCT as soon as the -INCLUDE is encountered. -RUN executes the request.

```
TABLE FILE SALES
PRINT PROD_CODE UNIT_SOLD
WHERE PROD_CODE = '&PRODUCT';
END
```

### *Example* Calling a Procedure With a Heading

The following incorporates a heading, which is stored as a procedure:

```
TABLE FILE SALES
-INCLUDE SALEHEAD
SUM UNIT_SOLD AND RETURNS AND COMPUTE
    .
    .
    .
```

The file SALEHEAD contains:

```
HEADING
"THE ABC CORPORATION"
"RETAIL SALES DIVISION"
"MONTHLY SALES REPORT"
```

This heading is included in the report request.

### *Example* Calling a Procedure for a Virtual Field

The following incorporates a virtual field from a procedure:

```
-INCLUDE DEFRATIO
   TABLE FILE SALES
-INCLUDE SALEHEAD
   SUM UNIT_SOLD AND RETURNS AND RATIO
   BY CITY
   .
   .
   .
```

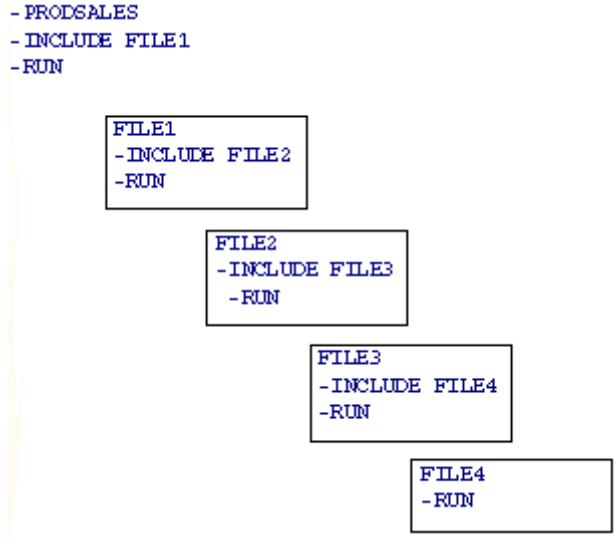
The file DEFRATIO creates a virtual field:

```
DEFINE FILE SALES
RATIO/D5.2=(RETURNS/UNIT_SOLD);
END
```

This virtual field will be dynamically included before the report request executes.

## Nesting Procedures With -INCLUDE

Any number of different procedures can be invoked from a single calling procedure. You can also nest -INCLUDE commands within each other as seen here. There is no limit to the number of -INCLUDE commands that can be nested.




Files one through four are incorporated into the original procedure. All of the included files are viewed as part of the original procedure.

## Calling Another Procedure With EXEC

You can call a procedure from another procedure with the EXEC command. The called procedure must be fully executable. It behaves as a completely separate procedure, with its own content. It cannot use any local variables (&variables) defined by the *calling* procedure (unless they are explicitly passed to the *called* procedure on the command line). However, the executed (called) procedure can use any global variables (&&variables) that have been defined in the calling procedure.

When an EXEC command is encountered, it is stacked and executed when the appropriate Dialogue Manager command is encountered. The called procedure must be fully executable.

Procedures called using the -INCLUDE command must be in the WebFOCUS Reporting Server's search path. For information about the search path, see *Storing and Searching for Application Files* in Chapter 2, *WebFOCUS Application Logic*.

In Developer Studio, you can insert the EXEC command with the Execute component.

### *Procedure* How to Execute a Procedure With the Execute Component

1. Click and hold the component connector at the point at which you wish to include the existing procedure.
2. The Component Connector toolbox opens.
3. Click the *Execute* button.

   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 Execute component inserts the EX command in the procedure's code. Right-click the *Execute* object and choose *Quick View* from the shortcut menu 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.

### *Syntax* How to Call a Procedure With the EXEC Command

```
EX[EC] procedure
```

where:

*procedure*

Is the name of the procedure.

### *Example* Calling a Procedure With EXEC

In the following example, a procedure calls DATERPT:

```
-IF &OPTION EQ 'S' GOTO PRODSALES ELSE GOTO PRODRETURNS;
     .
     .
     .
-PRODRETURNS
     EX DATERPT
     .
     .
     .
-RUN
```

## Enhancing an HTML Web Page With a Procedure

You can enhance the functionality of your Web page by enabling you to include HTML commands in your procedures using the Dialogue Manager -HTMLFORM command. -HTMLFORM supports all standard HTML elements, including character styling, hyperlinks, graphic images, tables, forms, and frames. You can combine a procedure and HTML commands in the following ways:

- Refer to a Web page file that is merged with your report output from a procedure. See *Referring to an External Web Page* on page 10-59 for the syntax.
- Embed HTML commands in a procedure. To do this, enter the appropriate syntax in your procedure using an editor. For this syntax, see *Embedding HTML Commands in a Procedure* on page 10-59.
- Embed a procedure in HTML commands. For details, see *Embedding a Procedure in an HTML Web Page* on page 10-62.
- Include local and global variables, identified with WebFOCUS escape sequences, in your Web page. See *Including Variables in an HTML Web Page* on page 10-65.

## Referring to an External Web Page

You can include HTML commands in your procedure by specifying an HTML Web page in your procedure.

### *Syntax* How to Refer to an External Web Page From a Procedure

```
-HTMLFORM filename
```

where:

*filename*

Is the HTML file that contains the target Web page. The name of the HTML file must be in uppercase letters.

- In UNIX and Windows NT/2000/XP, this is the 8-character file name of the HTML file that contains the Web page. The file extension must be .HTM, not .HTML.
- In Windows NT/200/XP and UNIX, the file extension must be .HTM.
- In OS/390, this is either the member name of a file in the PDS allocated to ddname WWWHTM, or it is the ddname of a sequential file. The file is variable blocked (VB).

If you use separate HTML files, you cannot use Dialogue Manager commands and variables in the file. However, the WebFOCUS escape sequences, described in *Including Variables in an HTML Web Page* on page 10-65, can be used.

## Embedding HTML Commands in a Procedure

You can include HTML commands in a procedure by embedding those commands in a procedure.

### *Syntax* How to Embed HTML Commands in a WebFOCUS Procedure

```
-HTMLFORM BEGIN
        .
        .
        .
-HTMLFORM END
```

where:

-HTMLFORM BEGIN

Indicates the beginning of HTML commands.

-HTMLFORM END

Indicates the end of HTML commands.

**Note:** All lines that are not Dialogue Manager commands are assumed to be HTML.

## *Example* Including a Web Page in a Procedure

The following procedure contains a report request and the -HTMLFORM command, which customizes the Web page the report is displayed on. The report is centered in the browser window, and displays with the Arial font in blue.

```
TABLE FILE SHORT
SUM PROJECTED_RETURN
BY COUNTRY
ON TABLE HOLD AS SHORT FORMAT HTMTABLE
END

-HTMLFORM BEGIN
<HTML><HEAD><STYLE>TD {FONT-FAMILY: ARIAL; COLOR: BLUE; }</STYLE></HEAD>
<BODY><DIV ALIGN="CENTER">
!IBI.FIL.SHORT;
</DIV></BODY></HTML>
-HTMLFORM END
```

The following Web page is produced:

| PAGE 1 | |
|---|---|
| Country | Projected Annualized Return |
| ARGENTINA | 4.200 |
| BRAZIL | 4.200 |
| CANADA | 3.990 |
| CZECH. REP | 12.110 |
| ENGLAND | 4.060 |
| FRANCE | 3.920 |
| GUATEMALA | 4.760 |
| HONDURAS | 4.760 |
| HONG KONG | 4.550 |
| ISRAEL | 3.780 |
| JAPAN | 4.550 |
| MEXICO | 5.040 |
| SAUDI ARABIA | 4.200 |
| SKLARVIA | 4.040 |
| UNITED STATES | 3.430 |

## Embedding a Procedure in an HTML Web Page

You can embed one or more reports or HTML files into your Web page using -HTMLFORM commands.

### *Procedure* How to Embed a Report in an HTML Web Page

To embed a report into a Web page created with -HTMLFORM commands, do the following:

1. Create a report procedure that retrieves and formats the report data, then saves the output from *each* report in a separate temporary file. The following syntax is used in the report procedure:

   ```
   ON TABLE HOLD FORMAT HTMTABLE AS reportname
   ```

   **Note:** The temporary file created for the output is not available to the user.

2. If you are using an external HTML file to display the Web page that embeds the report, the report procedure must specify the file using the following command:

   ```
   -HTMLFORM filename
   ```

   **Note:** If you are not using an external HTML file, the HTML commands for the Web page can reside in the same file using -HTMLFORM BEGIN and -HTMLFORM END to separate the HTML commands from the rest of the procedure.

3. In the HTML commands for the Web page, specify the location where *each* report or HTML file is to be displayed, using an HTML escape code. The escape code syntax is

   ```
   !IBI.FIL.reportname
   ```

   where:

   *reportname*
   Is the name of the report file to embed in the Web page.

   Files embedded with the !IBI.FIL escape sequence can contain other escape sequences. If reference is made to a file that does not exist, an error message is returned.

## *Example* A Web Page With Embedded Variables

This example generates a report consisting of product codes and quantities, and displays the current time and date found in Dialogue Manager system variables. The following explains how the elements required to display this report are combined. The numbers to the left of the example apply to the notes that follow.

### Step 1 - First.htm

Set up the Web page that will hold the report. This Web page must be saved in a directory within EDAPATH or APPATH in order for WebFOCUS to find it. The following file is called first.htm:

```
   <HTML>
   <BODY>
2. Time:
3. !IBI.FIL.&TOD <BR>
2. Date:
3. !IBI.FIL.&DATE <BR>
   <HR SIZE=5>
4. <!--WEBFOCUS TABLE UPPER>
   </BODY>
   </HTML>
```

### Step 2 - Amper.fex

Create your report request:

```
    TABLE FILE GGORDER
    SUM QUANTITY BY PCD
    IF PCD EQ 'B$$$'
1. ON TABLE HOLD FORMAT HTMTABLE AS UPPER
    END
5. -RUN
6. -HTMLFORM first
```

### Step 3 - Launch.htm

Create a Web page, which launches the report procedure:

```
<HTML>
<BODY>
<P>
<P>
<A HREF="/cgi-bin/ibi_cgi/ibiweb.exe?IBIF_ex=amper">Click here.</A>
</BODY>
</HTML>
```

When the procedure and both Web pages have been created, the Web page that launches the procedure can be called from the browser. When the procedure is launched, the report is executed:

1. The HOLD command extracts report output into a temporary file, from which it can be sent to a Web page.
2. The code Time: and Date: specify text that will appear above the report on the Web page.
3. The HTML comments are read by WebFOCUS and identify the variables to be printed on the Web page.
4. The HTML comment is read by WebFOCUS and identifies the report to be displayed. This table name is the name assigned in the HOLD command in the procedure. On the Web page, WebFOCUS substitutes the designated report for the corresponding code.
5. The -RUN command runs the report.
6. The Dialogue Manager command merges the report output with the contents of first.htm, the HTML file that contains your Web page.

The following Web page is generated:

Time: 10.29.36
Date: 08/30/01

| PAGE 1 | |
|---|---|
| Product Code | Ordered Units |
| B141 | 100427 |
| B142 | 285689 |
| B144 | 61498 |

## Including Variables in an HTML Web Page

In WebFOCUS, you can use Dialogue Manager variables to insert variable values in a Web page. This is done by specifying the variable name using a WebFOCUS escape code. You can use the value of a variable either as text on an HTML Web page or as a value in an HTML form input field. You can include a variable that you have created, or system variables like &DATE (today's date) and &TOD (time of day). A single Web page can contain many variable substitution tags.

To use a variable in a Web page created with the -HTMLFORM commands, the variable must first be declared with -SET or -DEFAULT. See *Customizing a Procedure With Variables* on page 10-9 for information on using variables.

### *Syntax* How to Return the Value of a Variable in an HTML Web Page

```
<!--WEBFOCUS VAR [&]&variable>
```

or

```
!IBI.{AMP|GLB}.[&]&variable;
```

where:

`AMP`
: Returns the value of a local variable.

`GLB`
: Returns the value of a global variable.

*`variable`*
: Is a local or global variable.

### *Syntax* How to Return the Length of a Variable in an HTML Web Page

```
!IBI.{AML|GLL}.[&]&variable;
```

where:

`AML`
: Returns the length of a local variable.

`GLL`
: Returns the length of a global variable.

### *Syntax* How to Use a Variable in a Web Page

The syntax for a local variable in a form input field is

```
<INPUT TYPE=TEXT NAME=variable     VALUE="!IBI.AMP.variable;">
```

where:

*variable*

Is the variable name.

When making substitutions, the full string is replaced, starting with the exclamation point and ending with the semicolon. You cannot include variables whose values include internal escape sequences in your command stream.

## Issuing Operating System Commands

You can issue an operating system command to set up an environment in which a request must run. For example, a program may change directories, rename files, unzip files, copy files, or perform other operations before executing a request. After the environment is customized, the program can begin the report and graph requests.

**Note:** In Developer Studio, only -DOS and -WINNT can be used.

### *Syntax* How to Execute an Operating System Command

```
op_system command
```

where:

*op_system*

Specifies the operating system.

`-DOS` specifies the DOS operating system.

`-MVS` specifies the OS/390 operating system.

`-TSO` specifies the OS/390/TSO operating system.

`-UNIX` specifies the UNIX operating system.

`-WINNT` specifies the Windows NT/2000/XP operating system.

*command*

Is an operating system command.

### Reviewing Command Output

Some operating system commands perform a function and do not display additional output. For example, the commands

```
-UNIX cp a.fex b.fex
-DOS copy a.fex b.fex
```

do not display any output when the command is executed.

Other commands return output when the command is executed. For example, the commands

```
-UNIX ls -l grep fex
-DOS dir *.fex
```

display a list of files in the current directory that ends in fex. WebFOCUS does not display operating system command results in the browser. To review the results of an operating system command, you must redirect the output to another file.

For example, the commands

```
-UNIX ls -l|grep fex > /tmp/output
-DOS dir *.fex > c:\tmp\output
```

redirect the command output to a file named OUTPUT, which you can review later in an editor.

## Controlling Passwords With Dialogue Manager

You can issue and control passwords with the -PASS command. This is especially useful for specifying a particular file or set of files that a given user can read to or write from. Passwords have detailed sets of functions associated with them through the DBA facility.

The procedure that sets passwords can be encrypted so that it and the passwords that it sets cannot be typed and made known.

A variable can also be associated with -PASS so that you can prompt for and assign a password value.

*Syntax* **How to Specify a Password**

```
-PASS password
```

where:

*password*

Is a password or a variable containing a password.

# Sending a Message to the Application

You can send a message to an application while a procedure is processing with the -TYPE command. You can use a message to do the following:

- Explain the purpose of the procedure.
- Display the results of a procedure or calculation during testing of a procedure.
- Present useful information.
- Pass CGI variables to WebFOCUS. For details, see Chapter 9, *Enhancing a User Interface*.

In WebFOCUS, depending on the execution of the procedure, the message may be displayed in the browser, or as a comment in the HTML file. To view a comment in the HTML file, use the appropriate Web browser option, for example, View Source in Microsoft Internet Explorer.

In Developer Studio, the message is displayed in the Console window.

***Syntax*** **How to Send a Message**

`-TYPE text`

or

`-label TYPE text`

where:

*text*

Is the message to be sent. If you include quotation marks around the text, they will be displayed as part of the message. The length of the message can be up to 495 characters.

In Developer Studio, the message is sent as soon as -TYPE is encountered in the processing of the procedure. In WebFOCUS, the message is included in the HTML source file.

*-label*

Is the target of a -GOTO or -IF.

**Note:** Labels are the only Dialogue Manager commands that can be placed on the same line as another Dialogue Manager command.

### *Example* Sending a Message

The following example illustrates the use of -TYPE to inform a client application about the contents of a report:

```
-* Version 1  06/26/00  SLRPT  Procedure
-* Component of Retail Sales Reporting Module
-TYPE This report calculates percentage of returns.
TABLE FILE SALES
 .
 .
 .
END
```

## Testing and Debugging a Dialogue Manager Procedure

You can test and debug your procedure with the following.

- The &ECHO variable displays command lines as they execute in order to test and debug procedures.
- The &STACK variable tests the logic of Dialogue Manager commands. Setting this variable allows you to run the procedure but prevent the execution of stacked commands. This gives you the ability to view the sequence of commands and how the variable values are resolved.
- The &RETCODE variable returns a code after a query is executed. If the query results in a normal display, the value of &RETCODE is 0. If a display error occurs, or no display results (as can happen when the query finds no data), the value of &RETCODE is 8. (If the error occurs on a ? SU, the value of &RETCODE is 16.)
- The &RETCODE can be used to test the result of the following:
  - A -READ command. This determines whether a value has been read from the external file.
  - A -WRITE command. This determines whether a value has been written to the external file.
  - An operating system command. This returns a value if a command has been executed.

## *Syntax* How to Test and Debug a Procedure

```
{-DEFAULT|-SET|EX} &ECHO = {ON|ALL|OFF}
```

where:

ON

Displays FOCUS commands that are expanded and stacked for execution.

ALL

Displays Dialogue Manager commands and FOCUS commands that are expanded and stacked for execution.

OFF

Suppresses the display of both stacked commands and Dialogue Manager commands. This value is the default.

## *Syntax* How to Test Dialogue Manager Command Logic

```
{-DEFAULT|-SET|EX} &STACK = {ON|OFF}
```

where:

ON

Executes stacked commands normally. This value is the default.

OFF

Prevents the execution of stacked commands. In addition, system variables (for example, &RECORDS or &LINES) are not set. Dialogue Manager commands are executed so you can test the logic of the procedure.

## *Example* Using the &RETCODE Variable to Test the Result of a Command

The following command tests the existence of the C:\MYDATA\WEEK27.DAT file. If it exists, the value of &RETCODE is 0, and the file is erased.

```
DOS STATE C:\MYDATA\WEEK27.DAT
-IF &RETCODE NE 0 GOTO START;
DOS ERASE C:\MYDATA\WEEK27.DAT
-START
```

# Dialogue Manager Syntax Reference

This topic describes all the Dialogue Manager commands in alphabetical order.

| **Command:** | `-*` |
|---|---|
| Function: | Signals the beginning of a comment line.<br><br>Any number of comment lines can follow one another, but each must begin with -*. A comment line may be placed at the beginning or end of a procedure, or in between commands. However, it cannot be on the same line as a command.<br><br>Use comment lines to document a procedure so that its purpose and history are clear to others. |
| Syntax: | `-* text`<br><br>where:<br><br>*text*<br>Is a comment. A space is not required between -* and *text*. |

| **Command:** | `-?` |
|---|---|
| Function: | Displays the current value of a local variable. |
| Syntax: | `-? &[string]`<br><br>where:<br><br>*string*<br>Is a variable name of up to 12 characters. If this parameter is not specified, the current values of all local, global, and defined system and statistical variables are displayed. |

| **Command:** | **-CLOSE** |
|---|---|
| Function: | Closes an external file opened with the -READ or -WRITE command using the NOCLOSE option. The NOCLOSE option keeps a file open until the -READ or -WRITE operation is complete. |
| Syntax: | -CLOSE *filename*<br>where:<br>*filename*<br>Is a symbolic name associated with a physical file known to the operating system. |

| **Command:** | **-DEFAULT** |
|---|---|
| Function: | Supplies a default value for a variable that had no value before the command was processed.<br>You can override values set with -DEFAULT by supplying new values when you call the WebFOCUS scripts, or by issuing -SET after -DEFAULT.<br>By supplying values to variables in a stored procedure, -DEFAULT helps ensure that it will run correctly. |
| Syntax: | -DEFAULT &[&]*name=value* [...]<br>where:<br>*name*<br>Is a name of a variable.<br>*value*<br>Is the default value assigned to the variable. |

| **Command:** | **-DOS** |
|---|---|
| Function: | Executes a DOS operating system command from a procedure. |
| Syntax: | -DOS *command*<br>where:<br>*command*<br>Is a DOS command. |

| **Command:** | **-EXIT** |
|---|---|
| Function: | Forces a procedure to end. All stacked commands are executed and the procedure exits. If the procedure was called by another one, the calling procedure continues processing.<br><br>Use -EXIT for terminating a procedure after processing a final branch that completes the desired task. The last line of a procedure is an implicit -EXIT. |
| Syntax: | -EXIT |

| **Command:** | **-GOTO** |
|---|---|
| Function: | Transfers control to a specified label.<br><br>If Dialogue Manager finds the label, processing continues with the line following it. If Dialogue Manager does not find the label, processing ends and an error message is displayed. |
| Syntax: | -GOTO *label*<br>.<br>.<br>.<br>-*label* [TYPE *text*]<br><br>where:<br><br>*label*<br>Is a user-defined name of up to 12 characters that specifies the target of the -GOTO action.<br>Do not use embedded blanks or the name of any other Dialogue Manager command except -QUIT or -EXIT. Do not use words that can be confused with functions, arithmetic and logical operations, and so on.<br><br>TYPE *text*<br>Sends a message to the client application. |

| **Command:** | **-HTMLFORM** |
|---|---|
| Function: | Sends report output to the HTML file you create for a Web page and displays that page in the browser. The output can also be saved as an HTML file that can be displayed later. |
| Syntax: | `-HTMLFORM filename [SAVE AS htmlpage]`<br>or<br>`-HTMLFORM BEGIN`<br>`.`<br>`.`<br>`.`<br>`-HTMLFORM END`<br>where:<br>*filename*<br>Is the HTML file that contains placeholders for the report or reports WebFOCUS creates with the command:<br>`ON TABLE HOLD HTMTABLE AS report`<br>`SAVE`<br>Indicates that the HTML page, created when the HTML file *filename* and the report or reports generated by WebFOCUS are combined, is to be saved.<br>`AS htmlpage`<br>Is the file name for the HTML page that is created when the HTML file *filename* and the report or reports generated by WebFOCUS are combined. This file is saved so that it can be displayed later.<br>`-HTMLFORM BEGIN`<br>Indicates the beginning of an inline HTML form in a procedure.<br>`-HTMLFORM END`<br>Indicates the end of an inline HTML form in a procedure. |

| **Command:** | **-IF** |
|---|---|
| Function: | Routes execution of a procedure based on the evaluation of the specified expression.<br><br>An -IF command without an explicitly specified ELSE whose expression is false continues processing with the line immediately following it. |
| Syntax: | `-IF` *`expression`* `[THEN] GOTO` *`label1`*`; [ELSE GOTO` *`label2`*`;]`<br>`[ELSE IF...;]`<br><br>where:<br><br>*`label1...label2`*<br>Is a user-defined name of up to 12 characters that specifies the target of the GOTO action.<br>Do not use embedded blanks or the name of any other Dialogue Manager command except -QUIT or -EXIT. Do not use words that can be confused with functions, arithmetic or logical operations, and so on.<br><br>*`expression`*<br>Is a valid expression. Literals do not need to be enclosed in single quotation marks unless they contain embedded blanks or commas.<br><br>`THEN`<br>Is optional code that increases readability of the command.<br><br>`ELSE GOTO`<br>Passes control to *label2* when the -IF test fails.<br><br>`ELSE IF`<br>Specifies a compound -IF test.<br><br>The semicolon is required at the end of the command, and continuation lines must begin with a hyphen. |

| **Command:** | **-INCLUDE** |
|---|---|
| Function: | Enables one procedure to call another one. A procedure can call any number of other procedures. Up to four -INCLUDE commands can be nested.<br><br>The called procedure can contain fully executable or partial code. The calling procedure cannot branch to a label in the called procedure and vice versa.<br><br>**Note:** When calling the WebFOCUS CGI, the CGI uses an -INCLUDE command. Therefore, the end user can include only three nested -INCLUDE commands. When using edastart, this does not occur. |
| Syntax: | -INCLUDE *filename*<br><br>where:<br><br>*filename*<br>Is the name of the called procedure.<br>In UNIX, the value for *filename* must be an absolute path and file name. |

| **Command:** | **-label** |
|---|---|
| Function: | Is the target of a -GOTO command or -IF criteria. |
| Syntax: | *-label* [TYPE *message*]<br><br>where:<br><br>*label*<br>Is a user-supplied name of up to 12 characters that identifies the target for a branch.<br>Do not use embedded blanks or the name of any other Dialogue Manager command except -QUIT or -EXIT. Do not use words that can be confused with functions, arithmetic or logical operations, and so on.<br><br>TYPE *message*<br>Sends a message to the client application. |

| **Command:** | **-MVS** |
|---|---|
| Function: | In WebFOCUS, executes an OS/390 operating system command from a procedure. |
| Syntax: | `-MVS command`<br>where:<br>*command*<br>Is an OS/390 command. |

| **Command:** | **-PASS** |
|---|---|
| Function: | Directly issues and controls passwords. This feature is especially useful for specifying a particular file or set of files that a given user can read or write. Passwords have detailed sets of functions associated with them through the DBA facility.<br>The procedure that sets passwords can be encrypted so that it and the passwords that it sets cannot be typed and made known.<br>A variable can be associated with -PASS so that you can prompt for and assign a password value. |
| Syntax: | `-PASS password`<br>where:<br>*password*<br>Is a literal FOCUS password or a variable containing a password. |

| **Command:** | **-QUIT** |
|---|---|
| Function: | Forces an immediate exit from a procedure. Stacked commands are not executed.<br>If the procedure was called by another one, control returns directly to the client application, not to the calling procedure.<br>-QUIT can also be the target of a branch. |
| Syntax: | `-QUIT` |

| **Command:** | `-QUIT FOCUS` |
|---|---|
| Function: | In Developer Studio, terminates the procedure and exits FOCUS. The user returns to the operating system, or the calling program.<br><br>Upon exiting FOCUS, the value is returned to the calling program, if one exists. This is useful when FOCUS is called from another Windows application to return a status code indicating the state at which FOCUS was exited.<br><br>-QUIT FOCUS can also be made the target of a branch, with the same results as those already described. |
| Syntax: | `-QUIT FOCUS [n]`<br><br>where:<br><br>*n*<br>Is the application return code value. It can be a constant or an integer variable.<br>If you do not supply a value, or if you supply a non-integer value for *n*, the return code is 0 (the default value). |

| **Command:** | `-READ` |
|---|---|
| Function: | Enables the reading of data from an external file that is defined to the operating system.<br><br>The length of the variable list must be known before -READ is encountered. Use -DEFAULT to establish the number of characters expected for each variable.<br><br>If the list of variables is longer than one line, end the first line with a comma and begin the next line with a hyphen if you are reading a free-format file:<br><br>`-READ EXTFILE, &CITY, &CODE1,`<br>`- &CODE2`<br><br>If you are reading a fixed-format file, begin the next line with a hyphen and comma as follows:<br><br>`-READ EXTFILE &CITY.A8. &CODE1.A3.,`<br>`-, &CODE2.A3.` |
| Syntax: | `-READ` *`filename`*`[,] [NOCLOSE] &`*`name`*`[.`*`format`*`.][,]...`<br><br>where:<br><br>*`filename`*`[,]`<br>Is the name of an external file to read, which must be defined to the operating system. A space after *filename* denotes a fixed-format file, while a comma after *filename* denotes a free-format file.<br><br>NOCLOSE<br>Keeps the external file open until the -READ operation is complete. Files kept open with NOCLOSE can be closed using the command -CLOSE *filename*. The option NOCLOSE is available only in OS/390.<br><br>*`&name`*`[,]...`<br>Is a list of variables. For free-format files, you may but are not required to separate the variable names with commas.<br><br>*`.format.`*<br>Is the format of the variable. For free-format files, you do not have to define the length of the variable, but you may. For fixed-format files, the format specifies the length or the length and type of the variable (A is the default type). The value of *format* must be delimited by periods. |

| **Command:** | `-REPEAT` |
|---|---|
| Function: | Allows looping in a stored procedure. It is only available when operating under OS/390.<br><br>A loop ends when any of the following occurs:<br><br>• It is executed in its entirety.<br>• A -QUIT or -EXIT command is issued.<br>• A -GOTO is issued to a label outside of the loop. If a -GOTO is later issued to return to the loop, the loop proceeds from the point it left off. |

| | |
|---|---|
| Syntax: | `-REPEAT label n TIMES`<br>or<br>`-REPEAT label WHILE condition`<br>or<br>`-REPEAT label FOR &variable [FROM fromval] [TO toval] [STEP s]`<br>where:<br>*label*<br>Identifies the code to be repeated (the loop). A label can include another loop if the label for the second loop has a different name than the first.<br>*n* TIMES<br>Specifies the number of times to execute the loop. The value of n can be a local variable, a global variable, or a constant. If it is a variable, it is evaluated only once, so you cannot change the number of times to execute the loop. The loop can only be ended early using -QUIT or -EXIT.<br>WHILE *condition*<br>Specifies the condition under which to execute the loop. The condition is any logical expression that can be true or false. The loop is run if the condition is true.<br>FOR *&variable*<br>Is a variable that is tested at the start of each execution of the loop. It is compared with the value of *fromval* and *toval* (if supplied). The loop is executed only if *&variable* is less than or equal to *toval* when STEP is positive, or greater than or equal to *toval* when STEP is negative. If *toval* is not used, another option must be substituted, such as *n* TIMES.<br>FROM *fromval*<br>Is a constant that is compared with *&variable* at the start of each execution of the loop. The default value is 1.<br>TO *toval*<br>Is a value that is compared with *&variable* at the start of each execution of the loop. The default is 1,000,000.<br>STEP *s*<br>Is a constant used to increment *&variable* at the end of each execution of the loop. It may be positive or negative. The default value is 1.<br>**Note:** The parameters FROM, TO, and STEP can appear in any order. |

| **Command:** | **-RUN** |
|---|---|
| Function: | Causes immediate execution of all stacked commands.<br><br>Following execution, processing of the stored procedure continues with the line that follows -RUN.<br><br>-RUN is commonly used to do the following:<br><br>• Generate results from a request that can then be used in testing and branching.<br>• Close an external file opened with -READ or -WRITE. When a file is closed, the line pointer is placed at the beginning of the file for a -READ. The line pointer for -WRITE is positioned depending on the allocation and definition of the file. |
| Syntax: | `-RUN` |

| **Command:** | **-SET** |
|---|---|
| Function: | Assigns a literal value, or a value that is computed in an arithmetic or logical expression, to a variable.<br><br>Single quotation marks around a literal value are optional unless it contains an embedded blank, comma, or equal sign, in which case you must include them. |
| Syntax: | `-SET &[&]name= {expression\|value};`<br><br>where:<br><br>*&name*<br>Is the name of a variable whose value will be set.<br><br>*expression*<br>Is a valid expression. Expressions can occupy several lines, so end the command with a semicolon.<br><br>*value*<br>Is a literal value assigned to the variable. If the literal value contains commas or embedded blanks, you must enclose the value in single quotation marks. If the value contains a single quote, place two single quotes where you want one to appear. |

| **Command:** | **-TSO** |
|---|---|
| Function: | In WebFOCUS, executes a TSO operating system command from a procedure. |
| Syntax: | `-TSO command`<br>where:<br>*command*<br>Is a TSO command. |

| **Command:** | **-TYPE** |
|---|---|
| Function: | Sends a message to a client application.<br>Any number of -TYPE commands can follow one another, but each must begin with -TYPE.<br>Variables may be embedded in the message. The values currently assigned to each variable will be displayed. |
| Syntax: | `-TYPE text`<br>where:<br>*text*<br>Is a message that will be sent to a client application, followed by a line feed. If you include quotation marks around *text*, they will be displayed as part of the message.<br>The length of *text* can be up to 256 bytes. |

| **Command:** | **-UNIX** |
|---|---|
| Function: | In WebFOCUS, executes a UNIX operating system command from a procedure. |
| Syntax: | `-UNIX command`<br>where:<br>*command*<br>Is a UNIX command. |

| **Command:** | **-VMS** |
|---|---|
| Function: | In WebFOCUS, executes a VMS operating system command from a procedure. |
| Syntax: | -VMS *command*<br>where:<br>*command*<br>Is a VMS command. |

| **Command:** | **-WINNT** |
|---|---|
| Function: | Executes a Windows NT/2000/XP operating system command from a procedure. |
| Syntax: | -WINNT *command*<br>where:<br>*command*<br>Is a Windows NT/2000/XP command. |

| **Command:** | **-WRITE** |
|---|---|
| Function: | Writes data to an external file.<br><br>If the command continues over several lines, put a comma at the end of the line and a hyphen at the beginning of each subsequent line.<br><br>Unless you specify the NOCLOSE option, an opened file is closed upon termination of the procedure with -RUN, -EXIT, or -QUIT. |
| Syntax: | `-WRITE` *`filename`* `[NOCLOSE]` *`text`*<br><br>where:<br><br>*`filename`*<br>Is the name of a physical external file being written to. The *filename* must be known to the operating system.<br><br>`NOCLOSE`<br>Keeps the external file open until the -WRITE operation is complete. Files kept open with NOCLOSE can be closed with the command -CLOSE *filename*. The option NOCLOSE is available in OS/390.<br><br>*`text`*<br>Is any combination of variables and text. |

CHAPTER 11

# Testing and Debugging a Procedure

**Topics:**

- Debugging Your Application With Query Commands
- Displaying Combined Structures
- Displaying Virtual Fields
- Displaying the Currency Data Source in Effect
- Displaying Available Fields
- Displaying the File Directory Table
- Displaying Field Information for a Master File
- Displaying Data Source Statistics
- Displaying Current ddnames Assigned With FILEDEF
- Displaying Defined Functions
- Displaying HOLD Fields
- Displaying JOIN Structures
- Displaying National Language Support
- Displaying Explanations of Error Messages
- Displaying the Current Search Path
- Displaying the Release Number
- Displaying the Values of a Remote Server
- Displaying Parameter Settings
- Displaying Parameters That Cannot Be Set in an Area
- Displaying Graph Parameters
- Displaying Command Statistics
- Displaying StyleSheet Parameter Settings
- Displaying Information About the SU Machine
- Displaying Data Sources Specified With USE
- Displaying Global Variable Values
- Identifying the Files Being Used

You can debug your application by querying your environment or by performing tracing for your application from the Web Console. For details on tracing, see *Exploring Your WebFOCUS Development Environment* in the *WebFOCUS Application Development Getting Started* manual.

# Debugging Your Application With Query Commands

You can use query commands to display information about your metadata, data sources, language environment, Developer Studio, and WebFOCUS. This is useful for determining any aspects of your application that may be preventing the proper behavior of your application.

You can query your environment with a query command executed in Developer Studio or WebFOCUS.

The result of a query command is returned in your browser window or as a comment in the HTML file. Use the applicable Web browser functions to view the HTML comments (for example, View Source in Microsoft Internet Explorer).

*Syntax*

## How to Issue a Query Command

```
? query [filename]
```

where:

*query*

Is the subject of the query.

*filename*

Is the name of the file that is the subject of the query. This parameter applies to only some queries.

To list the query commands, type a question mark in a procedure.

*Reference*

## Query Command Summary

The following is a list of query commands. A detailed description of each is in this topic.

| | |
|---|---|
| `? COMBINE` | Displays a list of combined file structures. |
| `? DEFINE` | Displays currently active virtual fields created by the DEFINE command or attribute. |
| `? SET EUROFILE` | Displays the currency data source in effect. |
| `?F` | Lists fields currently available to you. |
| `? FDT` | Displays physical attributes of a FOCUS data source. |
| `?FF` | Lists field names, aliases, and format information for an active Master File. |

| | |
|---|---|
| `? FILE` | Displays the number of segment instances in a FOCUS data source and the last time the data sources was changed. |
| `? FILEDEF` | Displays the current ddnames assigned by the FILEDEF command. |
| `? FUNCTION` | Displays functions created with the DEFINE command. |
| `? HOLD` | Displays fields described in a HOLD Master File. |
| `? JOIN` | Displays JOIN structures that exist between data sources. |
| `? LANG` | Displays information about National Language Support. |
| `? LOAD` | Provides information about all loaded files: the file type, file name and resident size. |
| `? n` | Displays an explanation of an error message (*n* represents the number of the error message). |
| `? PATH` | Displays the current search path. |
| `? RELEASE` | Displays the release number of your product. |
| `? REMOTE` | Displays the values of the remote servers. |
| `? SET` | Displays parameter settings that control Developer Studio and WebFOCUS. |
| `? SET GRAPH` | Displays parameter settings that control graphs produced with the GRAPH command. |
| `? SET NOT` | Produces a list of SET commands that cannot be set in a specific area. |
| `? STAT` | Displays statistics about the last command executed. |
| `? STYLE` | Displays the current settings for StyleSheet parameters. |
| `? SU` | Is communication available to the SU machine. |
| `? USE` | Displays data sources specified with the USE command. |
| `? &&` | Displays values of global variables. |

# Displaying Combined Structures

The ? COMBINE command displays files that are in the current combined structures.

## *Syntax* How to Display Combined Structures

```
? COMBINE [filename]
```

where:

*filename*

Is the data source containing the virtual fields. If *filename* is omitted, the command displays all virtual fields.

## *Example* Displaying Combined Structures

Issuing the command

```
? COMBINE
```

produces information similar to the following:

```
FILE=EMPLJOB  COMBINED FROM

   EMPLOYEE
   JOBFILE
```

## *Reference* ? COMBINE Query Information

The list of file structures has the following form

```
FILE=combine_name COMBINED FROM
file_1
file_2
   .
   .
   .
file_n
```

where:

*combine_name*

Is the name of the COMBINE structure.

*file_1...file_n*

Are the names of FOCUS data sources that comprise the combined structure.

# Displaying Virtual Fields

The ? DEFINE command lists the active virtual fields used in a request. The fields can be created by either the DEFINE command or DEFINE attribute in the Master File. The command displays field names of up to 32 characters. If a name exceeds 32 characters, a symbol in the 32nd position indicates a longer field name. In WebFOCUS this symbol is an ampersand (&), and in Developer Studio this symbol is a caret (>).

## *Syntax* How to Display Virtual Fields

```
? DEFINE [filename]
```

where:

*filename*

Is the data source containing the virtual fields. If *filename* is omitted, the command displays all virtual fields.

## *Example* Displaying Virtual Fields

Assume that you created a virtual field named NEW_DATE in a request against the EMPLOYEE database.

Issuing

```
? DEFINE
```

produces the following information:

```
FILE     FIELD NAME                          FORMAT  SEGMENT   VIEW         TYPE
EMPLOYEE NEW_DATE                            YYMD          2
```

### *Reference* ? DEFINE Query Information

The following information is listed for each virtual field created with DEFINE:

| | |
|---|---|
| `FILE` | Is the name of the data source containing the virtual field. |
| `FIELD NAME` | Is the name of the virtual field. |
| `FORMAT` | Is the format of the virtual field. The notation is the same as that used for the FORMAT attribute in a Master File. |
| `SEGMENT` | Is the number of the segment in the Master File containing the virtual field. During reporting, your application treats the virtual field as a field in this segment. To relate segment numbers to segment names, use ? FDT. |
| `VIEW` | Is the root segment of DEFINE that specifies an alternate view. For example: `DEFINE FILE EMPLOYEE.JOBCODE` |
| `TYPE` | Indicates whether the virtual field is created by the DEFINE attribute in the Master File, or by a DEFINE command, identified by MASTER or a blank, respectively. |

## Displaying the Currency Data Source in Effect

The ? SET EUROFILE command displays the currency data source in effect.

### *Syntax* How to Display the Currency Data Source in Effect

To display the currency data source in effect, issue the command:

```
? SET EUROFILE
```

### *Example* Displaying the Currency Data Source in Effect

Issuing the command

```
? SET EUROFILE
```

produces information similar to the following:

```
EUROFILE                CURRCODE
```

# Displaying Available Fields

The ?F command displays the fields that are currently available.

?F displays entire 66 character field names.

## *Syntax* How to Display Available Fields

```
?F filename
```

where:

*filename*
Is the name of a data source.

## *Example* Displaying Available Fields

Issuing the command

```
?F EMPLOYEE
```

produces the following information:

```
FILENAME=  EMPLOYEE

EMPINFO.EMP_ID                LAST_NAME      FIRST_NAME     HIRE_DATE
DEPARTMENT     CURR_SAL       CURR_JOBCODE   ED_HRS

BANK_NAME      BANK_CODE      BANK_ACCT      EFFECT_DATE

DAT_INC        PCT_INC        SALARY         PAYINFO.JOBCODE

TYPE           ADDRESS_LN1    ADDRESS_LN2    ADDRESS_LN3    ACCTNUMBER

PAY_DATE       GROSS

DED_CODE       DED_AMT

JOBSEG.JOBCODE                JOB_DESC

SEC_CLEAR

SKILLS         SKILL_DESC

DATE_ATTEND    ATTNDSEG.EMP_ID

COURSE_CODE    COURSE_NAME
```

# Displaying the File Directory Table

The ? FDT command displays the file directory table, which lists the physical characteristics of a FOCUS data source.

A FOCUS data source is composed of fixed-length, 4096-byte records called pages. Each segment and each index (those fields designated by the keyword FIELDTYPE=I in the Master File) occupies an integral number of pages. The file directory table shows the amount of space occupied by each segment instance in a page, the starting and ending page numbers, and the number of pages in between for each segment and index.

***Syntax*** **How to Display a File Directory Table**

```
? FDT filename
```

where:

*filename*

Is the name of the data source.

***Example*** **Displaying a File Directory Table**

Issuing the command

```
? FDT EMPLOYEE
```

produces the following information:

```
     DIRECTORY:C:\ibi\WebFOCUS42\demos\...\employee.foc ON 04/18/2000 AT 15.47.24
     DATE/TIME OF LAST CHANGE:   12/30/1999   15.26.04
     SEGNAME   LENGTH   PARENT   START     END    PAGES  LINKS  TYPE
  1  EMPINFO       22               1       1       1       6
  2  FUNDTRAN      10       1       2       2       1       2
  3  PAYINFO        8       1       3       3       1       3
  4  JOBSEG        11       3                               4
  5  SECSEG         4       4                               2
  6  SKILLSEG      11       4                               2
  7  ADDRESS       19       1       4       4       1       2
  8  SALINFO        6       1       5       5       1       3
  9  DEDUCT         5       8       6       8       3       2
 10  ATTNDSEG       7       1                               3
 11  COURSEG       11      10                               2
```

### *Reference* ? FDT Query Information

The following information is listed in the file directory table:

| | |
|---|---|
| SEGNAME | Is the name of each segment in the file. The segments are also numbered consecutively down the left of the table. Unnumbered entries at the foot of the table are indexes, which belong to fields having the attribute FIELDTYPE=I in the Master File. |
| LENGTH | Is the length in words (units of four bytes) of each segment instance. Divide this number into 992 to determine the number of instances that can fit into a page. |
| PARENT | Is the parent segment. Each number refers to a segment name in the SEGNAME column. |
| START | Is the page number on which the segment or index begins. |
| END | Is the page number on which the segment or index ends. |
| PAGES | Is the number of pages occupied by the segment or index. |
| LINKS | Is the length, in words, of the pointer portion in each segment instance. Every segment instance consists of two parts, data and pointers. Pointers are internal numbers that are used to find other instances. |
| TYPE | Is the type of index. NEW indicates a binary index. OLD indicates a hash index. Segments of type KU, LM, DKU, DKM, KL, and KLU are not physically in this file; therefore, this information is omitted from the table. |

## Displaying Field Information for a Master File

The ?FF command displays field names, aliases, and format information for an active Master File.

## Syntax How to Display Field Information for a Master File

```
?FF filename [string]
```

where:

*filename*

Is the name of the Master File.

*string*

Is a character string up to 66 characters long. The command displays information only for fields beginning with the specified character string. If you omit this parameter, the command displays information for all fields in the Master File.

## Example Displaying Field Information for a Master File

Issuing the command

```
?FF EMPLOYEE
```

produces the following information:

```
FILENAME=  EMPLOYEE

EMPINFO.EMP_ID  EMPINFO.EID   A9
LAST_NAME       LN            A15
FIRST_NAME      FN            A10
HIRE_DATE       HDT           I6YMD
DEPARTMENT      DPT           A10
CURR_SAL        CSAL          D12.2M
CURR_JOBCODE    CJC           A3
ED_HRS          OJT           F6.2

BANK_NAME       BN            A20
BANK_CODE       BC            I6S
BANK_ACCT       BA            I9S
EFFECT_DATE     EDATE         I6YMD
```

# Displaying Data Source Statistics

The ? FILE command displays information such as the number of segment instances in a FOCUS data source and when the data source was last changed.

## *Syntax* How to Display Data Source Statistics

```
? FILE filename
```

where:

*filename*

Is the name of the data source.

## *Example* Displaying Data Source Statistics

Issuing the command

```
? FILE EMPLOYEE
```

produces statistics similar to the following:

```
STATUS OF FOCUS FILE: D:\ibi\apps\...\employee.foc ON 07/14/2000 AT 14.59.14
                ACTIVE  DELETED    DATE OF    TIME OF    LAST TRANS
SEGNAME         COUNT   COUNT      LAST CHG   LAST CHG     NUMBER

EMPINFO           12             06/14/2000   11.04.01
FUNDTRAN           6             03/31/2000   11.16.15        6
PAYINFO           19             06/05/2000   16.07.49
ADDRESS           21             03/31/2000   11.16.15       21
SALINFO           70             03/31/2000   11.16.16      448
DEDUCT           448             03/31/2000   11.16.16      448
TOTAL SEGS       576
TOTAL CHAR      8984
TOTAL PAGES        8
LAST CHANGE                      06/14/2000   11.04.01
```

## *Reference* ? FILE Query Information

The following data source statistics are listed:

| | |
|---|---|
| SEGNAME | Is the name of each segment in the data source. After the segments, the indexes are listed, if applicable.<br>Indexes are those fields specified by the attribute FIELDTYPE=I in the Master File. |
| ACTIVE COUNT | Is the number of instances of each segment. |

| | |
|---|---|
| `DELETED COUNT` | Is the number of segment instances deleted, for which the space is not reused. |
| `DATE OF LAST CHG` | Is the date on which data in a segment instance or index was last changed. |
| `TIME OF LAST CHG` | Is the time of day, on a 24-hour clock, when the file's last update was made for that segment or index. |
| `LAST TRANS NUMBER` | Is the number of transactions performed by the last update request to access the segment. If the data source was changed under Simultaneous Usage mode, this column refers to the REF NUMB column of the CR HLIPRINT file. |
| `TOTAL SEGS` | Is the total number of segment instances in the file (shown under ACTIVE COUNT), and the number of segments deleted when the file was last changed (shown under DELETED COUNT). |
| `TOTAL CHARS` | Is the number of characters of data in the file. |
| `TOTAL PAGES` | Is the number of pages in the data source. Pages are physical records in FOCUS data sources. Each page is 4096 bytes. |
| `LAST CHANGE` | Is the date and time the data source was last changed. |

If a data source is disorganized by more than 29%, that is, the physical placement of data in the data source is considerably different from its logical or apparent placement, the following message appears

```
FILE APPEARS TO NEED THE -REBUILD- UTILITY
REORG PERCENT IS A MEASURE OF FILE DISORGANIZATION
0 PCT IS PERFECT -- 100 PCT IS BAD
REORG PERCENT IS x%
```

where:

*x*

Is a percentage between 30 and 100.

The variable &FOCDISORG also indicates the level of disorganization. Following is an example of how you can use &FOCDISORG in a Dialogue Manager -TYPE command:

```
-TYPE THE AMOUNT OF DISORGANIZATION OF THIS FILE IS: &FOCDISORG
```

This command, depending on the amount of disorganization, produces a message similar to the following:

```
THE AMOUNT OF DISORGANIZATION OF THIS FILE IS: 10
```

When using a -TYPE command with &FOCDISORG, a message is displayed even if the percentage of disorganization is less than 30%.

## Displaying Current ddnames Assigned With FILEDEF

The ? FILEDEF command displays the ddnames assigned for various files, input and output.

During a session, depending on the types of operations you are doing, Developer Studio can create many FILEDEFs. Four of the system-created FILEDEFs are apparent using the ? FILEDEF command. You can have up to 250 FILEDEFs including both user- and system-created FILEDEFs. For related information see Chapter 16, *Defining and Allocating WebFOCUS Files*.

### *Syntax* How to Display Current ddnames

To display current ddnames, issue the command:

```
? FILEDEF
```

### *Example* Displaying Current ddnames

Issuing the command

```
? FILEDEF
```

produces information similar to the following:

```
Lname   Device  Lrecl Recfm Append Expl Filename
=================================================
HOLD2   DISK     0   V     N    Y C:\VM\SMALL\HOLD2.FTM
```

## Displaying Defined Functions

The ? FUNCTION command displays all defined functions and their parameters.

### *Syntax* How to Display DEFINE Functions

To display defined functions, issue the command:

```
? FUNCTION
```

### *Example* Displaying DEFINE Functions

Issuing the command

```
? FUNCTION
```

produces information similar to the following:

```
FUNCTIONS CURRENTLY ACTIVE
Name      Format    Parameter     Format
--------  --------  ------------  --------
SUBTRACT  D8.2      VAL1          D8
                    VAL2          D8
```

## Displaying HOLD Fields

The ? HOLD command lists fields described in a Master File created by the ON TABLE HOLD command. The list displays the field names, their aliases, and their formats as defined by the FORMAT (USAGE) attribute. The ? HOLD command displays field names up to 32 characters. If a field name exceeds 32 characters, a symbol in the 32nd position indicates a longer field name. In WebFOCUS, this symbol is an ampersand, and in Developer Studio the symbol is a caret (>).

The ? HOLD command displays fields of a HOLD Master File created by the current request.

### *Syntax* How to Display HOLD Fields

```
? HOLD [filename]
```

where:

*filename*

Is the name assigned in the AS phrase in the ON TABLE HOLD command. If you omit the file name, it defaults to HOLD.

### *Example* Displaying HOLD Fields

Issuing the command

```
? HOLD
```

produces information similar to the following:

```
DEFINITION OF HOLD FILE:
FIELDNAME                          ALIAS        FORMAT

LAST_NAME                          E01          A15
FIRST_NAME                         E02          A10
```

## Displaying JOIN Structures

The ? JOIN command lists the JOIN structures currently in effect. The command displays field names up to 12 characters. If a field name exceeds 12 characters, a symbol in the twelfth position indicates a longer field name. In WebFOCUS this symbol is an ampersand (&), and in Developer Studio the symbol is a caret (>).

### *Syntax* How to Display JOIN Structures

To display JOIN structures, issue the command:

```
? JOIN
```

## *Example* Displaying JOIN Structures

Issuing the command

```
? JOIN
```

produces information similar to the following:

```
  JOINS CURRENTLY ACTIVE

HOST                                CROSSREFERENCE
FIELD         FILE      TAG         FIELD         FILE      TAG       AS         ALL WH
-----         ----      ---         -----         ----      ---       --         --- --
JOBCODE       EMPLOYEE              JOBCODE       JOBFILE                         N   N
```

## *Reference* ? JOIN Query Information

The following JOIN information is listed:

| | |
|---|---|
| `HOST FIELD` | Is the name of the host field that is joining the data sources. |
| `FILE` | Is the name of the host data source. |
| `TAG` | Is a tag name used as a unique qualifier for field names in the host data source. |
| `CROSSREFERENCE FIELD` | Is the name of the cross-referenced field used to join the data sources. |
| `FILE` | Is the name of the cross-referenced data source. |
| `TAG` | Is a tag name used as a unique qualifier for field names in the cross-referenced data source. |
| `AS` | Is the name of the joined structure. |
| `ALL` | Displays Y for a non-unique join and N for a unique join. |
| `WH` | Specifies whether the join is a conditional join or an equi-join. |

# Displaying National Language Support

The ? LANG command displays information about National Language Support.

*Syntax* **How to Display Information About National Language Support**

To display information about National Language Support:

```
? LANG
```

*Example* **Displaying Information About National Language Support**

Issuing the command

```
? LANG
```

produces information similar to the following:

```
          NATIONAL LANGUAGE INFORMATION
Language                  001/AMENGLISH   (    )
Code Page                 437
Client Code Page          437
Dollar                    24($)
Lowcase alphabet          YES
Decimal notation          OFF(.)
Currency symbol           $
Date/Time format          EDA
NLS sort                  NO
NLS upcase/lowcase        NO
DBCS Flag                 OFF(SBCS)
```

# Displaying Explanations of Error Messages

The ? *n* command displays a detailed explanation of an error message, providing assistance in correcting the error.

Error messages generated by certain data adapters, such as the DB2 and MODEL 204 data adapters, are also accessible through this feature.

*Syntax* **How to Display Explanations of Error Messages**

```
? n
```

where:

*n*

Is the error message number.

*Example* **Displaying Explanations of Error Messages**

If you receive the message

```
(FOC125) RECAP CALCULATIONS MISSING
```

issuing the command

```
? 125
```

produces the following message:

```
(FOC125) RECAP CALCULATIONS MISSING
The word RECAP is not followed by a calculation.  Either the RECAP
should be removed, or a calculation provided.
```

# Displaying the Current Search Path

The ? PATH query displays your current search path. The search for files extends over several directories. The order that these directories are searched is determined for each user via the SET DEFAULT command and the values for the FOC$DIR logicals.

*Syntax* **How to Display the Current Search Path**

```
? PATH
```

*Example* **Displaying the Current Search Path**

Issuing the command

```
? PATH
```

produces information similar to the following if the WebFOCUS Reporting Server is using EDAPATH:

```
 EDAPATH  = D:\ibi\apps\ggdemo\
    D:\ibi\apps\ncp\
    D:\ibi\apps\template\
    C:\VM\SMALL\
TEMPDIR  = D:\ibi\srv52\wfs\edatemp\ts000019\
CURRDIR  = D:\ibi\srv52\wfs\edatemp\ts000019\
EDASYN   = D:\ibi\srv52\wfs\catalog\
EDASYNR  = D:\ibi\srv52\wfs\catalog\
```

If the WebFOCUS Reporting Server is using APP PATH, ? PATH produces information similar to the following:

```
 BASEAPP  =  C:\ibi\apps\baseapp;
APPHOLD  =  C:\ibi\srv52\wfs\edatemp\ts000001;
APPPATH  =  C:\ibi\apps\session;
            C:\ibi\apps\session;
```

# Displaying the Release Number

The ? RELEASE command displays the number of the currently installed release of your product.

*Syntax* **How to Display the Release Number**

To display the release number, issue the command:

```
? RELEASE
```

*Example* **Displaying the Release Number**

Issuing the command

```
? RELEASE
```

produces information similar to the following:

```
 EDA999 Release * Current Software:
EDA999 Release RELEASE = R729999B
EDA999 Release GEN_NUM = 000.223968
EDA999 Release SOURCE_DATE = 10/16/2002 18:13:18
EDA999 Release BUILD_DATE = 10/17/2002 02:59:24
EDA999 Release INSTALLATION_DATE= 10-17-2002 16:53:16
```

# Displaying the Values of a Remote Server

The ? REMOTE command displays information about remote servers. By issuing the ? REMOTE command while connected to a WebFOCUS server, you can obtain information about your remote destination, service, user ID, EDA node, CGI path, and the Web Server user ID and password.

For Developer Studio, the MR Base directory is also displayed. The MR Dir is a parameter that enables a Developer Studio administrator or developer to specify a WebFOCUS repository if multiple repositories reside on one Web Server. This setting overrides the WebFOCUS default.

*Syntax* **How to Display the Values of a Remote Server**

```
? REMOTE
```

### *Example* Displaying the Values of a Remote Server

Issuing the command

```
? REMOTE
```

produces information similar to the following:

```
  Remote Destination   --> LOOPBACK
  Remote User ID is not set.
  Remote User Password is not set.
 There are no conversations active.
```

## Displaying Parameter Settings

The ? SET command lists the parameter settings that control your Developer Studio and WebFOCUS environments. Your application sets default values for these parameters, but you can change them with the SET command.

SET parameters are described in Chapter 15, *Customizing Your Environment*.

### *Syntax* How to Display Parameter Settings

```
? SET [ALL|[FOR] parameter]
```

where:

`ALL`

Displays all possible parameter settings.

*`parameter`*

Is a SET parameter. This displays the setting for the specific parameter.

`FOR`

Includes where the parameter can be set from in addition to the parameter setting.

## *Example* Displaying Parameter Settings

Issuing the command

```
? SET
```

produces information similar to the following:

```
                              PARAMETER SETTINGS

ALL.                    OFF    FOCSTACK SIZE          8    PRINTPLUS            OFF
ASNAMES               FOCUS    FORMFEED             ASCII  QUALCHAR               .
AUTOINDEX                ON    HDAY                        QUALTITLES           OFF
AUTOPATH                 ON    HIPERFOCUS             OFF  REBUILDMSG          1000
BINS                     64    HOLDATTRS            FOCUS  RECAP-COUNT          OFF
BLKCALC                 NEW    HOLDLIST               ALL  SAVEMATRIX          OFF
BUSDAYS            _MTWTF_     HOLDSTAT               OFF  SCREEN              OFF
BYPANELING              OFF    HOTMENU                OFF  SHADOW PAGE         OFF
CACHE                     0    IBMLE                  OFF  SPACES             AUTO
CARTESIAN               OFF    INDEX TYPE             NEW  SQLENGINE
CDN                     OFF    LANGUAGE         AMENGLISH  SUMPREFIX           LST
COLUMNSCROLL            OFF    LINES/PAGE            66    TCPIPINT            OFF
DATEDISPLAY             OFF    LINES/PRINT           57    TEMP DISK D:\ibi\srv5..
DATEFNS                  ON    MESSAGE                ON   TERMINAL         IBM3270
DATETIME   STARTUP/RESET       MODE                WINNT   TESTDATE           TODAY
DEFCENT                  19    MORE                   OFF  TITLES               ON
EMPTYREPORT             OFF    MULTIPATH        COMPOUND   VIEWNAMESIZE          60
EXTAGGR                  ON    NODATA                  .   WIDTH                130
EXTHOLD                  ON    PAGE-NUM               ON   WINPFKEY            OLD
EXTSORT                 OFF    PANEL                   0   XRETRIEVAL            ON
FIELDNAME               NEW    PAUSE                  OFF  YRTHRESH              0
FOC2GIGDB                ON    POOL                   OFF
FOCALLOC                OFF    PRINT               ONLINE
```

Some parameters are listed differently from the way you specify them in the SET command. These include:

| | |
|---|---|
| `FOCSTACK SIZE` | Is the same as the FOCSTACK parameter. |
| `INDEX TYPE` | Is the same as the INDEX parameter. |
| `LINES/PAGE` | Is the same as the PAPER parameter. |
| `LINES/PRINT` | Is the same as the LINES parameter. |
| `SHADOW PAGES` | Is the same as the SHADOW parameter. |

### *Example* Displaying a Single Parameter Setting

Issuing the command

```
? SET ONLINE-FMT
```

produces the following if the parameter is set to its default value:

```
ONLINE-FMT                HTML
```

### *Example* Displaying Where a Parameter Can Be Set

Issuing the command

```
? SET FOR EXTSORT
```

produces the following information:

```
 EXTSORT                   OFF

-------------------------------------------------
SETTABLE FROM COMMAND LINE                  :     YES
SETTABLE ON TABLE                           :     YES
SETTABLE FROM SYSTEM-WIDE PROFILE           :     YES
SETTABLE FROM HLI PROFILE                   :     YES
POOL TABLE BOUNDARY                         :     YES
```

## Displaying Parameters That Cannot Be Set in an Area

The ? SET NOT command produces a list of SET commands that cannot be set in a specific area. The areas for which you can find this information are the PROMPT command, report requests, the FOCPARM profile, the HLI profile, and Pooled Tables.

### *Syntax* How to Determine Where a Command Is Valid

```
? SET NOT area
```

where:

*area*

Is one of the following:

`PROMPT` is the PROMPT command.

`ONTABLE` is a report request.

`FOCPARM` is the FOCPARM profile.

`HLIPROF` is the HLI profile.

`PT` is in Pooled Tables.

### *Example* Displaying Parameters That Cannot Be Set With ONTABLE

Issuing the command

```
? SET NOT ONTABLE
```

produces the following information:

```
                    NON-SETTABLE ON TABLE PARAMETER SETTINGS

BINS                  64    FOC2GIGDB             ON    PRINTPLUS            OFF
BLKCALC              NEW    FOCSTACK SIZE          8    RADIOBUTTON          OFF
BYPANELING           OFF    FORMFEED           ASCII    REBUILDMSG          1000
CACHE                  0    HTMLMODE             OFF    SAVEMATRIX           OFF
CGI-RELEASE         5200    IFSORT                ON    SMARTMODE           ????
CODEBASE           ????    LANGUAGE       AMENGLISH    SORTNAME           XSBEG
COLUMNSCROLL         OFF    MAXPOOLMEM         32768    TALKCASE             OFF
DATEDISPLAY          OFF    MDIBINS             8000    TCPIPINT             OFF
DATEFNS               ON    MDIPROGRESS       100000    TEMP DISK D:\ibi\srv5..
DATEFORMAT           MDY    MODE               WINNT    TRMSD                  0
DEFCENT               19    MPRINT               NEW    TRMSW                  0
DEFINES              OLD    NLSCTRL              OFF    TRMTYP          64  64\s
DISTRIBUTE           OFF    POOL                 OFF    WEBHOME            32768
DTSTRICT              ON    POOLBATCH            OFF    WEEKFIRST              7
ERROROUT             OFF    POOLFEATURE          OFF    WIDTH                130
EUROFILE                    POOLMEMORY         16384    WINPFKEY             OLD
FIELDNAME            NEW    POOLRESERVE          100    YRTHRESH               0
```

## Displaying Graph Parameters

The ? SET GRAPH command lists the parameter settings that control graphs produced with the GRAPH command. These parameters are described further in Chapter 15, *Customizing Your Environment*.

### *Syntax* How to Display Graph Parameters

To display graph parameters, issue the command:

```
? SET GRAPH
```

### *Example* Displaying Graph Parameters

Issuing the command

```
? SET GRAPH
```

produces information similar to the following:

```
                    GRAPH PARAMETER SETTINGS

3DGRAPH          ON   GTREND          OFF   PIE              OFF
AUTOTICK         ON   GVIEWERMAX      OFF   TERMINAL     IBM3270
BARNUMB         OFF   GVIEWERPAUSE     ON   VAUTO            ON
BARSPACE          0   HAUTO            ON   VAXIS            66
BARWIDTH          1   HAXIS           130   VCLASS          .00
BSTACK          OFF   HCLASS          .00   VGRID           OFF
FORMATGRAPH    HTML   HISTOGRAM        ON   VMAX            .00
GMISSING        OFF   HMAX           .00   VMIN            .00
GMISSVAL        .00   HMIN           .00   VTICK           .00
GPROMPT         OFF   HSTACK         OFF   VZERO           OFF
GRIBBON         OFF   HTICK          .00
GRID            OFF   LOOKGRAPH     NONE
```

## Displaying Command Statistics

The ? STAT command lists statistics for the most recently executed command.

Each statistic applies only to a certain command. If another command is executed, the statistic is either 0 or does not appear in the list at all. When you execute commands in stored procedures, these statistics are automatically stored in Dialogue Manager statistical variables. For more information, see Chapter 10, *Managing Flow of Control in an Application*.

### *Syntax* How to Display Command Statistics

To display command statistics, issue the command:

```
? STAT
```

### *Example* Displaying Command Statistics

Issuing the command

```
? STAT
```

produces information similar to the following:

```
                    STATISTICS OF LAST COMMAND
RECORDS        =          0       SEGS DELTD     =          0
LINES          =          0       NOMATCH        =          0
BASEIO         =         47       DUPLICATES     =          0
SORTIO         =          0       FORMAT ERRORS  =          0
SORT PAGES     =          0       INVALID CONDTS =          0
READS          =          0       OTHER REJECTS  =          0
TRANSACTIONS   =          0       CACHE READS    =          0
ACCEPTED       =          0       MERGES         =          0
SEGS INPUT     =          0       SORT STRINGS   =          0
SEGS CHNGD     =          0       INDEXIO        =          0

INTERNAL MATRIX CREATED: YES       AUTOINDEX USED:          NO
SORT USED:             FOCUS       AUTOPATH USED:           NO
AGGREGATION BY EXT.SORT:  NO       HOLD FROM EXTERNAL SORT:  NO
```

### *Reference* ? STAT Query Information

The following information displays:

## Displaying StyleSheet Parameter Settings

The ? STYLE command displays the current settings for StyleSheet parameters.

### *Syntax* How to Display StyleSheet Parameter Settings

```
? [SET] STYLE
```

### *Example* Displaying StyleSheet Parameter Settings

Issuing the command

```
? STYLE
```

produces information similar to the following:

```
                    STYLESHEET PARAMETER SETTINGS

ONLINE-FMT     HTML
OFFLINE-FMT    STANDARD
STYLESHEET     ON
SQUEEZE        OFF
LABELPROMPT    OFF
STYLEMODE      FULL
ORIENTATION    PORTRAIT
UNITS          INCHES
TOPMARGIN      .250
BOTTOMMARGIN   .250
LEFTMARGIN     .250
RIGHTMARGIN    .250
TARGETFRAME    ????
FOCEXURL       /cgi-bin/ibi_cgi/ibiweb.exe?IBIMR_drill=X,corporat/corporat.htm&
BASEURL        OFF
```

## *Reference* ? STYLE Query Information

The following StyleSheet information is listed:

| | |
|---|---|
| ONLINE-FMT | Is the format of report output. Can produce output as HTML or as un-styled character-based output. |
| OFFLINE-FMT | Is the format of report output. Can produce output as HTML or as un-styled character-based output. |
| STYLESHEET | Rejects or accepts StyleSheet parameters that specify formatting options such as page size, orientation, and margins. |
| LABELPROMPT | Specifies on which label of the first page to begin printing a multi-pane report, such as a mailing label report. |
| STYLEMODE | Speeds the retrieval of large report output by displaying output in multiple HTML tables where each table is a separate report page. |
| ORIENTATION | Is the page orientation for styled reports. Can be either portrait or landscape. |
| UNITS | Is the unit of measure for PostScript and PDF report output, as either inches, centimeters, or points. |
| TOPMARGIN | Is the top boundary for a page of report output. |
| BOTTOMMARGIN | Is the bottom boundary for a page of report output. |
| LEFTMARGIN | Is the left boundary for a page of report output. |

| | |
|---|---|
| `RIGHTMARGIN` | Is the right boundary for a page of report output. |
| `TARGETFRAME` | Is a frame to which all drill-down hyperlinks are directed. |
| `FOCEXURL` | Is the location of a procedure and the method used to execute it from the WebFOCUS Client. |
| `BASEURL` | Is the default location where the browser searches for relative URLs specified in the HTML documents created by your application. |

## Displaying Information About the SU Machine

The ? SU command displays the communication available to the FOCUS Database Server.

### *Syntax* How to Display Information About the FOCUS Database Server

`? SU [userid|ddname]`

where:

*userid*

Is a sink machine user ID.

*ddname*

Is a valid ddname.

### *Example* Displaying Information About the FOCUS Database Server

Issuing the command

```
? SU SYNCA
```

produces the following information:

```
USERID          FILEID     QUEUE
WIBMLH          QUERY
WIBJBP          CAR
```

## Displaying Data Sources Specified With USE

The ? USE command displays data sources specified with the USE command.

### *Syntax* How to Display Data Sources Specified With USE

To display data sources specified with the USE, issue the command:

```
? USE
```

### *Example* Displaying Data Sources Specified With USE

Issuing the command

```
? USE
```

produces information similar to the following:

```
 DIRECTORIES IN USE ARE:
C:\IBI\apps\ibisamp\CENTINV.FOC AS CENTINV
C:\IBI\apps\ibisamp\CENTHR.FOC AS CENTHR
```

# Displaying Global Variable Values

The ? && command lists Dialogue Manager global variables and their current values. Global variables maintain their values for the duration of the stored procedure.

See your documentation about Dialogue Manager for details on global and other variables.

*Syntax* **How to Display Global Variable Values**

```
? &&
```

Your site may replace the ampersand (& or &&) indicating Dialogue Manager variables, with another symbol. In that case, use the replacement symbol in your query command. For example, if your installation uses the percent sign (%) to indicate Dialogue Manager variables, list global variables by issuing:

```
? %%
```

You can query all Dialogue Manager variables (local, global, and system) from a stored procedure by issuing:

```
-? &
```

*Example* **Displaying Global Variable Values**

Issuing the command

```
? &&
```

produces information similar to the following:

```
  &&CITY        'STAMFORD'
 &&CODE1        'B10'
 &&CODE2        'B20'
```

# Identifying the Files Being Used

You can debug your application by identifying the files you are using. You can identify the file name and location of a procedure, data source, Master File, or StyleSheet. WebFOCUS searches on the path of the WebFOCUS Reporting Sever and identifies the file with one of the following WHENCE commands:

- WHENCE *fexname* FOCEXEC identifies the procedure being used.
- WHENCE *focusfile* FOCUS identifies the FOCUS data source being used.
- WHENCE *masterfile* MASTER identifies the Master File being used.
- WHENCE *accessfile* ACCESS identifies the Access File being used.
- WHENCE *stylesheet* FOCSTYLE identifies the StyleSheet being used.

### *Syntax* How to Query the Procedure Being Used

```
WHENCE procedure FOCEXEC
```

where:

*procedure*
Is a procedure.

### *Example* Querying the Procedure Being Used

Issuing the command

```
WHENCE FINANCE FOCEXEC
```

produces output similar to the following:

```
C:\ibi\apps\session\layout.fex
```

### *Syntax* How to Query the FOCUS Data Source Being Used

```
WHENCE focusfile FOCUS
```

where:

*focusfile*
Is a FOCUS data source.

### *Example* Querying the FOCUS Data Source Being Used

Issuing the command

```
WHENCE CENTINV FOCUS
```

produces output similar to the following:

### *Syntax* How to Query the Master File Being Used

```
WHENCE masterfile MASTER
```

where:

*masterfile*
Is a Master File.

### *Example* Querying the Master File Being Used

Issuing the command

```
WHENCE CENTINV MASTER
```

produces output similar to the following:

### *Syntax* How to Query the Access File Being Used

```
WHENCE accessfile ACCESS
```

where:

*accessfile*
Is an Access File.

### *Example* Querying the Master File Being Used

Issuing the command

```
WHENCE GKEIVP ACCESS
```

produces output similar to the following:

### *Syntax* How to Query the StyleSheet Being Used

```
WHENCE stylesheet FOCSTYLE
```

where:

*stylesheet*
Is a StyleSheet.

### *Example* Querying the StyleSheet Being Used

Issuing the command

```
WHENCE GGORDERS FOCSTYLE
```

produces output similar to the following:

```
C:\ibi\apps\session\classic.sty
```

CHAPTER 12

# Partitioning and Deploying Project Files

**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)

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. 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.

# Deployment Basics

Once you create project files as described in Chapter 3, *Creating a Reporting Application*, 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

**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.
- Step 2. Create a deployment scenario.
- Step 3. Define the partitioning of the project files in the scenario.
- Step 4. Move the project files to a Web server and WebFOCUS Reporting Server as defined in the scenario.

You can optionally configure the target servers for your site. For details, see *Configuring the Target Servers (Optional)* on page 12-14.

## 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 Chapter 13, *Managing the WebFOCUS Environment*, in this manual, and the *Developer Studio Installation and Configuration* manual.

## Step 2. Create a Deployment Scenario

A deployment scenario provides a means of identifying project files for distribution to target servers in a WebFOCUS environment.

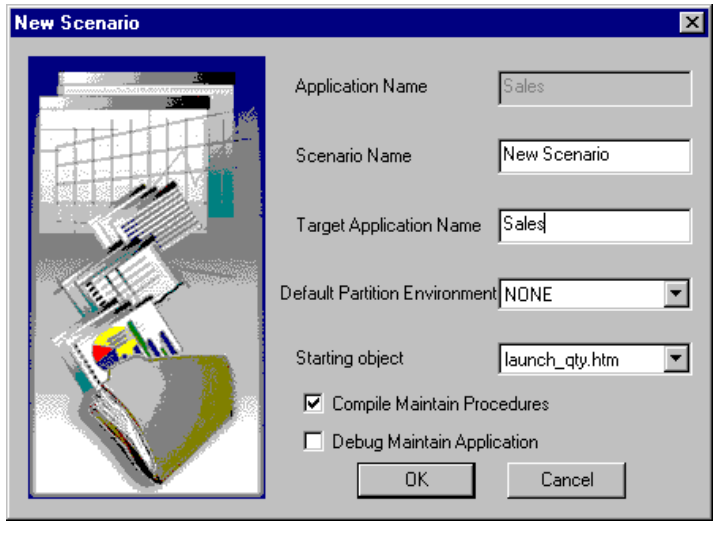
You can create more than one deployment scenario for a project, and you can view and modify the properties of an existing scenario.

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. Select *Deploy* from the shortcut menu, then select *New Scenario*.

   The New Scenario dialog box opens:



**2.** Respond as follows:

- **Scenario Name.** Enter a name for the new deployment scenario. You can change the name at a later time with the Rename option.
- **Target Application Name.** 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 a WebFOCUS Reporting Server on OS/390 or MVS, 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.
- **Default Partition Environment.** From the drop-down list, you can 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 Maintain files) are partitioned to the default WebFOCUS Reporting Server.

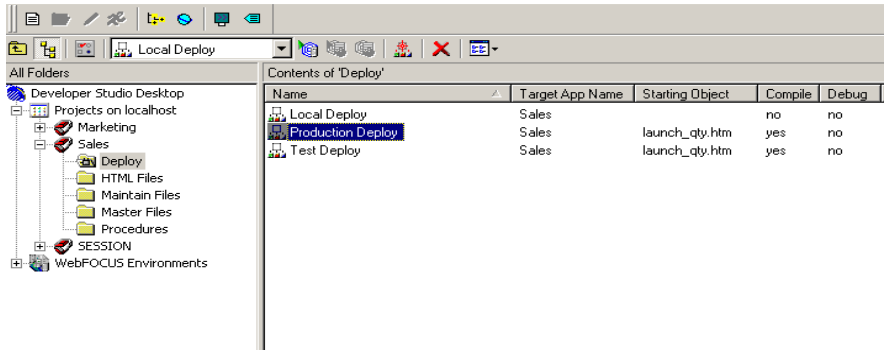
  For an example, see *Working in a Default Partition Environment* on page 12-8.

  Accept the value NONE to suppress the use of a default environment and automatic partitioning of files.

  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.
- **Starting Object.** From the drop-down list, select the object that will execute first when a user runs the application. Typically the starting object is an HTML page. All files that are valid starting objects appear in the list.
- **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.
- **Debug Maintain Application.** This option is reserved for debugging purposes. It can affect performance. For more information, see the *Developing WebFOCUS Maintain Applications* manual.

Click *OK* when you are done. You have now created a new deployment scenario.

3. 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 and Modify Deployment Scenario Properties

1. Select and right-click the name of the deployment scenario and select *Properties* from the shortcut menu. The Scenario Properties dialog box opens.

2. You can change the properties as desired. For example, you can change the name of the target application or the starting object. You can also modify the values for your Maintain applications.

## Step 3. Partition the Project 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.

## *Procedure* How to Partition the Project Files

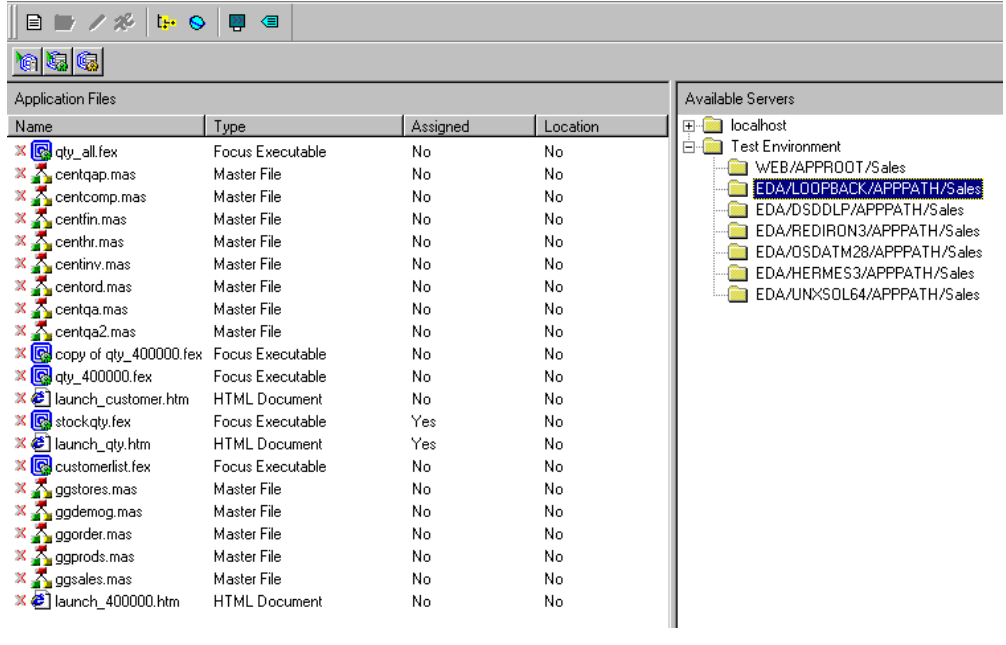
1. Open the DEPLOY folder for the project. Select and right-click the deployment scenario that you are defining, and select *Open*.
2. The left window pane displays information about the individual files that have been added to the project.

   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.
3. The right window pane displays the environments you have identified and optionally configured to accept 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.

   **b.** Select *New Folder* from the shortcut menu. At the cursor location in the entry field, enter the name of the new folder.
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, a development server, a deployment target server, and four alternate server nodes for the deployment server:



In the above example, 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 12-8.

**5.** Drag one or more files from the left window pane and drop them into the right window pane on the applicable server.

To remove a partitioned file, right-click it and select *Delete*.

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.

**Tip:** Typically you will deploy HTML, GIF, JPG, VBScript, and JavaScript files to a Web server. Executable WebFOCUS procedures (.FEX and Maintain files) will go to a WebFOCUS Reporting Server. Do not deploy Master Files (.MAS), Access Files (.ACX), or proprietary FOCUS data sources (.FOC). Master Files, Access Files, and FOCUS data sources already reside on the deployment WebFOCUS Reporting Server or subserver in applications configured to contain them.

**For files containing the command -HTLMFORM filename:**

- If you are deploying a procedure that contains -HTMLFORM *filename* (including procedures created with the Resource Layout Tool), 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.

- In the case of 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.

**6.** On the toolbar, three buttons allow you to do the following while you remain in the deployment scenario window:

*Deploy the current scenario.* Deploys (moves) the files to the servers based on the partitioning defined in the current deployment scenario.

*Deploy and run the current scenario.* Deploys the files to the servers and then runs the starting object defined in the current deployment scenario.

*Run the current deployment scenario.* Runs the starting object defined in the current deployment scenario.

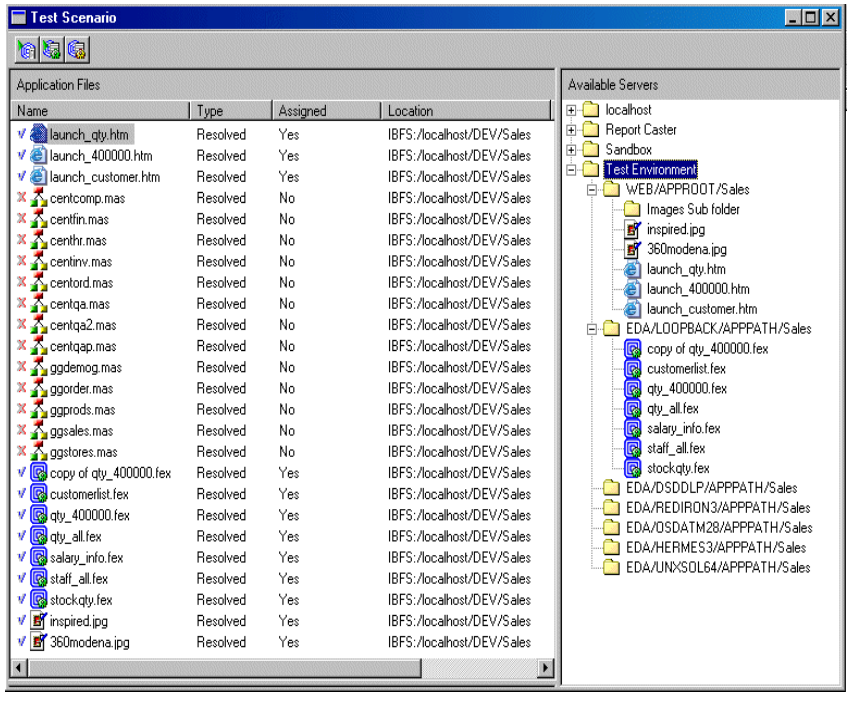
These buttons are also available on the toolbar in the Explorer.

7. When you have finished partitioning files, choose one of the options in step 6, or close the window. You can perform any of the options in step 6 outside the deployment scenario window. See *Step 4. Deploy the Project Files* on page 12-9.

## Working in a Default Partition Environment

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.




### *Reference* File Status Icons

The icons in the first column to the left of the file in the Name field display the status of the file.

| Icon | Definition |
|---|---|
| | The file has been partitioned. |
| | The file has not been partitioned. |

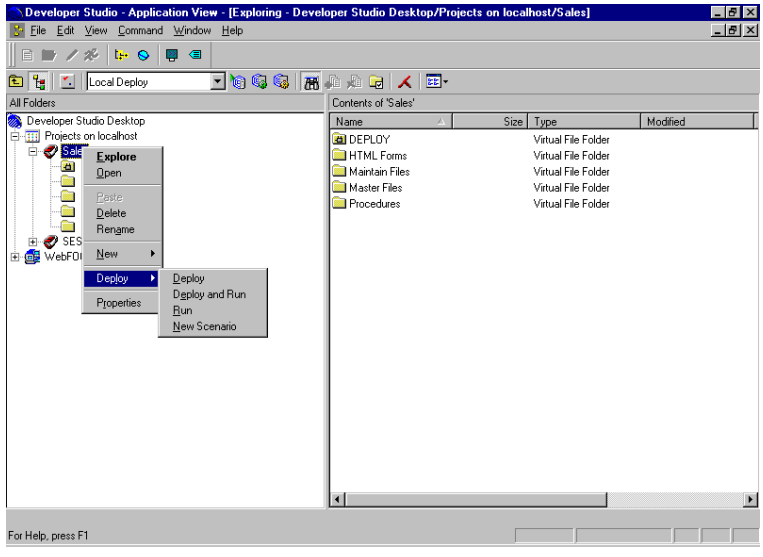
## Step 4. Deploy the Project Files

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 deployment scenario window. See *Step 3. Partition the Project Files* on page 12-5. The buttons on the toolbar in the deployment scenario window are also available in the Explorer.

***Procedure*** **How to Deploy Project Files in the Explorer Using the Shortcut Menu**

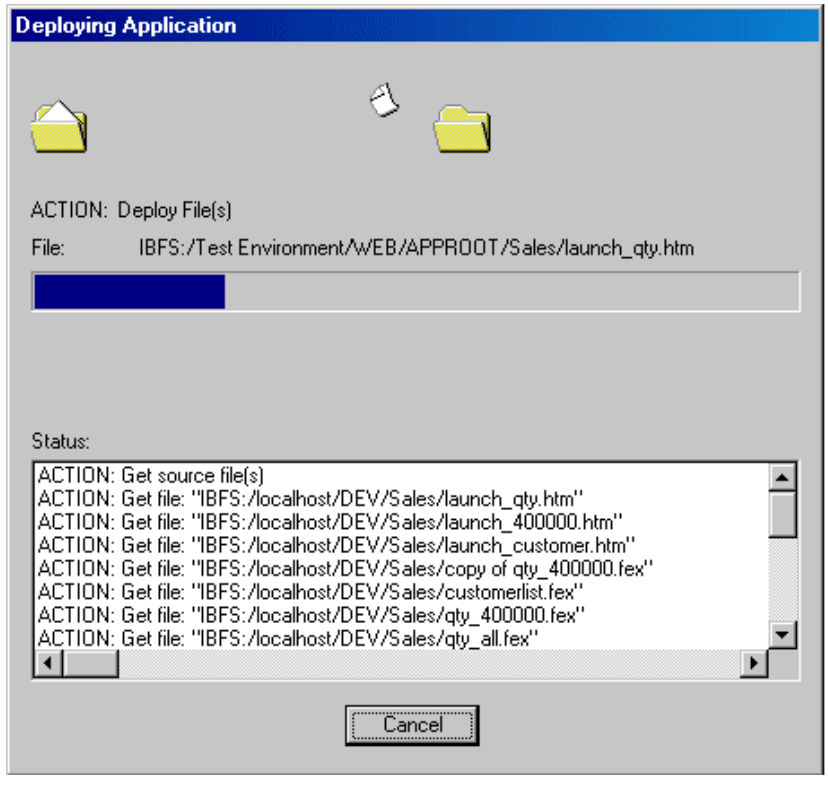
1. Select the project. From the drop-down list on the toolbar, select the deployment scenario.
2. Right-click the project. Select *Deploy* from the shortcut menu. Then select:
   - *Deploy* to move the files to the servers based on the partitioning defined in the deployment scenario.
   - *Deploy and Run* to move the files to the servers and then run the starting object defined in the deployment scenario.




The option Run on the shortcut menu runs the starting object defined in the deployment scenario. In this example, the scenario is Local Deploy, as shown in the drop-down list.

**3.** During deployment, a dialog box tracks the status. You can click *Cancel* to terminate the process.

The following window shows successful deployment of HTML files to the Web server and procedures to the WebFOCUS Reporting Server:




After deployment, the ACTION field indicates that one of the following occurred:

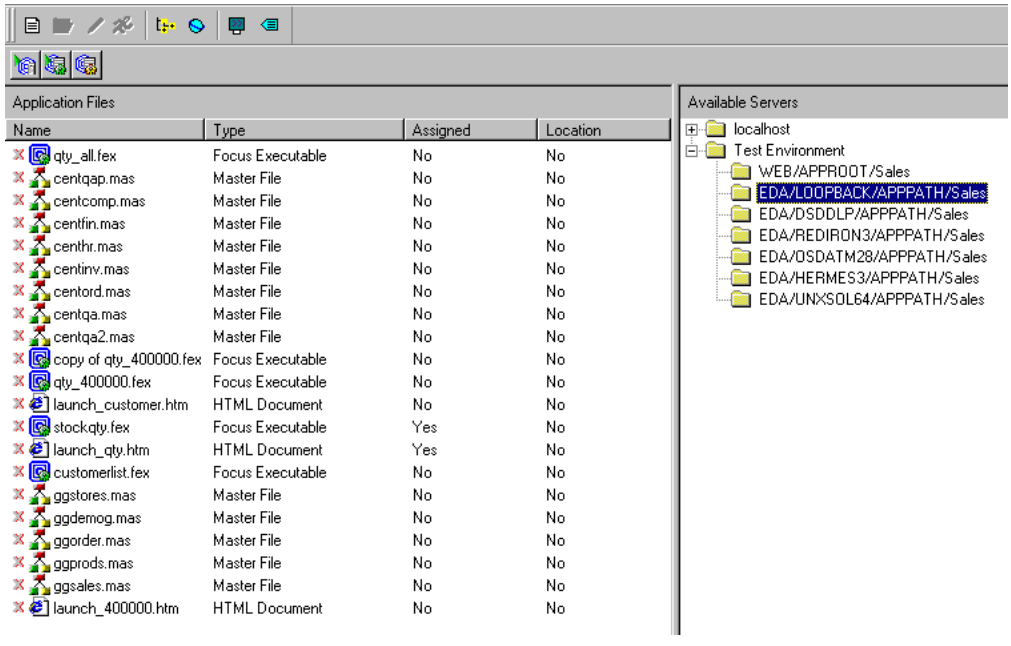
- Deployment completed successfully.
- Deployment failed.
- Deployment was canceled.

During the deployment process, the File field shows the names of objects as they are deployed to a server.

The progress bar indicates what percentage of the process is complete, as the process takes place.

The Status field displays the events that took place in the deployment process.

## *Reference* Partition Project Files Window




### Project Files

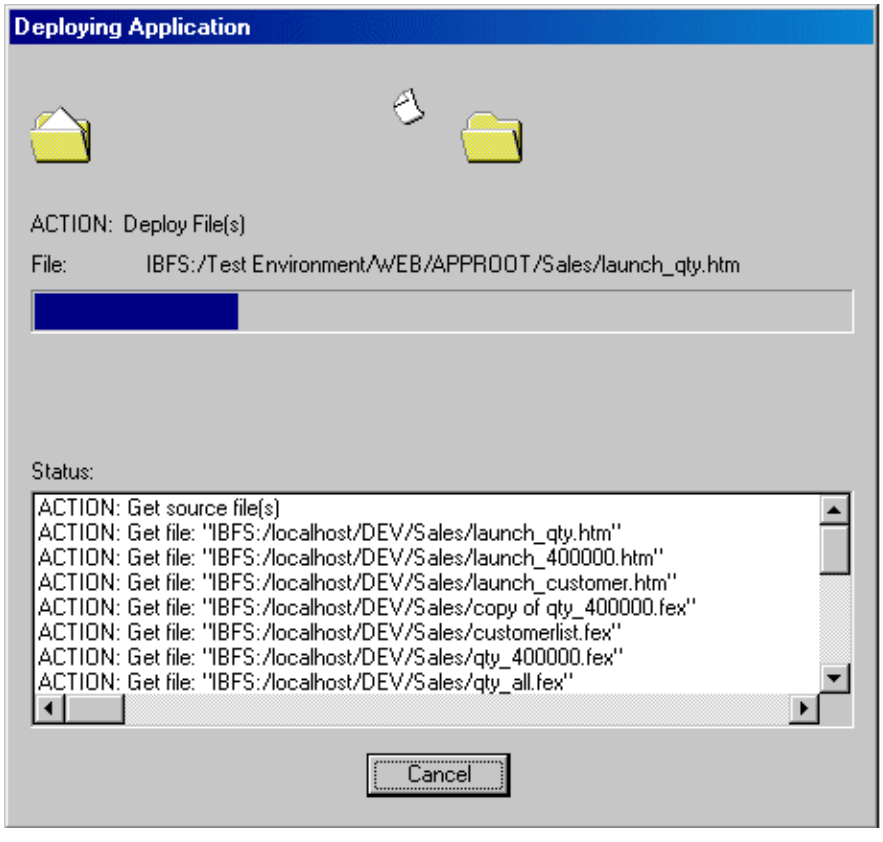
Displays the files in the selected project and their properties.

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.

## *Reference* Deploying Application Dialog Box




**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 completed successfully.
  - Deployment failed.
  - Deployment was canceled.
- During the deployment process, the File field shows the names of objects as they are deployed to the server.
- The progress bar indicates what percentage of the process is complete, as the process takes place.
- The Status field displays the events that took place in the deployment process.

# Configuring the Target Servers (Optional)

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 Client Console and the 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.

*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:

| Type of Installation | Location of cgivars.wfs |
|---|---|
| Developer Studio with localhost | *install_drive*:\ibi\srv52\wfs\etc |
| WebFOCUS | *install_drive*:\ibi\client52\wfc\etc |

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:

| Type of Installation | Location of edaserve.cfg |
|---|---|
| Developer Studio with localhost | *install_drive*:\ibi\srv52\wfs\bin |
| WebFOCUS | *install_drive*:\ibi\srv52\wfc\bin |

## *Procedure* How to Set the Web Server Deployment Path Using the WebFOCUS Client Console (Windows NT/2000/XP and UNIX)

1. Access the WebFOCUS Client Console:
   - From the Start menu, choose *Programs*, *WebFOCUS Developer Studio 5.2*, *WebFOCUS Developer Studio Utilities*, and *WebFOCUS Client Console*.
   - Enter your user ID and password as prompted and click *Logon*.
2. Select *Customize Settings* from the left pane. Select *Change Client Defaults*.
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 *The APPROOT Alias* on page 12-18 and *How to Create an Alias on the Web Server (Windows 2000)* on page 12-18.

   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 12-5.
4. Click *Save* and exit the WebFOCUS Client Console.

## *Procedure* How to Set the WebFOCUS Reporting Server Deployment Path Using the Server Console (Windows NT/2000/XP)

1. Access the Server Console:
   - From the Start menu, choose *Programs*, *WebFOCUS Server 5.2*, and *Web Console*.

   or
   - In Developer Studio, choose *Server Console* from the Command menu.
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 Server Console.

## *Procedure* How to Set the Web Server and WebFOCUS Reporting Server Deployment Path Using a Text Editor (Windows NT/2000/XP 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 *The APPROOT Alias* on page 12-18 and *How to Create an Alias on the Web Server (Windows 2000)* on page 12-18.

   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 12-5.
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 (OS/390)

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.V5R2M00.APP
```

For more information on the APPSNS parameter, see the *iWay Server for MVS Configuration and Operations Manual* and the *iWay Server Administration for UNIX, Windows, OpenVMS, OS/400, OS/390, and z/OS Manual*.

If you are using WebFOCUS Reporting Server for OS/390 for deployment, the base application files are allocated during configuration (APPSNS=YES).

### *Example* Allocating Base Application Files (OS/390)

The following is an example of  base files allocated for the application:

```
EDAYHC.V5R2M00.APP.BASEAPP.ACCESS.DATA
EDAYHC.V5R2M00.APP.BASEAPP.FOCEXEC.DATA
EDAYHC.V5R2M00.APP.BASEAPP.FOCSTYLE.DATA
EDAYHC.V5R2M00.APP.BASEAPP.GIF.DATA
EDAYHC.V5R2M00.APP.BASEAPP.HTML.DATA
EDAYHC.V5R2M00.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 *xxx*BIN/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* The 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, you can choose automatic configuration of the 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*. 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 NT/2000/XP 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:

| Type of Installation | Location of ibiweb.cfg |
|---|---|
| Developer Studio with localhost | *install_drive*:\ibi\srv52\wfs\web\cgi |
| WebFOCUS | *install_drive*:\ibi\client52\wfc\web\cgi |

## *Procedure* How to Restrict Authorization to Deploy Files Using the WebFOCUS Client Console

1. Access the WebFOCUS Client Console:
   - From the Start menu, choose *Programs*, *WebFOCUS Developer Studio 5.2, WebFOCUS Developer Studio Utilities*, and *WebFOCUS Client Console*.
   - Enter your user ID and password as prompted and click *Logon*.
2. The WebFOCUS Client Configurator opens. Select *Configuration* from the options on the left.
3. Select *Change/Add Default Configuration*.
4. On the ibiweb 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 Client Console.

## *Procedure* How to Restrict Authorization to Deploy Files Using a Text Editor

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.

# Developer's Reference

CHAPTER 13

# Managing the WebFOCUS Environment

**Topic:**

- Adding an Environment
- Editing Environment Properties
- Working With WebFOCUS Environment Properties
- Setting the Development Environment
- Refreshing or Removing an Environment

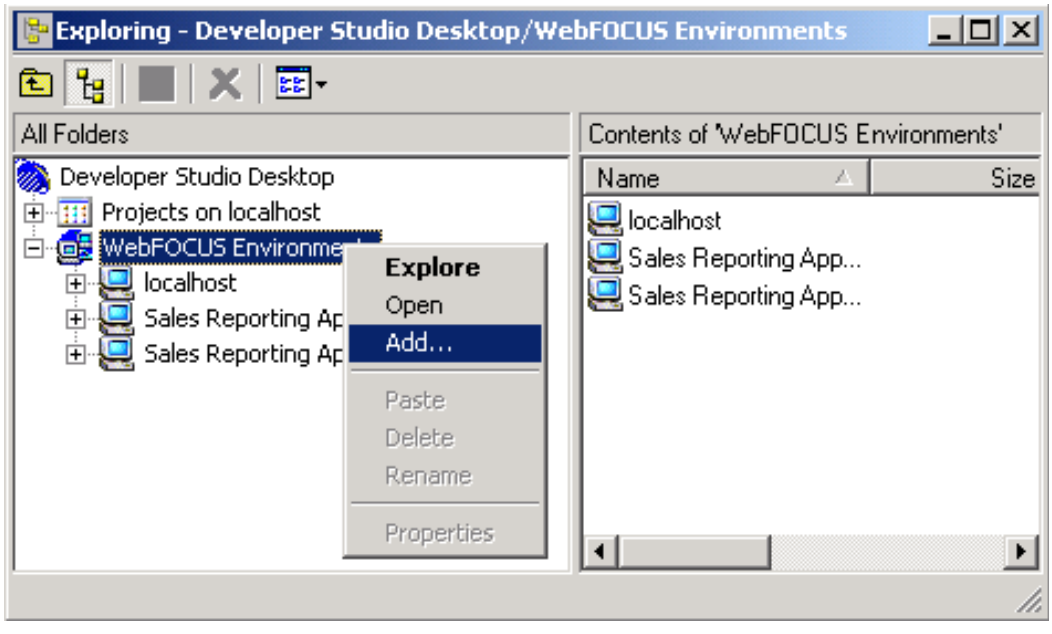
You define and manage your WebFOCUS environments from the WebFOCUS Environment Properties dialog box. You access this dialog box from either the WebFOCUS Environments node in the Explorer window or from the Environments List dialog box.

# Adding an Environment

You add a WebFOCUS environment from either the WebFOCUS Environments node in the Explorer window or from the Environments List dialog box.

*Procedure* **How to Add an Environment**

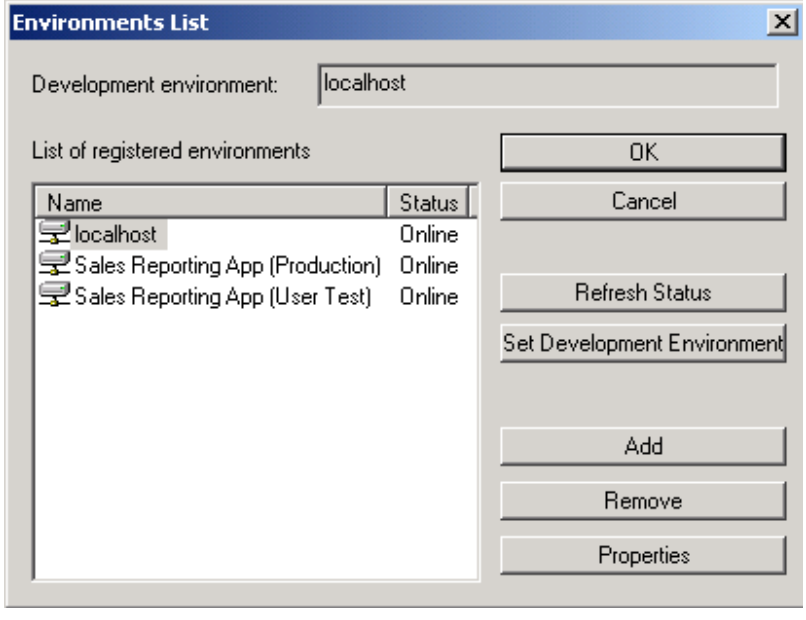
Right-click the WebFOCUS Environments node in the Explorer window and select *Add*.




The WebFOCUS Environment Properties dialog box opens. See *Working With WebFOCUS Environment Properties* on page 13-4 for details.

OR

1. Select *Show Environments* from the File menu. The Environments List dialog box opens.




2. Click *Add*.

   The Environment Properties dialog box opens.

3. Edit the environment properties.

4. Click *OK*.

# Editing Environment Properties

You can edit existing environment properties from the WebFOCUS Environment Properties dialog box.

*Procedure* **How to Edit Environment Properties**

Right-click the environment in the Explorer window and select *Properties*.

OR

1. Select *Show Environments* from the File menu. The Environments List dialog box opens.
2. Select the environment whose properties you want to change, and click *Properties*.

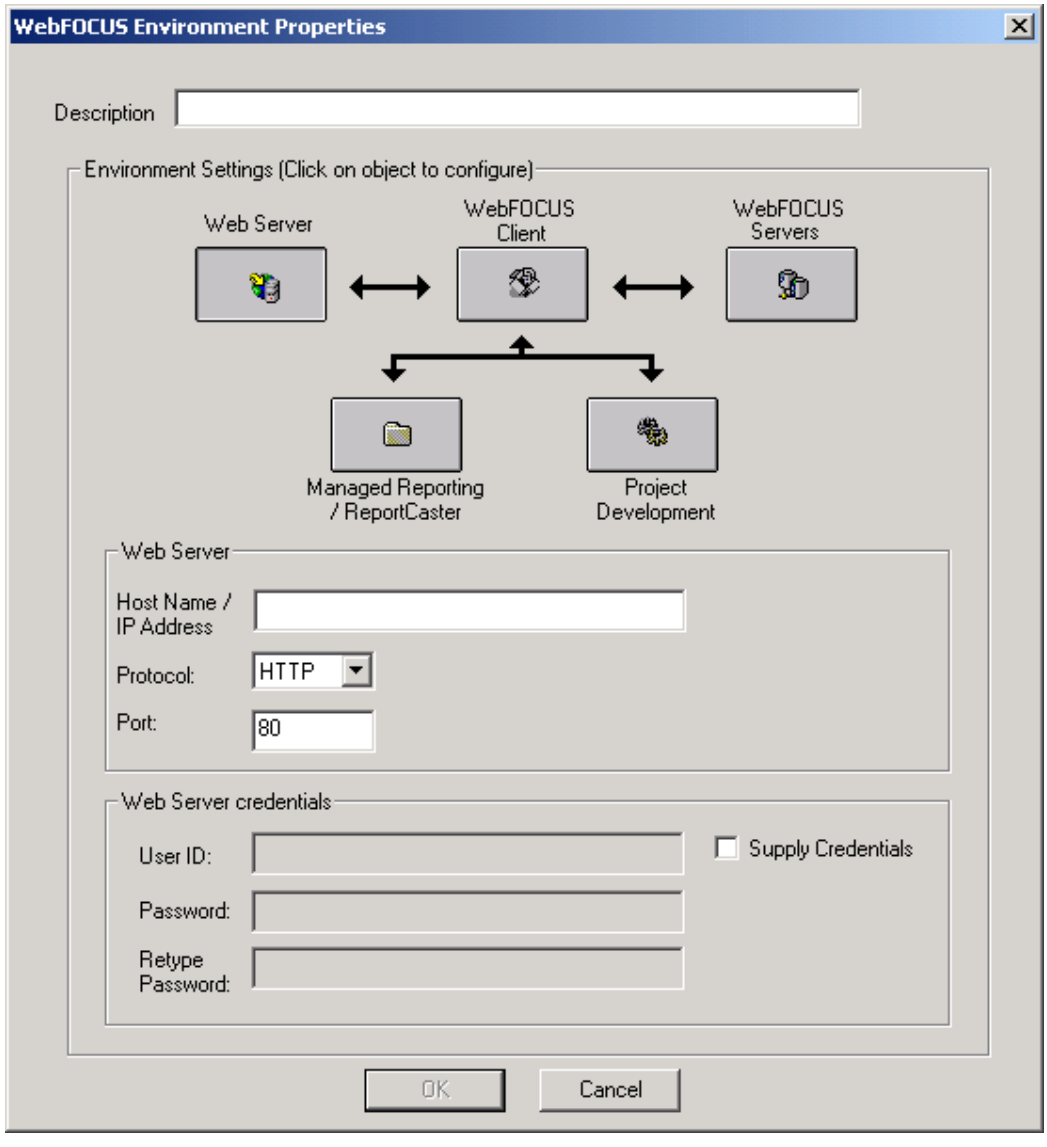
   The WebFOCUS Environment Properties dialog box opens. See *Working With WebFOCUS Environment Properties* on page 13-4 for details.
3. Edit the environment properties.
4. Click *OK*.

# Working With WebFOCUS Environment Properties

You define WebFOCUS environments and edit existing environments in Developer Studio from the WebFOCUS Environment Properties dialog box.

The WebFOCUS Environment Properties dialog box is used to set the properties required when adding an environment to the WebFOCUS Environments node of the Explorer tree. You can also use this dialog box to change the properties of an existing environment or configure the *localhost* environment, which is associated with the Project-based development features of Developer Studio. Generally, you will use this dialog box to configure access to remote WebFOCUS environments such as your test and production environments.

**Note:** Managed Reporting development is done exclusively with remote WebFOCUS environments that you add to your tree with information given to you by your WebFOCUS Administrator.




The WebFOCUS Environment dialog box contains two parts:

- The upper part contains a text box where you enter a description for the environment. This description displays on the Explorer tree under the WebFOCUS Environments node. The upper part also contains a series of buttons, arranged in a diagram, each of which depicts one component in the WebFOCUS environment. When the dialog opens, the first component button Web Server is pre-selected and the other buttons are disabled.

- The lower part of the dialog contains the configurable properties for the active component. Thus, Web server properties are displayed when the dialog first opens.

Because of the nature of a WebFOCUS environment, the properties that you configure must be supplied in a particular order. For example, if Web server security is enabled, you cannot set the WebFOCUS Client script name until you have provided valid Web server credentials. Similarly, you cannot retrieve a list of WebFOCUS Reporting Servers until you have provided a valid WebFOCUS Client script name. As you select a component button in the dialog box, Developer Studio ensures that it has the necessary information before it displays that component's properties in the lower part of the dialog. If the required information is not available you will not be able to continue to the next component.

Properties are stored in a persistent file on your computer so that you do not have to re-enter them each time you start Developer Studio. Logon IDs and passwords are encrypted to keep them confidential. This file is written to:

```
drive:\Documents and Settings\user_id\Application Data\Information
  Builders\wfscom.wfs
```

where:

*user_id*

Is your logged on Windows ID.

See:

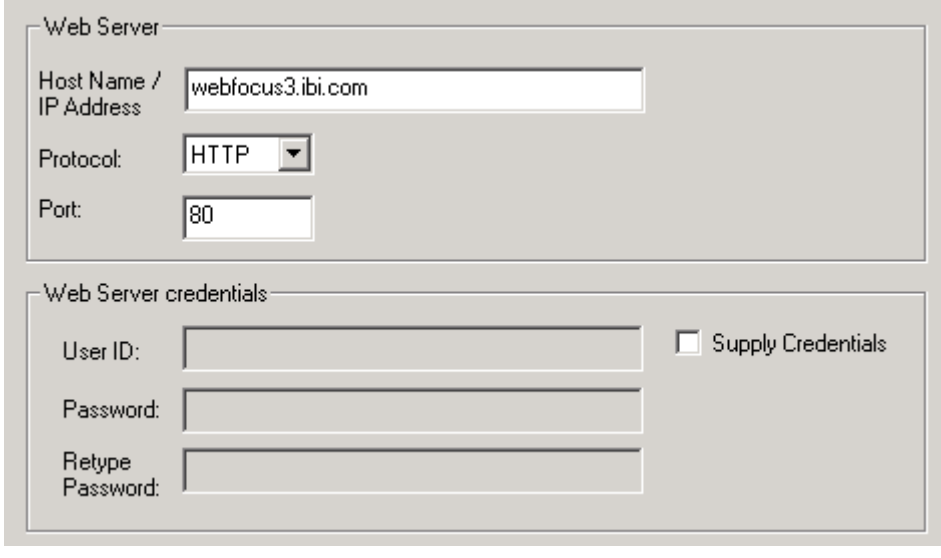
- *Web Server Properties* on page 13-7
- *WebFOCUS Client Properties* on page 13-8
- *WebFOCUS Server Properties* on page 13-10
- *Managed Reporting/ReportCaster Properties* on page 13-9
- *Project Development Properties* on page 13-11

for details on the purpose of each set of component properties.

## Web Server Properties

The following properties are available:

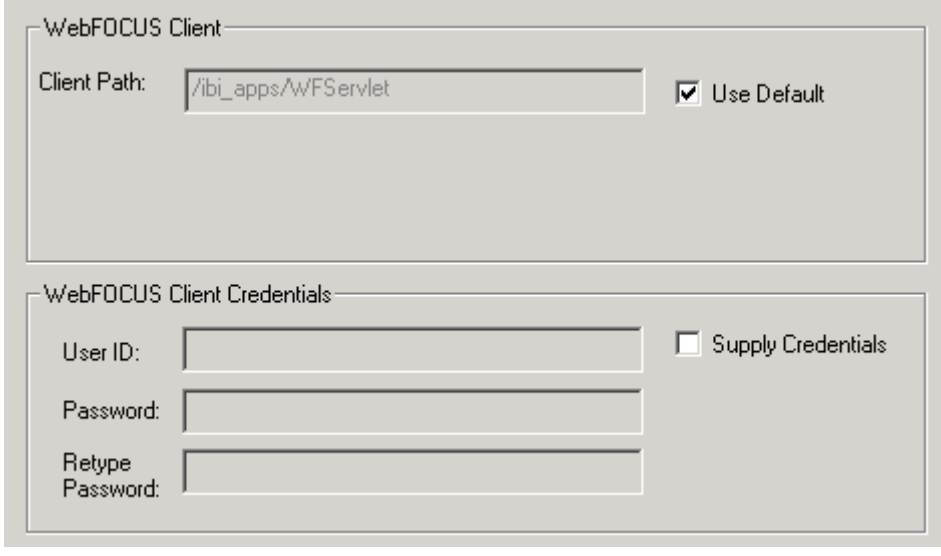
- **Host Name/IP Address.** The host name or TCP/IP address where your Web server is running. This field is required.
- **Protocol.** Currently only HTTP is supported. HTTPS will be enabled in a future release.
- **Port.** The TCP/IP port that your Web server is listening on. Port 80 is the default. This field is required
- **Supply Credentials.** If required, you can check this box and enter a Web server ID and password. If you leave this box unchecked, the Web server must allow anonymous access.




After you have entered your Web server information, the WebFOCUS Client and Managed Reporting/ReportCaster component buttons become enabled.

## WebFOCUS Client Properties

Before the WebFOCUS Client properties are displayed, the Web server properties are processed and Developer Studio makes a connection to your Web server to retrieve information about the WebFOCUS installation.




The following properties are available:

- **WebFOCUS Client Path.** This text box is initially disabled and displays the actual WebFOCUS Client path and script name configured on your Web server. If you are working with a remote WebFOCUS environment, the value you will probably see is:

  /ibi_apps/WFServlet

  but this could have been changed by your administrator. If you are working with the localhost environment, you will see /cgi-bin/ibi_cgi/webapi.dll.

  You can change this value by deselecting the Use Default check box. You should only do this if instructed to do so by your WebFOCUS Administrator. Typical values for the different implementations of the WebFOCUS Client are:

  - Servlet - /ibi_apps/WFServlet.
  - CGI - /cgi-bin/ibi_cgi/ibiweb.exe.
  - ISAPI - /cgi-bin/ibi_cgi/webapi.dll.

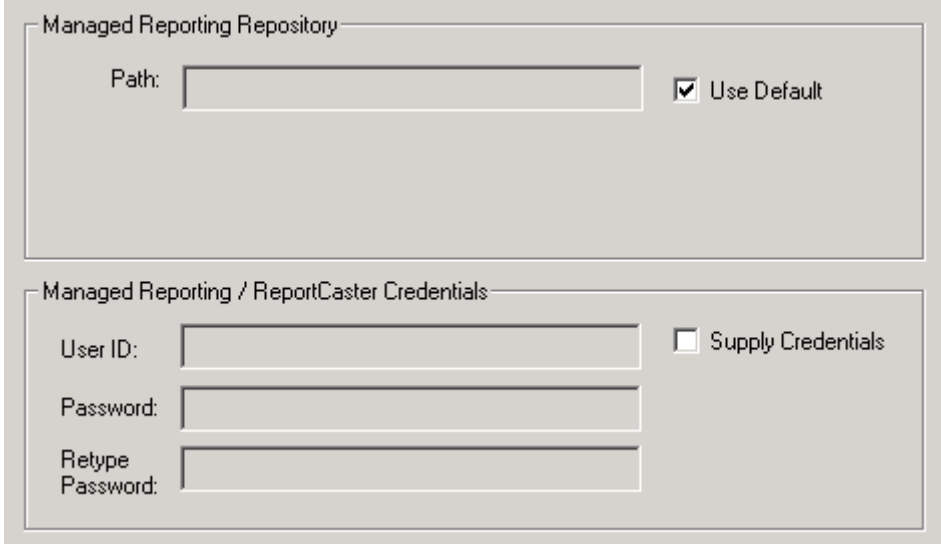
**Note:** If you do not see a value in this field and Use Default is selected, then there is a problem connecting to your Web server. Return to the Web Server component properties and review the information you supplied. Make sure that you have supplied Web server credentials if required. If the problem persists, the Web server may not be running or there may be a problem with the WebFOCUS software configuration. Contact your WebFOCUS Administrator.

- **Supply Credentials**. If required, you can check this box and enter your WebFOCUS Client ID and password. These credentials are only used for Project-based development and do not apply to Managed Reporting. They are used to check that the developer is authorized to perform certain activities such as deploying an application. By default, WebFOCUS is configured to allow all developers to perform these kinds of activities. Your WebFOCUS Administrator will let you know if you need to supply WebFOCUS Client credentials.

Once Developer Studio has detected your environment's default WebFOCUS Client Path, the remaining component buttons in the dialog box become enabled.

## Managed Reporting/ReportCaster Properties

If you plan to work with Managed Reporting (MR) and/or ReportCaster, you can optionally set credentials here. Your WebFOCUS Administrator may have integrated MR authentication with the Web server, in which case you would not supply MR credentials here because they will be picked up from the Web Server component properties.

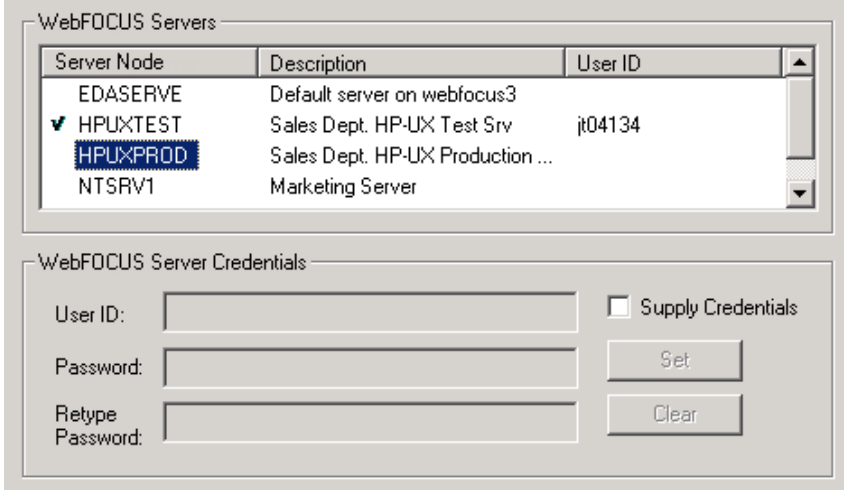


The MR Repository Default Path field is initially disabled and the Use Default check box is selected. You will not see a value in this field when Use Default is checked because it is not currently possible for WebFOCUS to detect the list of potential MR Repositories on the Web server. Your WebFOCUS Administrator may instruct you to deselect the Use Default check box and enter a path, for example:

- e:\user_test\basedir01 (for Windows NT/2000).
- /usr/user_test/basedir01(for UNIX).

## WebFOCUS Server Properties

Before the WebFOCUS Server properties are displayed, Developer Studio makes a connection to the WebFOCUS Client and retrieves a list of servers from its communication configuration file. The servers are displayed in a list that shows their Node Name and Description (optional). If credentials are stored for a particular server then that User ID is shown in the list also.




The following properties are available:

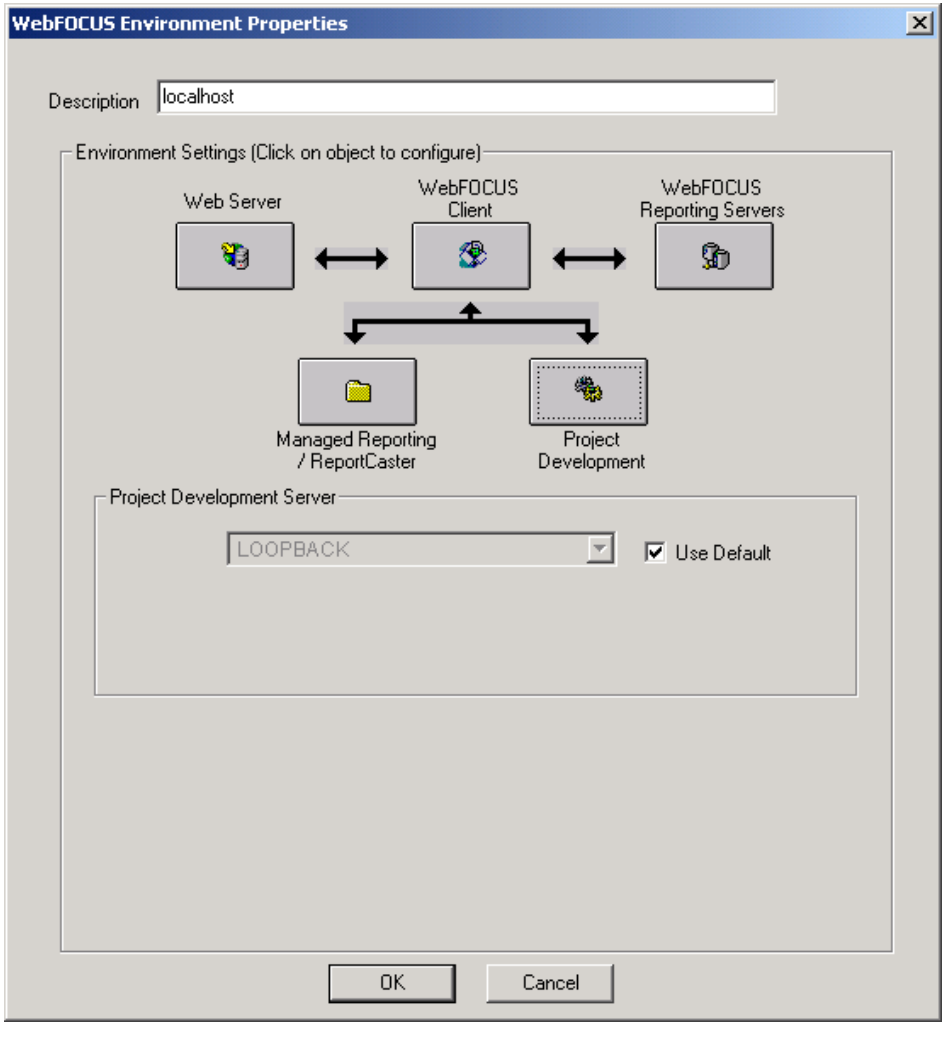
- **Supply Credentials.** If checked, you can enter a WebFOCUS Server ID and password for the server highlighted in the list. Clicking Set stores the credentials with the environment properties and the ID entered is shown next to the server in the list. The credentials are checked the next time you use a feature that goes to that server, not when you click the Set button. The Clear button removes stored credentials for the selected server.

**Note:** If you are working with a remote environment, the node EDASERVE may appear in the list. If you are working in the localhost environment, you will typically only see a node called LOOPBACK.

## Project Development Properties

Before the WebFOCUS Server properties are displayed, Developer Studio makes a connection to the WebFOCUS Client and retrieves a list of servers from its communication configuration file. This list is displayed in a drop-down list that can be selected. Initially the property Use Default is checked and the list is disabled.

Typically, you would only modify this property if you have changed the setting for your development environment (see *Setting the Development Environment* on page 13-12) from localhost to another environment. In this case, the WebFOCUS Server you select for Project development must be configured to point to the same application root directory (APPROOT) as that of the WebFOCUS Client.



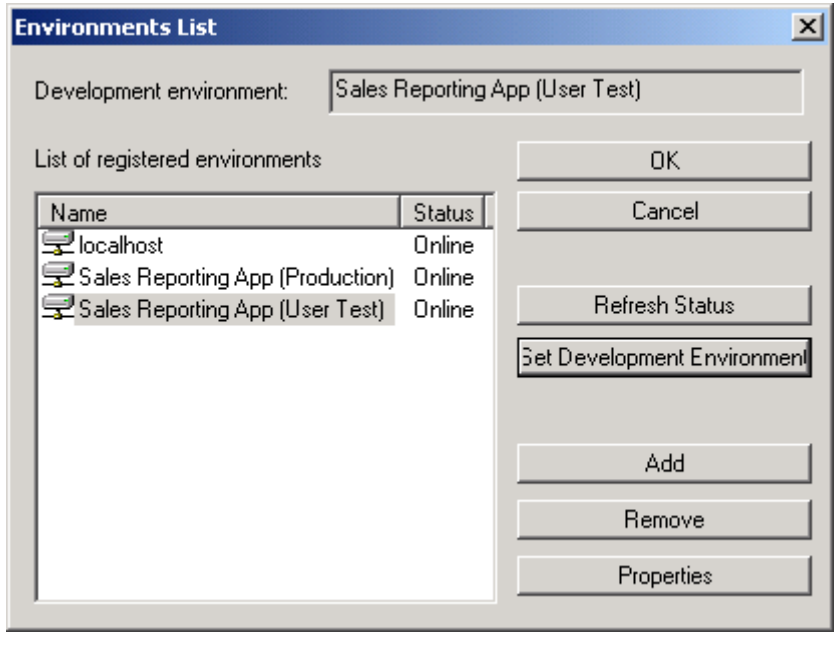
# Setting the Development Environment

If you have more than one available WebFOCUS environment, you can select which one to use for your Developer Studio projects.

## *Procedure* How to Set the Development Environment

1. Select *Show Environments* from the File menu. The Environments List dialog box opens.
2. Select the environment you want to set as the development environment, and click *Set Development Environment*.

   Your new development environment appears in the Development environment field.




3. Click *OK*.

## Refreshing or Removing an Environment

You can refresh the status of an environment or remove an environment from the Environments List dialog box.

### *Procedure* How to Remove an Environment

Right-click the environment in the Explorer window and select *Delete*.

OR

1. Select *Show Environments* from the File menu. The Environments List dialog box opens
2. Select the environment you want to delete, and click *Remove*.
3. Click *OK*.

### *Procedure* How to Refresh an Environment's Status

1. Select *Show Environments* from the File menu. The Environments List dialog box opens
2. Select the environment you want to refresh, and click *Refresh Status*.
3. Click *OK*.

CHAPTER 14

# Accessing a FOCUS Data Source

**Topics:**

- The USE Command
- The Developer Studio Use Tool
- Identifying a FOCUS Data Source
- Using Alternative File Specifications
- Identifying a New Data Source
- Protecting a Data Source
- Concatenating Data Sources
- Displaying the Current USE Options
- Clearing the USE Options

This topic describes how to access a FOCUS data source with the USE command. Some of the functionality of the USE command can be performed with the DATASET attribute in the Master File. For information on the DATASET attribute, see the *Describing Data With WebFOCUS Language* manual.

This topic contains information and examples for the following operating systems: Windows, OS/390, and UNIX. References to Windows include Windows 98, NT, and 2000.

You can also use Universal Concatenation to retrieve data from unlike data sources in a single request. See *Creating Reports With WebFOCUS Language* for more information.

# The USE Command

When you issue a command to access a FOCUS data source, such as TABLE FILE filename, WebFOCUS searches for a Master File with the specified file name, and then searches for a data source with the same name. The USE command specifies the name and location of a FOCUS data source. This is useful under the following conditions:

- The default naming convention is not used.
- The data source cannot be found on the standard search path.
- An explicit extra option is desired.

The USE command can be issued in a procedure with the Use tool in Developer's Studio, or with the USE command in Developer's Studio or WebFOCUS. When you identify a FOCUS data source with the USE command or tool, a USE directory is created, which is a list of data source definitions. When a USE directory is in effect, WebFOCUS locates a data source using the information in the directory instead of searching for it using the default name and search path. A USE directory enables you to access up to 255 data sources and applies only to FOCUS data sources. You can simplify the creation of a USE directory in the following ways:

- Include all USE specifications in a single file that you call with the -INCLUDE command at the beginning of each procedure that needs to access the USE directory. For details on the -INCLUDE command see *Calling Another Procedure With -INCLUDE* in Chapter 10, *Managing Flow of Control in an Application*.
- Include all USE specifications in a WebFOCUS Reporting Server profile. When the profile is invoked, the USE commands will be active.

The DATASET attribute in the Master File can perform some, but not all, of the same tasks as the USE command. For information on DATASET, see the *Describing Data With WebFOCUS Language* manual.

*Reference* **Additional File Specifications for FOCUS Data Sources**

On some operating systems, WebFOCUS also uses additional file specifications to identify a FOCUS data source:

- On **Windows** and **UNIX**, the FOCUS data source has a default file extension of .FOC. For example, the command TABLE FILE EMPLOYEE uses EMPLOYEE.FOC.
- On **OS/390**, additional specifications are not required. The command TABLE FILE EMPLOYEE uses the data source allocated to ddname EMPLOYEE.

# The Developer Studio Use Tool

In Developer Studio, you can identify a data source as well as perform the same functions as the USE command.

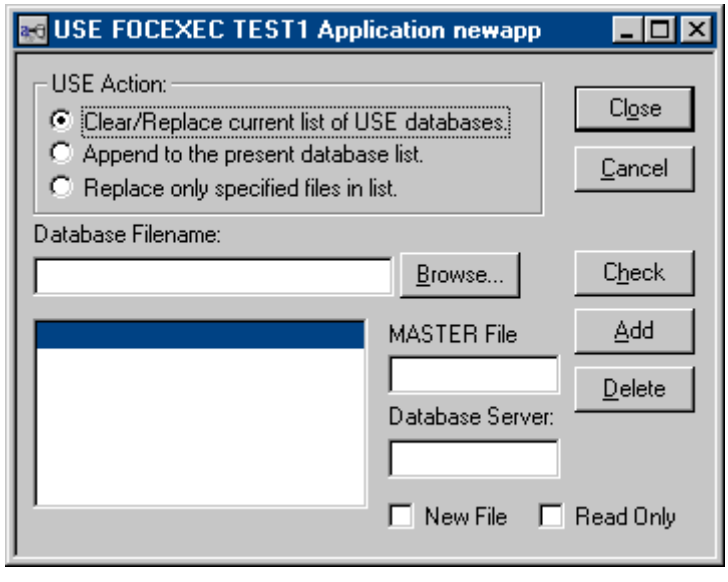
## *Procedure* How to Access the Use Tool in Developer Studio

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), then click the *Use* button on the Component Connector toolbox.

   The Use tool opens. For details on the Use dialog box, see *USE Dialog Box* on page 14-3.

## *Reference* USE Dialog Box



The USE dialog box has the following fields/options:

**USE Action**

**Clear/Replace current list of USE databases**

Clears all entries or replaces the current list of USE data sources with those in the list box.

**Append to the present database list**

Adds data sources to the current Use directory.

**Replace only specified files in list**

Changes one or more data source assignments in the current USE directory.

**Database Filename**

Displays the name of the data source, including the drive letter and complete path where applicable. All current file names appear in the box.

**MASTER File**

Displays the name of the Master File that is used to read the data source.

**Database Server**

Identifies the database server on which the data source resides.

**New File**

Creates a data source to the current USE directory.

**Read only**

Specifies read-only access to the selected data source.

**Close**

Closes the Use tool.

**Cancel**

Closes the USE dialog box without saving the USE settings.

**Check**

Displays the USE command in code, and specifies any errors.

**Add**

Adds a new data source to the current USE directory.

**Delete**

Removes the selected definition from the USE directory.

# Identifying a FOCUS Data Source

You can identify a FOCUS data source with the Use tool in Developer's Studio, or with the USE command in Developer Studio or WebFOCUS.

You can use a single USE command to specify one or several data sources. For examples, see *Identifying a Single Data Source* on page 14-7 and *Identifying Multiple Data Sources* on page 14-7.

***Procedure*** **How to Identify a Data Source With the Use Tool in Developer Studio**

1. In the Use tool, do one of the following:
   - Type the name of a FOCUS data source in the Database Filename field.
   - Click *Browse* to search the network drives and directories for the file you want to add.
   - Type the Universal Naming Convention (UNC) name of the file in the Database Filename field. For example:

     ```
     \\Server1\Disk1\employee.foc
     ```

     The Universal Naming Convention (UNC) allows you to enter the server name without entering the explicit path to the server. However, in order to take advantage of the UNC you must first attach to the server. For information on attaching to a server or mapping network drives, consult your Network Administrator.

     Enter the Master File name to be associated with the data source in the Master File field.
2. If the data source is not located locally, enter the name of the server on which the file resides in the Database Server field.
3. Click *Add*.
4. Repeat steps 3 through 6 for any additional data sources you want to identify.
5. Select a USE action:

   *Clear/Replace current list of USE databases* clears all entries or replaces the current list of USE data sources with those in the list box.

   *Append to the present database list* adds data sources to the current Use directory.

   *Replace only specified files in list* changes one or more data source assignments in the current USE directory.
6. Click *Close*. You are prompted to save your changes.

## *Syntax* How to Identify a Data Source With the USE Command

```
USE {ADD|CLEAR|REPLACE}
fileid [READ|NEW] [AS mastername] [ON resource]
[fileid [READ|NEW] [AS mastername] [ON resource] ]
.
.
.
END
```

where:

`ADD`

Appends one or more new file IDs to the present directory. If you issue the USE command without the ADD option, the list of data sources you specify replaces the existing USE directory.

`CLEAR`

Erases the USE directory. Any other specified options are ignored. The END command is not required with this option.

`REPLACE`

Replaces an existing file ID in the USE directory. This option enables you to change the specification for the file ID and the options following the ID.

*`fileid`*

Is any valid file name for the specific operating system:

| | |
|---|---|
| **Windows** | Any valid file name in *drive*:*path**filename.ext* format. The drive and path specifications are optional. |
| **OS/390** | A *ddname* allocated to the full OS/390 specification of the file. |
| **UNIX** | Any valid file name in */directory/filename.ext* format. The directory specification is optional. |

`READ`

Restricts the data source to read-only access.

`NEW`

Indicates that the data source is new.

`AS` *`mastername`*

Specifies the name of the Master File associated with the file ID.

Both the READ and NEW option can be used with the AS option. Any other combination of options after the file ID is *not* valid.

`ON resource`

Specifies a specially configured WebFOCUS Reporting Server that synchronizes FOCUS data sources for use by multiple end users.

## *Example* Identifying a Single Data Source

You can specify a data source with the USE command. For example,

**Windows:**

```
USE
EMP026.FOC AS PRODUCTS
END
```

**OS/390:**

```
USE
EMP026 AS PRODUCTS
END
```

**UNIX:**

```
USE
emp026.foc AS PRODUCTS
END
```

## *Example* Identifying Multiple Data Sources

You can specify several data sources in one USE command, each with different parameters. For example,

**Windows:**

```
USE
EMP024.FOC ON MULTID
EMP025.FOC AS PRODUCTS
EMP026.FOC READ AS ACCOUNTS
END
```

**OS/390:**

```
USE
EMP024 ON MULTID
EMP025 AS PRODUCTS
EMP026 READ AS ACCOUNTS
END
```

**UNIX:**

```
USE
emp024.foc ON MULTID
emp025.foc AS PRODUCTS
emp026.foc READ AS ACCOUNTS
END
```

# Using Alternative File Specifications

Issue the USE command to specify an alternate file specification for a FOCUS data source if:

- The data source has a name different from the associated Master File.
- The data source has a type or extension different from the default.
- The data source is located in a directory outside the standard search path.

**Note:** A data source named by the LOCATION attribute in a Master File assumes default values unless you issue a USE command. If the physical file has a specification or location other than the default, it must be in the USE directory.

*Procedure* **How to Use an Alternative File Specification With the Use Tool in Developer Studio**

1. In the User tool, do one of the following:
   - Type the name of a FOCUS data source in the Database Filename field.
   - Click *Browse* to search the network drives and directories for the file you want to add.
   - Type the Universal Naming Convention (UNC) name of the file in the Database Filename field. For example:

     ```
     \\Server1\Disk1\employee.foc
     ```

     The Universal Naming Convention (UNC) allows you to enter the server name without entering the explicit path to the server. However, in order to take advantage of the UNC you must first attach to the server. For information on attaching to a server or mapping network drives, consult your Network Administrator.

     Enter the Master File name to be associated with the data source in the Master File field.
2. If the data source is not located locally, enter the name of the server on which the file resides in the Database Server field.
3. Click *Add*.
4. Repeat steps 3 through 6 for any additional data sources you want to identify.

5. Select a USE action:

   *Clear/Replace current list of USE databases* clears all entries or replaces the current list of USE data sources with those in the list box.

   *Append to the present database list* adds data sources to the current Use directory.

   *Replace only specified files in list* changes one or more data source assignments in the current USE directory.

6. Click *Run*.
7. Click *Close*. You are prompted to save your changes.

## *Syntax* How to Use an Alternative File Specification

```
USE
fileid AS mastername
END
```

where:

*fileid*

Is any valid file name for the specific operating system.

*mastername*

Is the name of the Master File associated with the file ID.

## *Example* Specifying a Different File Name

To read the EMP026 data source described by the Master File EMPLOYEE, issue the USE command. After issuing the USE command, you can read EMP026 with the command TABLE FILE EMPLOYEE.

**Windows:**

```
USE
EMP026.FOC AS EMPLOYEE
END
```

**OS/390:**

```
USE
EMP026 AS EMPLOYEE
END
```

**UNIX:**

```
USE
emp026.foc AS EMPLOYEE
END
```

## *Example* Specifying a Different File Type or Extension

**Windows:**

To read the EMP026 data source with an extension of .DAT, described by the Master File EMPLOYEE, issue the command:

```
USE
EMP026.DAT AS EMPLOYEE
END
```

**OS/390:**

To read the emp026 data source with an extension of .dat, described by the Master File EMPLOYEE, issue the command:

```
USE
EMP026 AS EMPLOYEE
END
```

**UNIX:**

To read the emp026 data source with an extension of .dat, described by the Master File EMPLOYEE, issue the command:

```
USE
emp026.dat AS EMPLOYEE
END
```

After issuing the USE command, you can read EMP026 with the command TABLE FILE EMPLOYEE.

## *Example* Specifying a Different File Location

**Windows:**

To read the EMP026 data source located in the C:\ACCOUNTING\ directory, described by the Master File EMPLOYEE, issue the command:

```
USE
C:\ACCOUNTING\EMP026.FOC AS EMPLOYEE
END
```

**UNIX:**

To read the emp026 data source located in the /accounting/ directory, described by the Master File EMPLOYEE, issue the command:

```
USE
/accounting/emp026.foc AS EMPLOYEE
END
```

# Identifying a New Data Source

You can identify a data source that does not exist yet. When you identify a new data source, you can accept the default file specification conventions or specify different ones.

**Note:** For OS/390, you must allocate the file before issuing the USE command. You can do this using the OS/390 command ALLOCATE or the WebFOCUS command DYNAM. You must also issue the OS/390 command CREATE to physically create the file. This is not a requirement for platforms that dynamically allocate files.

*Procedure* **How to Identify a New Data Source With the Use Tool in Developer Studio**

1. In the User tool, do one of the following:
   - Type the name of a FOCUS data source in the Database Filename field.
   - Click *Browse* to search the network drives and directories for the file you want to add.
   - Type the Universal Naming Convention (UNC) name of the file in the Database Filename field. For example:

     ```
     \\Server1\Disk1\employee.foc
     ```

     The Universal Naming Convention (UNC) allows you to enter the server name without entering the explicit path to the server. However, in order to take advantage of the UNC you must first attach to the server. For information on attaching to a server or mapping network drives, consult your Network Administrator.

     Enter the Master File name to be associated with the data source in the Master File field.
2. If the data source is not located locally, enter the name of the server on which the file resides in the Database Server field.
3. Click *New File*.
4. Click *Add*.
5. Repeat steps 3 through 6 for any additional data sources you want to identify.
6. Select a USE action:

   *Clear/Replace current list of USE databases* clears all entries or replaces the current list of USE data sources with those in the list box.

   *Append to the present database list* adds data sources to the current Use directory.

   *Replace only specified files in list* changes one or more data source assignments in the current USE directory.

7. Click *Run*.
8. Click *Close*. You are prompted to save your changes.

## *Syntax* How to Identify a New Data Source

```
USE
fileid NEW
END
```

where:

*fileid*

Is any valid file specification for the operating system. The file ID will be assigned to the data source when it is created.

NEW

Specifies that the data source does not yet exist. If you omit the NEW parameter, a message is returned stating that the data source cannot be found. The USE command is not executed.

## *Example* Identifying a New Data Source

The following command creates the data source WAGES using the WAGES Master File.

**Windows:**

```
USE
C:\DATA\WAGES.FOC NEW
END
CREATE FILE WAGES
```

**OS/390:**

```
USE
WAGES NEW
END
CREATE FILE WAGES
```

**UNIX:**

```
USE
wages.foc NEW
END
CREATE FILE WAGES
```

## Protecting a Data Source

You can protect a data source from changes by including the READ parameter when issuing a USE command. A protected data source can be read by various tools and commands, but cannot be changed.

*Procedure* **How to Protect a Data Source With the Use Tool in Developer Studio**

1. In the Use tool, do one of the following:
   - Type the name of a FOCUS data source in the Database Filename field.
   - Click *Browse* to search the network drives and directories for the file you want to add.
   - Type the UNC name of the file in the Database Filename field.
   - Enter the Master File name to be associated with the data source in the Master File field.
2. If the data source is not located locally, enter the name of the server on which the file resides in the Database Server field.
3. Click *Add*.
4. Repeat steps 3 through 6 for any additional data sources you want to identify.
5. Select a USE action:

   *Clear/Replace current list of USE databases* clears all entries or replaces the current list of USE data sources with those in the list box.

   *Append to the present database list* adds data sources to the current Use directory.

   *Replace only specified files in list* changes one or more data source assignments in the current USE directory.
6. Select *Read Only*.
7. Click *Run*.
8. Click *Close*. You are prompted to save your changes.

### *Syntax* How to Protect a Data Source

```
USE
fileid READ
END
```

where:

*fileid*

Is any valid file specification for the operating system.

READ

Restricts the data source to read-only access.

### *Example* Protecting a Data Source

The following command protects the EMPLOYEE data source from modification.

**Windows:**

```
USE
EMPLOYEE.FOC READ
END
```

**OS/390:**

```
USE
EMPLOYEE READ
END
```

**UNIX:**

```
USE
/usr/mydata/employee.foc READ
END
```

## Concatenating Data Sources

If several FOCUS data sources are described by the same Master File, you can read all of them in one request by issuing a USE command that concatenates them.

You can also concatenate multiple files that cross-reference a common file. This is done by specifying the host files in the USE command, then specifying the cross-reference file.

**Note:** You cannot specify a concatenated file as a cross-reference file in a JOIN command.

## *Procedure* How to Concatenate Data Sources With the Use Tool in Developer Studio

1. In the User tool, do one of the following:
   - Type the name of a FOCUS data source in the Database Filename box.
   - Click *Browse* to search the network drives and directories for the file you want to add.
   - Type the Universal Naming Convention (UNC) name of the file in the Database Filename box. For example:

     ```
     \\Server1\Disk1\employee.foc
     ```

     The Universal Naming Convention (UNC) allows you to enter the server name without entering the explicit path to the server. However, in order to take advantage of the UNC you must first attach to the server. For information on attaching to a server or mapping network drives, consult your Network Administrator.
2. Enter the Master File name to be associated with the data source in the Master File field. When concatenating data sources, use the same Master File name for each data source.
3. If the data source is not located locally, enter the name of the server on which the file resides in the Database Server box.
4. Click *Add*.
5. Repeat steps 3 through 6 for any additional data sources you want to identify.
6. Select a USE action:

   *Clear/Replace current list of USE databases* clears all entries or replaces the current list of USE data sources with those in the list box.

   *Append to the present database list* adds data sources to the current Use directory.

   *Replace only specified files in list* changes one or more data source assignments in the current USE directory.
7. Click *Run*.
8. Click *Close*. You are prompted to save your changes.

## *Syntax* How to Concatenate Data Sources

```
USE
fileid-1 AS masternamе
fileid-2 AS masternamе
.
.
fileid-n AS masternamе
END
```

where:

*fileid-1...fileid-n*

Are any valid file specifications for the files being concatenated.

*mastername*

Is the name of the Master File that describes the data sources.

## *Example* Concatenating Data Sources

The following command concatenates the FOCUS data sources EMP024, EMP025, and EMP026, all described by the Master File EMPLOYEE. You can then read all three data sources with a single TABLE FILE EMPLOYEE command.

**Windows:**

```
USE
C:\DATA\EMP024.FOC AS EMPLOYEE
C:\DATA\EMP025.FOC AS EMPLOYEE
C:\DATA\EMP026.FOC AS EMPLOYEE
END
```

**OS/390:**

```
USE
EMP024 AS EMPLOYEE
EMP025 AS EMPLOYEE
EMP026 AS EMPLOYEE
END
```

**UNIX:**

```
USE
/usr/mydata/emp024.foc AS EMPLOYEE
/usr/mydata/emp025.foc AS EMPLOYEE
/usr/mydata/emp026.foc AS EMPLOYEE
END
```

### *Example* Specifying Multiple Concatenations

The following command concatenates the EMP01 and EMP02 data sources described by the Master File EMPLOYEE, and concatenates the SALES01 and SALES02 data sources, described by the Master File SALES.

**Windows:**

```
USE
EMP01.FOC AS EMPLOYEE
EMP02.FOC AS EMPLOYEE
SALES01.FOC AS SALES
SALES02.FOC AS SALES
END
```

**OS/390:**

```
USE
EMP01 AS EMPLOYEE
EMP02 AS EMPLOYEE
SALES01 AS SALES
SALES02 AS SALES
END
```

**UNIX:**

```
USE
emp01.foc AS EMPLOYEE
emp02.foc AS EMPLOYEE
sales01.foc AS SALES
sales02.foc AS SALES
END
```

You can read the EMP01 and EMP02 data sources with the TABLE FILE EMPLOYEE command. You can read the SALES01 and SALES02 data sources with the TABLE FILE SALES command.

## *Syntax* How to Concatenate Data Sources and a Single Cross-reference File

```
USE
fileid-1 AS mastername
fileid-2 AS mastername
.
.
fileid-n AS mastername
reffile

END
```

where:

*fileid-1...fileid-n*

Are any valid file specifications for the files being concatenated.

*mastername*

Is the name of the Master File that describes the data sources.

*reffile*

Is the cross-reference file for the files being concatenated.

### *Example* Concatenating Data Sources and a Single Cross-reference File

EMPLOYEE is made up of the two files EMP01 and EMP02, both of which cross-reference the common file EDUCFILE. To read the files together, issue the USE command.

**Windows:**

```
USE
EMP01.FOC AS EMPLOYEE
EMP02.FOC AS EMPLOYEE
EDUCFILE.FOC
END
```

**OS/390:**

```
USE
EMP01 AS EMPLOYEE
EMP02 AS EMPLOYEE
EDUCFILE
END
```

**UNIX:**

```
USE
/usr/mydata/emp01.foc AS EMPLOYEE
/usr/mydata/emp02.foc AS EMPLOYEE
/usr/mydata/educfile.foc
END
```

### *Example* Concatenating Data Sources and Multiple Cross-reference Files

If EMPLOYEE consists of two files, EMP01 and EMP02, and each has its own cross-reference file, ED01 and ED02, you can read all four files in one command. Issue the USE command in which each host file is followed by its cross-reference file.

**Windows:**

```
USE
EMP01.FOC AS EMPLOYEE
ED01.FOC AS EDUCFILE
EMP02.FOC AS EMPLOYEE
ED02.FOC AS EDUCFILE
END
```

**OS/390:**

```
USE
EMP01 AS EMPLOYEE
ED01  AS EDUCFILE
EMP02 AS EMPLOYEE
ED02  AS EDUCFILE
END
```

**UNIX:**

```
USE
/usr/mydata/emp01.foc AS EMPLOYEE
/usr/mydata/ed01.foc AS EDUCFILE
/usr/mydata/emp02.foc AS EMPLOYEE
/usr/mydata/ed02.foc AS EDUCFILE
END
```

## Displaying the Current USE Options

To display the USE options in effect, issue the ? USE query. This query displays a list of data sources specified with the USE command and the options currently in effect.

*Syntax* **How to Display the Current USE Options**

```
? USE
```

*Example* **Displaying Current USE Options**

A sample response to the ? USE command is:

```
 DIRECTORIES IN USE ARE:
C:\IBI\apps\ibisamp\CENTINV.FOC AS CENTINV
C:\IBI\apps\ibisamp\CENTHR.FOC AS CENTHR
```

## Clearing the USE Options

You can clear the USE directory with the USE command.

*Syntax* **How to Erase the USE Directory**

```
USE CLEAR
```

CHAPTER 15

# Customizing Your Environment

**Topics:**

- When Do You Use the SET Command?
- Ways to Issue a SET Command
- Issuing a SET Command With the Set Tool
- Coding a SET Command
- Types of SET Parameters
- SET Parameter Syntax

The SET command enables you to customize both the Developer Studio and WebFOCUS environments. It controls the way reports and graphs disPublish Utilityplay on the screen or printer; the content of reports and graphs; data retrieval characteristics that affect performance; and system responses to end user requests. It also helps you set up your metadata and manipulate information such as dates.

The SET command governs many important aspects of your work.

## When Do You Use the SET Command?

If you are an application developer, use the SET command to:

- Help you work efficiently and meet your testing and debugging needs.
- Provide a uniform and appropriate run-time environment for your end users with desirable defaults that do not need customization.

In Developer Studio, Managed Reporting developers and administrators can also use SET commands to:

- Customize Developer Studio to standardize development and debug procedures.
- Customize the administrator environment as they develop domains and reports, and test them before distribution to their end users.
- Create a secure, precisely defined environment for their end users.

If you are creating your own ad hoc reports, use SET commands when you need to tailor the report presentation or content to meet your individual needs.

## Ways to Issue a SET Command

You can issue a SET command by using the SET tool to generate the command in the stored procedure, or by coding it in a stored procedure. If you use the SET tool, a SET command will be generated for the report you are developing. If you code the SET command, you have the following options for its placement:

- In a supported profile on a WebFOCUS Reporting Server. When issued in a profile, the setting applies to the server environment during the connection for executing the current procedure.
- In a stored procedure:
  - Before the report request.
  - With ON TABLE SET or ON TABLE GRAPH in a report or graph request. SET used in this manner applies only to the report or graph request in which it is issued, even though there may be multiple report or graph requests in the stored procedure.
- In a type Other reporting object, in the Managed Reporting administrator's environment. A setting issued in a reporting object (type Other) applies to a single report template.
- In the WebFOCUS script file site.wfs. When issued in site.wfs, the setting applies to all connections to the WebFOCUS Reporting Server from the WebFOCUS Client.

### *Reference* Precedence of the SET Command

If you issue more than one SET command for the same parameter, the precedence is as follows:

**Developer Studio**

1. ON TABLE SET or ON GRAPH SET in a stored procedure.
2. SET in a stored procedure.

**WebFOCUS**

1. SET in a stored procedure.
2. SET in the site.wfs or ibidir.wfs file.
3. SET in a profile on the WebFOCUS Reporting Server.
   a. In a global profile.
   b. In a group profile.
   c. In a user profile.

## Issuing a SET Command With the Set Tool

In Developer's Studio, the Set tool provides a graphical option for issuing the SET command. The setting lasts for a single procedure.

Not all parameters can be set using the Set tool. For a complete list of parameters, see *SET Parameter Syntax* on page 15-20.

### *Procedure* How to Set Environment Parameters With the Set Tool

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), then click the *Set*

button on the Component Connector toolbox.

   The Set tool opens. See *Set Dialog Box* on page 15-4.
3. In the Available Settings list in the Set window:
   - Select the parameter to change, then click *Add*.

     or
   - Double-click the parameter.

The new parameter is added to the Used Settings list.

4. Highlight the parameter by single-clicking it.

   Possible values appear in the Current Value list, and the current value is highlighted.

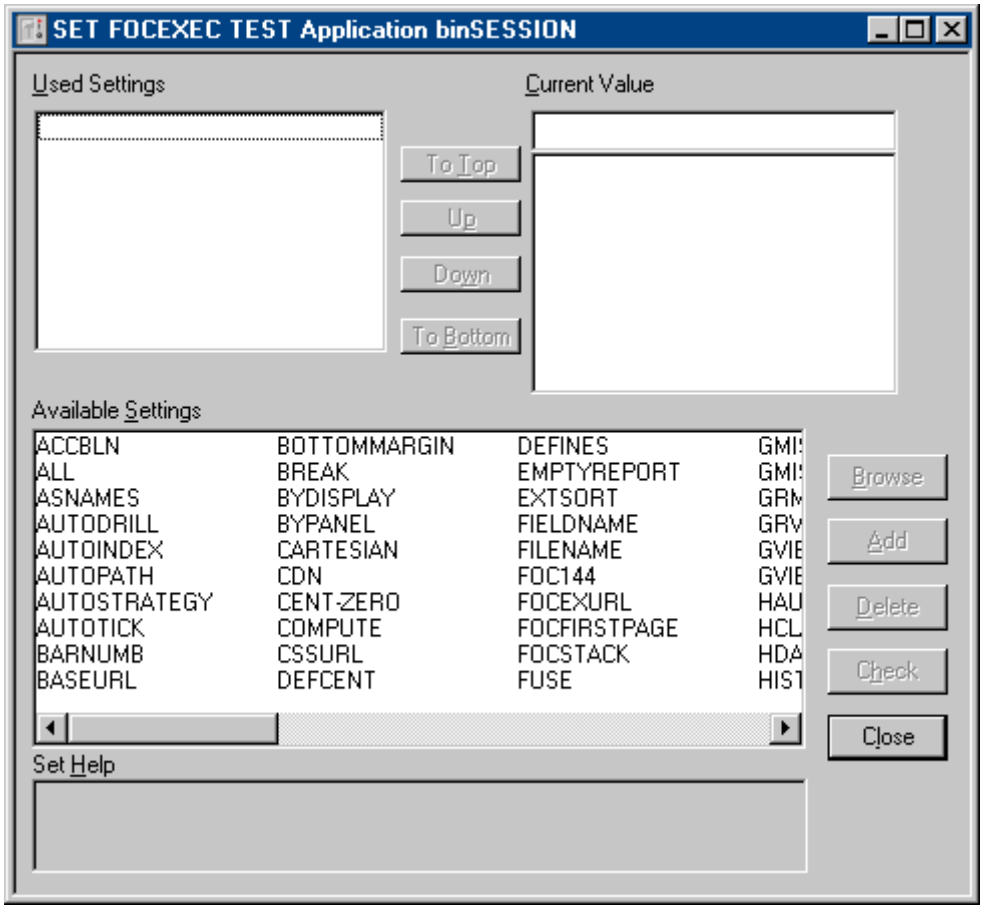
5. Specify the value:

   - Choose a value from the Current Value list.

     or

   - Enter a valid value in the box at the top.

## *Reference* Set Dialog Box



The Set window has the following entry fields and buttons:

**Available Settings**

Lists available parameters. Use the left and right arrow buttons to scroll through the list of parameters. For a full list see *SET Parameter Syntax* on page 15-20.

**Used Settings**

Lists parameters that have values assigned to them for this application.

To move parameters within the list, select a parameter and use the *To Top*, *Up*, *Down*, and *To Bottom* buttons.

**Current Value**

Lists the possible values for the selected parameter in the Used Settings list; the current value is highlighted.

**Set Help**

Displays a description of the currently highlighted parameter.

**Add**

Adds a parameter from the Available Settings list to the Used Settings list.

**Check**

Displays the SET command in code, and specifies any errors.

**Delete**

Removes a parameter from the Used setting list.

**Close**

Closes the Set tool.

## Coding a SET Command

The following guidelines apply to SET command syntax:

- You can set several parameters in one command by separating each with a comma.
- You can include as many parameters as you can fit on one line. If you exceed one line, repeat the SET command for each new line.
- You can set many, but not all, parameters using ON TABLE SET or ON GRAPH SET within a request. Parameters that cannot be set in this way are specified in the detailed description.

For the specific syntax of a parameter with its valid values, see *SET Parameter Syntax* on page 15-20.

## Syntax How to Set Parameters

```
SET parameter = option[, parameter = option,...]
```

where:

*parameter*

Is the setting you wish to change.

*option*

Is a valid value for the parameter.

## Example Setting a Single Parameter

In the following example, the PAGE-NUM parameter suppresses default page numbering.

```
SET PAGE-NUM = OFF

TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
END
```

The output is:

| LAST_NAME | FIRST_NAME |
|---|---|
| STEVENS | ALFRED |
| SMITH | MARY |
| JONES | DIANE |
| SMITH | RICHARD |
| BANNING | JOHN |
| IRVING | JOAN |
| ROMANS | ANTHONY |
| MCCOY | JOHN |
| BLACKWOOD | ROSEMARIE |
| MCKNIGHT | ROGER |
| GREENSPAN | MARY |
| CROSS | BARBARA |

**Note:** In this example, the output was formatted to eliminate grids. For information on formatting your output, see the *Creating Reports With WebFOCUS Language* manual.

## *Example* Setting Multiple Parameters

The following example sets two parameters in one command in a stored procedure. The first parameter, NODATA, changes the default character for missing data from a period to the word NONE; the second parameter, PAGE-NUM, suppresses default page numbering.

```
SET NODATA = NONE, PAGE-NUM = OFF
TABLE FILE EMPLOYEE
PRINT CURR_SAL BY EMP_ID
ACROSS DEPARTMENT
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
END
```

In the output, NONE appears when there is no salary information for a specific employee because that employee does not work in the department that is referenced. There is no page number at the top of the output.

The output is:

```
                DEPARTMENT
                       MIS     PRODUCTION
EMP_ID
-----------------------------------------------------
071382660             NONE       $11,000.00
112847612        $13,200.00             NONE
117593129        $18,480.00             NONE
119265415             NONE        $9,500.00
119329144             NONE       $29,700.00
123764317             NONE       $26,862.00
126724188             NONE       $21,120.00
219984371        $18,480.00             NONE
326179357        $21,780.00             NONE
451123478             NONE       $16,100.00
543729165         $9,000.00             NONE
818692173        $27,062.00             NONE
```

**Note:** In this example, the output was formatted to eliminate grids. For information on formatting your output, see the *Creating Reports With WebFOCUS Language* manual.

## *Syntax* How to Set Parameters in a Report Request

```
ON TABLE SET parameter value [AND parameter value ...]
```

where:

*parameter*

Is the setting you wish to change.

*value*

Is a valid value for the parameter.

## *Example* Setting Parameters in a Report Request

In the following example, the command ON TABLE SET changes the default character for missing data from a period to the word NONE and suppresses default page numbering.

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL BY EMP_ID
ACROSS DEPARTMENT
ON TABLE SET NODATA NONE AND PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
END
```

In the output, NONE appears when there is no salary information for a specific employee. There is no page number at the top of the output.

The output is:

| | DEPARTMENT | |
|---|---|---|
| | MIS | PRODUCTION |
| EMP ID | | |
| 071382660 | NONE | $11,000.00 |
| 112847612 | $13,200.00 | NONE |
| 117593129 | $18,480.00 | NONE |
| 119265415 | NONE | $9,500.00 |
| 119329144 | NONE | $29,700.00 |
| 123764317 | NONE | $26,862.00 |
| 126724188 | NONE | $21,120.00 |
| 219984371 | $18,480.00 | NONE |
| 326179357 | $21,780.00 | NONE |
| 451123478 | NONE | $16,100.00 |
| 543729165 | $9,000.00 | NONE |
| 818692173 | $27,062.00 | NONE |

**Note:** In this example, the output was formatted to eliminate grids. For information on formatting your output, see the *Creating Reports With WebFOCUS Language* manual.

## *Syntax* How to Set Parameters in a Graph Request

```
ON GRAPH SET parameter value [AND parameter value ...]
```

where:

*parameter*

Is the setting you wish to change.

*value*

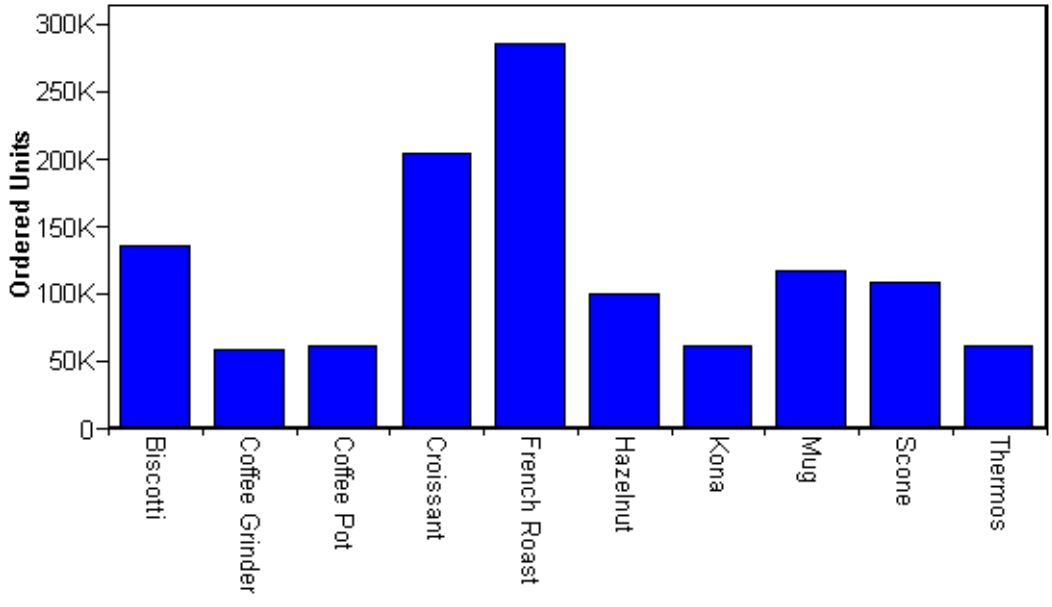
Is a valid value for the parameter.

## *Example* Setting Parameters in a Graph Request

In the following example, the command ON GRAPH SET changes the default setting for the 3D parameter to OFF.

```
GRAPH FILE GGORDER
SUM QUANTITY
ACROSS PRODUCT_DESC
ON GRAPH SET 3D OFF
END
```

The output is:



# Types of SET Parameters

This topic lists the types of tasks that can be accomplished, and the SET parameters that allow you to perform these tasks. If a single parameter applies to more than one activity, it appears in more than one category. For more detailed descriptions, as well as the syntax for each parameter, see *SET Parameter Syntax* on page 15-20.

The following are the types of tasks performed with SET parameters:

**Calculations**

Affects the way calculations are performed in WebFOCUS.

**Data and Metadata**

Determines the way data is stored and processed.

**Date Manipulation Tasks**

Controls the way dates are processed and displayed in reports.

**WebFOCUS Specific Tasks**

Affects the behavior of Web-specific tasks in your application.

**Developer Studio Specific Tasks**

Affects the environment in projects being developed in Developers Studio.

**Graph Tasks**

Controls the processing and display of graphs.

**Memory Setup and Optimization Tasks**

Affects the memory and optimization of your application.

**Report Code and Content Tasks**

Determines the content and processing of a request.

**Report Layout and Display Tasks**

Affects the display of a report.

**Security Tasks**

Controls users' access to data sources and procedures.

## Calculations

The following parameters control the behavior of calculations in WebFOCUS.

| | |
|---|---|
| CDN | Specifies the punctuation used in numerical notation. |
| COMPUTE | Controls the compile of computes. |
| PACKPOS | In WebFOCUS, defines the positive character value for packed fields. |
| %STRICTMATH | In WebFOCUS, determines the behavior of a calculation when dividing by zero. |

## Data and Metadata

The following parameters determine the way data is stored and processed.

| | |
|---|---|
| ASNAMES | Controls the FIELDNAME attribute in a HOLD Master File. |
| DATEFORMAT | Specifies the order of the date components (month/day/year) when date-time values are entered in a formatted-string or translated-string format. |
| DEFINES | Compiles virtual fields into machine code to improve performance. |
| DTSTRICT | Controls the use of strict processing. |
| EUROFILE | Activates the data source that contains information for the currency you want to convert. |
| FIELDNAME | Controls the use of long field names (66 characters). |
| HOLDFORMAT | Determines the default format for HOLD files. |
| HOLDLIST | Determines what fields in a report request are included in the HOLD file. |
| HOLDMISS | Distinguishes between missing data and default data in a HOLD file. |
| HOLDSTAT | Determines if comments and DBA information are included in HOLD Master Files. |
| KEEPDEFINES | Controls whether a virtual field created for a host or joined structure is retained after a JOIN command is run. |
| MAXLRECL | Specifies the maximum length of a record described with the Master File OCCURS attribute. |

| | |
|---|---|
| PCOMMA | Enables FOCUS to read PC-type comma delimited files with double quotation marks around alpha data and a carriage return line at the end of each record. |
| QUALCHAR | Specifies the qualifying character to be used in qualified field names. |
| SAVEDMASTERS | Saves a Master File to memory after it has been used in a request. |
| TEMPERASE | Determines if the temporary files created during a WebFOCUS connection are retained after the connection is closed. |
| WEEKFIRST | Specifies what day of the week is the start of the week. |
| XRETRIEVAL | Controls the retrieval of data when previewing a report. |

## Date Manipulation Tasks

The following parameters control the way dates are processed and displayed in reports.

| | |
|---|---|
| BUSDAYS | Specifies which days are considered business days and which days are not. |
| DATEDISPLAY | Controls the display of date format fields that contain the value zero. |
| DATEFNS | Activates year 2000-compliant versions of date subroutines. |
| DATETIME | In iWay, controls the format in which CREATE SYNONYM creates date-time columns in a Master File. |
| DEFCENT | Defines a default century for your application. |
| HDAY | Specifies the holiday file from which to retrieve dates that are considered holidays. |
| LEADZERO | Avoids the truncation of leading zeros. |
| TESTDATE | Temporarily alters the system date in order to test a dynamic window. |
| YRTHRESH | Defines the start of a 100-year window. |

## WebFOCUS Specific Tasks

The following parameters affect the behavior of Web-specific tasks in WebFOCUS.

| | |
|---|---|
| `BASEURL` | Specifies a default location where your browser searches for relative URLs referenced in the HTML documents created by WebFOCUS. |
| `FOCEXURL` | Runs and executes drill downs remotely. |
| `WEBVIEWER` | In WebFOCUS, enables on-demand paging, and invokes the WebFOCUS Viewer. |

## Developer Studio Specific Tasks

The following parameters control your Developer Studio environment and the display of your reports in Developer Studio.

| | |
|---|---|
| `LANG[UAGE]` | Specifies the National Language Support (NLS) environment. |
| `LINKMODE` | Specifies whether the WebFOCUS CGI is accessed or if the iWay Server is access directly when a procedure is run. |
| `MESSAGE` | Controls the display of informational messages. |
| `OFFLINE-FMT` | Determines the format of printed report output generated from a request submitted from the Command Console or created in the Text Editor. |
| `PAGESIZE` | Specifies the page size for printed output. |
| `PRINT` | determines whether report output is sent to your screen or to the printer. |
| `SHIFT` | Controls the use of “shift" strings. |

## Graph Tasks

The following parameters control the processing and display of graphs.

| | |
|---|---|
| 3D | Sets a chart to either three dimensions or two dimensions. |
| AUTOTICK | Sets the tick mark intervals for graphs. |
| BARNUMB | Places summary numbers at the end of bars on bar charts, or slices on pie charts. |
| BSTACK | Specifies whether the bars on a bar chart are stacked or placed side by side. |
| GRAPHEDIT | In WebFOCUS, displays a button with your output that accesses the Graph Editor. |
| GRAPHSERVURL | Creates a GIF file of the graph by sending an HTTP request to a servlet. |
| GRID | Draws a grid of parallel horizontal lines at the vertical class marks on the graph. |
| HAUTO | Performs automatic scaling of the horizontal axis for the given values. |
| HAXIS | Specifies the width, in characters, of the horizontal axis. |
| HCLASS | Specifies the horizontal interval mark when AUTOTICK is OFF. |
| HISTOGRAM | Draws a histogram instead of a curve when the values on the horizontal axis are not numeric. |
| HMAX | Sets the maximum value on the horizontal axis when automatic scaling is not used (HAUTO=OFF). |
| HMIN | Sets the minimum value on the horizontal axis when automatic scaling is not used (HAUTO=OFF). |
| HSTACK | Stacks the bars on a histogram instead of placing them side by side. |
| HTICK | Sets the horizontal axis interval mark when AUTOTICK is OFF. |
| LOOKGRAPH | Specifies a graph style. |
| VAUTO | Performs automatic scaling of the vertical axis for the given values. |
| VAXIS | Specifies the length of the vertical axis, in lines. |
| VCLASS | Specifies the vertical interval mark when AUTOTICK is OFF. |

| | |
|---|---|
| VGRID | Draws a grid at the horizontal and vertical class marks of the graph. |
| VMAX | Sets the maximum value on the vertical axis when automatic scaling is not used (VAUTO=OFF). |
| VMIN | Sets the minimum value on the vertical axis when automatic scaling is not used (VAUTO=OFF). |
| VTICK | Sets the vertical axis interval mark when AUTOTICK is OFF. |
| VZERO | Treats missing values on the vertical axis as zeros. |

## Memory Setup and Optimization Tasks

The following parameters control the memory and optimization of your application.

| | |
|---|---|
| AUTOINDEX | Retrieves data faster by automatically taking advantage of indexed fields in a FOCUS data source. |
| AUTOPATH | Dynamically selects an optimal retrieval path. |
| AUTOSTRATEGY | Determines when WebFOCUS stops the search for a key field specified in a WHERE or IF test. |
| BINS | Specifies the number of pages of core (blocks of 4,096 bytes) used for data source buffers. |
| CACHE | Stores 4K FOCUS data source pages in memory and buffers them between the data source and BINS. |
| COMPUTE | Controls the compile of computes. |
| ESTRECORDS | Passes the estimated number of records to be sorted in the request. |
| FIXRET[RIEVE] | Enables keyed retrieval. |
| FOCSTACK | Specifies the amount of space, in thousands of bytes, used by FOCUS commands waiting for execution. |
| RECORDLIMIT | Limits the number of records retrieved. |
| SQLTOPTTF | Enables the SQL Translator to generate TABLEF commands instead of TABLE commands. |
| TEMP | In WebFOCUS, assigns temporary files to a specific directory. |
| TEMPDIR | In Developer Studio, assigns temporary files to a specific directory. |

## Report Code and Content Tasks

The following parameters affect the content of a report.

| | |
|---|---|
| ALL | Handles missing segment instances in a report. |
| ALLOWCVTERR | Controls the display of a row of data that contains an invalid date format. |
| ASNAMES | Controls the FIELDNAME attribute in a HOLD Master File. |
| BUSDAYS | Specifies which days are considered business days. |
| CARTESIAN | Generates a report containing all combinations of non-related data instances in the case of a multi-path request. |
| CDN | Specifies punctuation used in numerical notation. |
| CENT-ZERO | Displays a leading zero in decimal-only numbers. |
| COMPUTE | Controls the compile of computes. |
| DATEDISPLAY | Controls the display of date format fields that contain the value zero. |
| DATEFNS | Activates year 2000-compliant versions of date subroutines. |
| DATETIME | In iWay, controls the format in which CREATE SYNONYM creates date-time columns in a Master File. |
| DEFCENT | Defines a default century for your application. |
| EMPTYREPORT | Controls the output generated when a report request retrieves zero records. |
| ERROROUT | Terminates a request and returns an error message when an error is encountered. |
| ESTRECORDS | Passes the estimated number of records to be sorted in the request. |
| EXTAGGR | Enables aggregation in an external sort. |
| EXTHOLD | Enables you to use an external sort to create HOLD files. |
| EXTSORT | Activates the external sorting feature. |
| FIELDNAME | Controls the use of long field names (66 characters). |
| FILE[NAME] | Specifies a file to be used, by default, in commands. |
| FILTER | In WebFOCUS, activates declared filters. |

| | |
|---|---|
| `FOC144` | Suppresses warning message FOC 144, which reads “Warning Testing in Independent sets of Data." |
| `FORMULTIPLE` | Allows you to include data in multiple FOR categories. |
| `HOLDATTR[S]` | Includes the TITLE and ACCEPT attributes from the original Master File in the HOLD Master File. |
| `JOINOPT` | Enables you to join two data sources containing different numeric data types. |
| `KEEPDEFINES` | Controls whether a virtual field created for a host or joined structure is retained after a JOIN command is run. |
| `LEADZERO` | Avoids the truncation of leading zeros. |
| `MULTIPATH` | Controls whether a parent segment is included in report output when selection tests are done on independent paths. |
| `NODATA` | Determines the character string that indicates missing data in a report. |
| `QUALCHAR` | Specifies the qualifying character to be used in qualified field names. |
| `SAVEMATRIX` | Saves the matrix from your request to protect it from being overwritten when using Dialogue Manager commands. |
| `SUMPREFIX` | Allows users to choose the answer set display order when using an external sort to perform aggregation of alphanumeric or smart date formats. |
| `TITLE` | Uses predefined column titles in the Master File as column titles in report output. |

## Report Layout and Display Tasks

The following parameters affect the layout and display of a report.

| | |
|---|---|
| ACCESSIBLE | Generates report output in HTML code which supports products for handicapped accessibility. |
| AUTODRILL | Enables you to automatically drill down to the next level of a dimension within the body of the report. |
| BOTTOMMARGIN | Sets the StyleSheet bottom boundary for report contents on a page. |
| BYPANEL | Controls the repetition of BY fields on a panel. |
| CENT-ZERO | Displays a leading zero in decimal-only numbers. |
| CSSURL | Links an HTML report to an external Cascading Style Sheet (CSS) file in order to style the report. |
| CURRENCY[SYM] | Specifies the symbol used to represent currency. |
| FOCFIRSTPAGE | Assigns a page number to the first page of output. |
| GRMERGE | Generates a three-dimensional graph that reflects all output data from a request containing both a BY and ACROSS phrase. |
| GRWIDTH | Places an output graph in an HTML table of a specified width, in cells. |
| HTMLCSS | Creates an inline Cascading Style Sheet command in the HTML page that displays the report output. |
| LEFTMARGIN | Sets the left boundary for report contents on a page. |
| LINES | Sets the maximum number of lines of printed output that appear on a page, from the heading at the top to the footing on the bottom. |
| ONLINE-FMT | Determines the format of report output. |
| ORIENTATION | Specifies the page orientation for styled reports. |
| PAGE[-NUM] | Controls the numbering of output pages. |
| PSPAGESETUP | Used with the PAGESIZE parameter, sets the paper size for a PDF or PostScript report. |
| QUALTITLES | Uses qualified column titles in report output when duplicate field names exist in a Master File. |

| | |
|---|---|
| `RECAP-COUNT` | Includes lines containing a value created with RECAP when counting the number of lines per page for printed output. |
| `RIGHTMARGIN` | Sets the right boundary for report contents on a page. |
| `RPAGESET` | Controls how the number of lines per printed page are determined when output contains text created with SUBFOOT and a field value created with RECAP. |
| `SPACES` | Sets the number of spaces between columns in a report. |
| `SQUEEZE` | Determines the column width in report output. |
| `STYLEMODE` | For large report output, displays output in multiple HTML tables where each table is a separate report page. |
| `STYLE[SHEET]` | Controls the format of report output by accepting or rejecting StyleSheet parameters. |
| `TARGETFRAME` | Includes the HTML code <BASE TARGET="*framename*"> in the heading of the HTML file that WebFOCUS displays in your browser. |
| `TOPMARGIN` | Sets the top boundary on a page for report output. |
| `UNITS` | Specifies the unit of measure for page margins, column positions, and column widths. |

## Security Tasks

The following parameters specify user access to data sources and procedures.

| | |
|---|---|
| `PASS` | Enables user access to a data source or stored procedure protected by Information Builders security. |
| `USER` | In WebFOCUS, enables user access to a data source or stored procedure protected by Information Builders security. |

# SET Parameter Syntax

This topic alphabetically lists the SET parameters that control the environment, with a description and their syntax.

| | |
|---|---|
| **Parameter:** | 3D |
| **Description:** | Sets a chart to either three dimensions or two dimensions. |
| **Syntax:** | `SET 3D = {ON\|OFF}`<br>where:<br>`ON`<br>Produces a three-dimensional chart. This is the default.<br>`OFF`<br>Produces a two-dimensional chart. |

| | |
|---|---|
| **Parameter:** | ACCESSIBLE |
| **Description:** | Generates report output in HTML code which supports products for handicapped accessibility and is compliant with Section 508 level accessibility requirements.<br>HTML code that supports products for handicapped accessibility automatically turns off pagination, column titles and page headings display only at the beginning of the report, and page footings are displayed only at the end of the report. |
| **Syntax:** | `SET ACCESSIBLE = {508\|OFF}`<br>where:<br>`508`<br>Generates report output in HTML code which supports products for handicapped accessibility and is compliant with Section 508 level accessibility requirements.<br>`OFF`<br>Does not generate report output that is compliant with Section 508 level accessibility requirements. This value is the default. |

| | |
|---|---|
| **Parameter:** | ALL |
| **Description:** | Handles missing segment instances in a report. |

| | |
|---|---|
| **Syntax:** | `SET ALL = {ON\|OFF\|PASS}`<br>where:<br>`ON`<br>Includes missing segment instances in a report when fields in the segment are not screened by WHERE or IF criteria in the request. The missing field values are denoted by the NODATA character, set with the NODATA parameter (see NODATA).<br>`OFF`<br>Omits missing segment instances from a report. This value is the default.<br>`PASS`<br>Includes missing segment instances in a report regardless of WHERE or IF criteria in the request. This option is not supported when MULTIPATH = COMPOUND (see MULTIPATH). |

| | |
|---|---|
| **Parameter:** | ALLOWCVTERR |
| **Description:** | This parameter applies to non-FOCUS data sources when converting from the way the date is stored (ACTUAL attribute) to the way it is formatted (FORMAT or USAGE attribute).<br><br>Controls the display of a row of data that contains an invalid date format (formerly called a smart date). When it is set to ON, the invalid date format is returned as the base date or a blank, depending on the settings for the MISSING and DATEDISPLAY parameters. |
| **Syntax:** | `SET ALLOWCVTERR = {ON\|OFF}`<br><br>where:<br><br>`ON`<br>Displays a row of data that contains an invalid date format. When ALLOWCVTERR is set to ON, the display of invalid dates is determined by the settings of the MISSING attribute and DATEDISPLAY command. The results are explained in the following table:<br><br>`OFF`<br>Does not display a row of data that contains an invalid date format and generates an error message. This value is the default. |

| DATEDISPLAY | MISSING | RESULT |
|---|---|---|
| OFF | OFF | A blank is returned. |
| | ON | The value of the NODATA character (a period, by default) is returned. (See NODATA). |
| ON | OFF | The value of the NODATA character (a period, by default) is returned. |
| | ON | The base date is returned (either December 31, 1900, for dates with YMD or YYMD format; or January 1901, for dates with YM, YYM, YQ, or YYQ format). |

| | |
|---|---|
| **Parameter:** | ASNAMES |
| **Description:** | Controls the FIELDNAME attribute in a HOLD Master File. When an AS phrase is used in a report request, the specified literal is used as a field name in a HOLD file. Also controls how field names are specified for the values of an ACROSS field when a HOLD file is created. |
| **Syntax:** | `SET ASNAMES = {ON\|OFF\|FOCUS}`<br>where:<br>`ON`<br>Uses the AS phrase for the field name, and controls the way ACROSS fields are named in HOLD files in any format.<br>`OFF`<br>Does not use the AS phrase for the field name, or affect the way ACROSS fields are named.<br>`FOCUS`<br>Uses the AS phrase for the field name, and controls the way ACROSS fields are named in HOLD files only. This value is the default. |

| | |
|---|---|
| **Parameter:** | AUTODRILL |
| **Description:** | This parameter applies to OLAP.<br><br>Enables the end user to automatically drill down to the next level of a dimension within the body of the current report. That is, the server automatically generates a hyperlink (the <A HREF tag) in the HTML code for every output data item described by the Master File WITHIN attribute. This occurs when AUTODRILL is set to ON.<br><br>If there is no other level, and the selected field is part of a BY command in the request (that is, it is a vertical sort field), the report is generated without a column for that dimension. If there is no other level, and the selected field is part of an ACROSS command in the request (it is a horizontal sort field), the report is generated without a row for that dimension. AUTODRILL must act on a sort field (that is also defined as a dimension in the Master File).<br><br>If the OLAP Control Panel is open, the output data item selected on the screen is added to the Selection Criteria list in the panel, enabling the end user to choose multiple items before running the drill-down report.<br><br>If the OLAP Control Panel is not open, the drill-down report runs automatically, based on the output data item selected on the screen. |
| **Syntax:** | `SET AUTODRILL = {ON\|OFF\|ALL}`<br><br>where:<br><br>`ON`<br>Automatically generates a hyperlink in the HTML source code for every output data item described by the Master File WITHIN attribute. This value is the default.<br><br>`OFF`<br>Disables the automatic generation of hyperlinks.<br><br>`ALL`<br>Allows you to drill down on any value in the report by creating every value in the report as a hyperlink, including Measures. Drilling on a measure generates a more detailed report based on the lowest drill down (BY) field. |

| | |
|---|---|
| **Parameter:** | AUTOINDEX |
| **Description:** | Retrieves data faster by automatically taking advantage of indexed fields in most cases where TABLE requests contain equality or range tests on those fields. Applies only to FOCUS data sources.<br><br>AUTOINDEX is never performed when the TABLE request contains an alternate file view, for example, TABLE FILE *filename.filename*. Indexed retrieval is not performed when the TABLE request contains BY HIGHEST or BY LOWEST phrases and AUTOINDEX is ON. |
| **Syntax:** | `SET AUTOINDEX = {ON\|OFF}`<br><br>where:<br><br>`ON`<br>Uses indexed retrieval when possible.<br><br>`OFF`<br>Uses indexed retrieval only when explicitly specified via an indexed view, for example, TABLE FILE *filename.indexed-fieldname*. This value is the default. |

| | |
|---|---|
| **Parameter:** | AUTOPATH |
| **Description:** | Dynamically selects an optimal retrieval path for accessing a FOCUS data source by analyzing the data source structure and the fields referenced, and choosing the lowest possible segment as the entry point. Use AUTOPATH only if your field is not indexed. |
| **Syntax:** | `SET AUTOPATH = {ON\|OFF}`<br><br>where:<br><br>`ON`<br>Dynamically selects an optimal retrieval path. This value is the default.<br><br>`OFF`<br>Uses sequential data retrieval. The end user controls the retrieval path through *filename.segname*. |

| | |
|---|---|
| **Parameter:** | AUTOSTRATEGY |
| **Description:** | Determines when WebFOCUS stops the search for a key field specified in a WHERE or IF test. When set to ON, the search ends when the key field is found, optimizing retrieval speed. When set to OFF, the search continues to the end of the data source. |
| **Syntax:** | `SET AUTOSTRATEGY = {ON\|OFF}`<br>where:<br>`ON`<br>Stops the search when a match is found. This value is the default.<br>`OFF`<br>Searches the entire data source. |

| | |
|---|---|
| **Parameter:** | AUTOTICK |
| **Description:** | Automatically sets the tick mark intervals for graphs. (See also HTICK and VTICK.) |
| **Syntax:** | `SET AUTOTICK = {ON\|OFF}`<br>where:<br>`ON`<br>Automatically sets the tick mark intervals. This value is the default.<br>`OFF`<br>Does not set the tick mark intervals; you must set them with parameters including HTICK and VTICK. |

| | |
|---|---|
| **Parameter:** | BARNUMB |
| **Description:** | Places summary values at the end of bars on bar charts, or slices on pie charts. |
| **Syntax:** | `SET BARNUMB = {ON\|OFF}`<br>where:<br>`ON`<br>Places the summary values at the ends of bars on bar charts, or slices on pie charts.<br>`OFF`<br>Does not include summary values. This value is the default. |

| | |
|---|---|
| **Parameter:** | BASEURL |
| **Description:** | Specifies a default location where your browser searches for relative URLs referenced in the HTML documents created by WebFOCUS. This allows you to hyperlink to files, images, and Java files using only their file names rather than their full URLs. |
| **Syntax:** | `SET BASEURL = url`<br>where:<br>*url*<br>Is the fully qualified directory in which additional HTML files, graphics files, and Java applet class files reside. The URL must begin with http:// and end with a slash (/). |

| | |
|---|---|
| **Parameter:** | BINS |
| **Description:** | Specifies the number of pages of core (blocks of 4,096 bytes) used for data source buffers. You can vary BINS from 13 to 63 pages depending on the size of your core. |
| **Syntax:** | `SET BINS = nn`<br>where:<br>*nn*<br>Is the number of core pages used for data source buffers. Valid values are 13 to 64. The default value is two-thirds of the core remaining after you start, about 64 pages. |

| | |
|---|---|
| **Parameter:** | BOTTOMMARGIN |
| **Description:** | Sets the StyleSheet bottom boundary for report contents on a page.<br>This parameter applies only to PostScript and PDF report formats. |
| **Syntax:** | `SET BOTTOMMARGIN = {value\|.250}`<br>where:<br>*value*<br>Is the bottom boundary for report contents on a page. If you do not supply a value, the default is .250 inches. |

| | |
|---|---|
| **Parameter:** | BUSDAYS |
| **Description:** | Specifies which days are considered business days and which days are not if your business does not follow the traditional Monday through Friday week. |
| **Syntax:** | `SET BUSDAYS = {week\|_MTWTF_}`<br>where:<br>*week*<br>Is SMTWTFS, representing the days of the week. Any day that you do not wish to designate as a business day must be replaced with an underscore in that day's designated place.<br>If a letter is not in its correct position, or if you replace a letter with a character other than an underscore, you receive an error message. The default value is _MTWTF_. |

| | |
|---|---|
| **Parameter:** | BYPANEL |
| **Description:** | This parameter applies only to HOTSCREEN.<br>Controls the repetition of sort (BY) field values on panels. |
| **Syntax:** | `SET BYPANEL = option`<br>where:<br>*option*<br>Is one of the following:<br>`ON` repeats the sort field values on each report panel.<br>`OFF` does not repeat sort field values on each report panel. Fields are displayed only on the first panel, and columns may split between panels. This value is the default.<br>`0` does not repeat sort field values on each report panel, and columns do not split between panels.<br>*n* repeats *n* columns of sort fields on each report panel. The value for *n* can be equal to or less than the total number of sort fields specified in the request. |

| | |
|---|---|
| **Parameter:** | CACHE |
| **Description:** | Stores 4K FOCUS data source pages in memory and buffers them between the data source and BINS.<br><br>When a procedure calls for a read of a data source page, FOCUS first searches BINS, then cache memory, and then the data source on disk. If the page is found in cache, FOCUS does not have to perform an I/O to disk.<br><br>When a procedure calls for a write of a data source page, the page is written from BINS to disk. The updated page is also copied into cache memory so that the cache and disk versions remain the same. Unlike reads, cache memory does not save disk I/Os for write procedures.<br><br>FOCSORT pages are also written to cache; when the cache becomes full, they are written to disk. For optimal results, set cache to hold the entire data source plus the size of FOCSORT for the request. To estimate the size of FOCSORT for a given request, issue the ? STAT command (discussed in Chapter 11, *Testing and Debugging a Procedure*), then add the number of SORTPAGES listed to the number of data source pages in memory. Issue a SET CACHE command for that amount. If cache is set to 50, 50 4K pages of contiguous storage are allocated to cache. The maximum number of cache pages can be set at installation. For more information, see Technical Memo 7838.1, *Setting the Maximum Number of Cache Pages*.<br><br>To clear the CACHE setting issue a SET CACHE = *n* command. This command flushes the buffer; that is, everything in cache memory is lost. |
| **Syntax:** | `SET CACHE = {0\|n}`<br><br>where:<br><br>`0`<br>Allocates no space to cache; cache is inactive. This value is the default.<br><br>*n*<br>Is the number of 4K pages of contiguous storage allocated to cache memory. The minimum is two pages; the maximum is determined by the amount of memory available. |

| | |
|---|---|
| **Parameter:** | CARTESIAN |
| **Description:** | Applies to requests containing PRINT or LIST commands.<br><br>Generates a report containing all combinations of non-related data instances in the case of a multi-path request. |
| **Syntax:** | `SET CARTESIAN = {ON\|OFF}`<br><br>where:<br><br>`ON`<br>Generates a report with non-related records.<br><br>`OFF`<br>Disables the Cartesian product. This value is the default. |

| | |
|---|---|
| **Parameter:** | CDN |
| **Description:** | Specifies punctuation used in numerical notation. When set to ON, a comma marks the decimal position, and periods separate groups of three significant digits.<br><br>Continental Decimal Notation (CDN) is supported for output in TABLE requests. It is not supported in DEFINE or COMPUTE commands. |
| **Syntax:** | `SET CDN = {ON\|OFF\|SPACE\|QUOTE}`<br><br>where:<br><br>`ON`<br>Uses CDN. For example, the number 3,045,000.76 is represented as 3.045.000,76.<br><br>`OFF`<br>Turns CDN off. For example, the number 3,045,000.76 is represented as 3,045,000.76. This value is the default.<br><br>`SPACE`<br>Separates groups of three significant digits with a space instead of a comma, and marks a decimal position with a comma instead of a period. For example, the number 3,045,000.76 is represented as 3 045 000,76.<br><br>`QUOTE`<br>Separates groups of three significant digits with a single quotation mark instead of a comma, and marks a decimal position with a comma instead of a period. For example, the number 3,045,000.76 is represented as 3’045’000,76. |

| | |
|---|---|
| **Parameter:** | CENT-ZERO |
| **Description:** | Displays a leading zero in decimal-only numbers. The setting of CDN determines whether a decimal point or comma is the decimal separator. |
| **Syntax:** | `SET CENT-ZERO = {ON\|OFF}`<br>where:<br>`ON`<br>Displays decimal-only numbers with a leading zero.<br>`OFF`<br>Does not display decimal-only numbers with a leading zero. This value is the default. |

| | |
|---|---|
| **Parameter:** | COMPUTE |
| **Description:** | Controls the compile of COMPUTE calculations when a request is executed. |
| **Syntax:** | `SET COMPUTE = {NEW\|OLD}`<br>where:<br>`NEW`<br>Compiles COMPUTE calculations when a request is executed. This value is the default.<br>`OLD`<br>Does not compile COMPUTE calculations when a request is executed; the old logic is used. |

| | |
|---|---|
| **Parameter:** | CSSURL |
| **Description:** | Links an HTML report to an external Cascading Style Sheet (CSS) file in order to style the report. |
| **Syntax:** | `SET CSSURL = link`<br>where:<br>*link*<br>Is the URL location of the CSS file. This can be an absolute or relative link. |

| | |
|---|---|
| **Parameter:** | CURRENCY[SYM] |
| **Description:** | Specifies a symbol used to represent currency. |
| **Syntax:** | `SET CURRENCY[SYM] = {'string'\|$}`<br>where:<br>*string*<br>Is a string of 1 to 4 characters representing currency. The default value is a dollar sign ($). |

| | |
|---|---|
| **Parameter:** | DATEDISPLAY |
| **Description:** | Controls the display of date format fields that contain the value zero. When set to on, the base date is displayed for those fields. By default, the value zero in a date format field such as YYMD is returned as a blank. |
| **Syntax:** | `SET DATEDISPLAY = {ON\|OFF}`<br>where:<br>ON<br>Returns the base date for a date format field containing the value zero.<br>OFF<br>Returns a blank for a date format field containing the value zero. This is the default. |

| | |
|---|---|
| **Parameter:** | DATEFNS |
| **Description:** | Activates year 2000-compliant versions of date subroutines. |
| **Syntax:** | `SET DATEFNS = {ON\|OFF}`<br>where:<br>ON<br>Activates the date subroutines that support year-2000 dates. This value is the default.<br>OFF<br>Deactivates the date subroutines that support year-2000 dates. |

| | |
|---|---|
| **Parameter:** | DATETIME |
| **Description:** | In iWay, controls the format in which CREATE SYNONYM creates date-time columns in a Master File. |
| **Syntax:** | `SET DATETIME = option`<br>where:<br>*option*<br>Is one of the following:<br>STARTUP is the time and date when you began your FOCUS session. This setting is the default.<br>CURRENT\|NOW changes each time it is interrogated. For example, if your batch job starts before midnight at 11:59 P.M., it won't complete until the next day. If DATETIME is set to NOW or CURRENT, any reference to the variable gives the current date, not the date when the job started.<br>RESET freezes the date and time of the current run for the rest of the session or until another SET DATETIME command is issued. |

| | |
|---|---|
| **Parameter:** | DEFCENT |
| **Description:** | Defines a default century globally or on a field-level for an application that does not contain an explicit century. DEFCENT is used in conjunction with YRTHRESH to interpret the current century according to the given values. When assigned globally, the time span created by these parameters applies to every 2-digit year used by the application unless you specify file-level or field-level values. (See *YRTHRESH* on page 15-77.)<br>**Note:** This same result can be achieved by including the FDEFCENT and FYRTHRESH attributes in the Master File. |
| **Syntax:** | `SET DEFCENT = {cc\|19}`<br>where:<br>*cc*<br>Is the default century. If you do not supply a value, *cc* defaults to 19, for the twentieth century. |

| | |
|---|---|
| **Parameter:** | DEFINES |
| **Description:** | This parameter applies to Windows NT/2000/XP and UNIX operating systems only.<br>Increases the speed of calculations in virtual fields by compiling virtual fields into machine code. |
| **Syntax:** | `SET DEFINES = {COMPILED\|OLD}`<br>where:<br>`COMPILED`<br>Compiles virtual fields. This value is the default.<br>`OLD`<br>Does not compile virtual fields. |

| | |
|---|---|
| **Parameter:** | DTSTRICT |
| **Description:** | Controls the use of strict processing. Strict processing checks date-time values when they are input by an end user, read from a transaction file, displayed, or returned by a subroutine to ensure that they represent a valid date and time. For example, a numeric month must be between 1 and 12, and the day must be within the number of days for the specified month. |
| **Syntax:** | `SET DTSTRICT = {ON\|OFF}`<br>where:<br>`ON`<br>Invokes strict processing. This value is the default. If you attempt to enter a value that is not valid, the following message displays:<br>`(FOC177) INVALID DATE CONSTANT: dt_constant`<br>`OFF`<br>Does not invoke strict processing. Date-time components can have any value within the constraint of the number of decimal digits allowed in the field. For example, if the field is a two-digit month, the value can be 12 or 99, but not 115. |

| | |
|---|---|
| **Parameter:** | EMPTYREPORT |
| **Description:** | Controls the output generated when a TABLE request retrieves zero records.<br><br>**Note:** EMPTYREPORT is not supported with TABLEF. When a TABLEF request retrieves zero records, an empty report is generated. |
| **Syntax:** | `SET EMPTYREPORT = {ON\|OFF}`<br><br>where:<br><br>`ON`<br>Generates an empty report when zero records are found, and includes column titles, and page and report headings (if they were specified).<br><br>`OFF`<br>Generates a message when zero records are found. This value is the default. |

| | |
|---|---|
| **Parameter:** | ERROROUT |
| **Description:** | Terminates a job and returns an error message when an error is encountered in a request. |
| **Syntax:** | `SET ERROROUT = {ON\|OFF}`<br><br>where:<br><br>`ON`<br>Terminates a job and returns an error message when an error is encountered.<br><br>`OFF`<br>Continues processing when an error is encountered. |

| | |
|---|---|
| **Parameter:** | ESTRECORDS |
| **Description:** | Passes the estimated number of records to be sorted in the request.<br><br>ESTRECORDS can only be set with the ON TABLE SET command within the TABLE, MATCH, or GRAPH request. |
| **Syntax:** | `ON TABLE SET ESTRECORDS n`<br><br>where:<br><br>*n*<br>Is the estimated number of records to be sorted. |

| | |
|---|---|
| **Parameter:** | EUROFILE |
| **Description:** | Activates the data source that contains information for the currency you want to convert. This setting can be changed during a session to access a different currency data source. This parameter cannot be issued in a report request.<br><br>**Note:** You cannot set any additional parameters on the same line as EUROFILE. WebFOCUS ignores any other parameters specified on the same line. |
| **Syntax:** | `SET EUROFILE = {ddname\|OFF}`<br><br>where:<br><br>*ddname*<br>Is the name of the Master File for the currency data source you want to use. The ddname must refer to a read-only data source accessible by WebFOCUS. By default, there is no data source assigned.<br><br>OFF<br>Deactivates the current currency data source and removes it from memory. |

| | |
|---|---|
| **Parameter:** | EXTAGGR |
| **Description:** | Enables aggregation in an external sort. |
| **Syntax:** | `SET EXTAGGR = {ON\|OFF\|NOFLOAT}`<br><br>where:<br><br>ON<br>Allows aggregation by an external sort. This value is the default.<br><br>OFF<br>Does not allow aggregation by an external sort.<br><br>NOFLOAT<br>Allows aggregation if there are no floating point data fields present. |

| | |
|---|---|
| **Parameter:** | EXTHOLD |
| **Description:** | Enables you to use an external sort to create HOLD files. |
| **Syntax:** | `SET EXTHOLD = {OFF\|ON}`<br>where:<br>`ON`<br>Enables HOLD files by an external sort. This value is the default.<br>`OFF`<br>Disables HOLD files by an external sort. |

| | |
|---|---|
| **Parameter:** | EXTSORT |
| **Description:** | Activates the external sorting feature.<br>If the report can be processed entirely in memory, external sorting does not occur. In order to determine if the report can be processed in memory, issue the ? STAT command. The SORT USED parameter indicates whether the report can be processed from memory. |
| **Syntax:** | `SET EXTSORT = {ON\|OFF}`<br>where:<br>`ON`<br>Enables the selective use of a dedicated external sorting product to sort reports. This value is the default.<br>`OFF`<br>Uses the internal sorting procedure to sort all reports. |

| | |
|---|---|
| **Parameter:** | FIELDNAME |
| **Description:** | Controls the use of long field names up to 66 characters. |
| **Syntax:** | `SET FIELDNAME = {NEW\|NOTRUNC\|OLD}`<br>where:<br>`NEW`<br>Supports long field names. This is the default.<br>`NOTRUNC`<br>Does not support unique truncations. Supports long and qualified field names.<br>`OLD`<br>Limits field names to 12 characters. Qualified field names are not supported. |

| | |
|---|---|
| **Parameter:** | FILE[NAME] |
| **Description:** | Specifies a file to be used, by default, in commands. When you set a default file name, you can use that file without specifying its name. |
| **Syntax:** | `SET FILE[NAME] = filename`<br>where:<br>*filename*<br>Is a default file to be used in commands. |

| | |
|---|---|
| **Parameter:** | FILTER |
| **Description:** | In WebFOCUS, activates declared filters. |
| **Syntax:** | `SET FILTER= {*\|xx[yy zz]} IN file {ON\|OFF}`<br>where:<br>`*`<br>Activates all declared filters. This value is the default.<br>*`xx, yy, zz`*<br>Are the names of filters as declared in the NAME = syntax of the FILTER FILE command.<br>*`file`*<br>Is the name of the data source you are assigning screening conditions to.<br>`ON`<br>Activates all (*) or specifically named filters for the data source. The maximum number of filters you can activate for a data source is limited by the number of WHERE/IF phrases the filters contain, not to exceed the limit of WHERE/IF criteria in any single report request.<br>`OFF`<br>Deactivates (*) or specifically named filters for the data source. This value is the default. |

| | |
|---|---|
| **Parameter:** | FIXRET[RIEVE] |
| **Description:** | Enables keyed retrieval. That is, the retrieval process stops when an equality or range test on a key holds true. |
| **Syntax:** | `SET FIXRET[RIEVE] = {OFF\|ON}`<br>where:<br>`ON`<br>Enables keyed retrieval. This is the default value.<br>`OFF`<br>Disables keyed retrieval. |

| | |
|---|---|
| **Parameter:** | FOC144 |
| **Description:** | Suppresses warning message FOC 144, which reads “Warning Testing in Independent sets of Data." |
| **Syntax:** | `SET FOC144 = {NEW\|OLD}`<br>where:<br>`NEW`<br>Does not display the warning message. This value is the default.<br>`OLD`<br>Displays the warning message. |

| | |
|---|---|
| **Parameter:** | FOCEXURL |
| **Description:** | Runs and executes drill downs remotely. The drill down program can be on your local machine or on a remote machine. |
| **Syntax:** | `SET FOCEXURL = path`<br>where:<br>*path*<br>Is the location of the WebFOCUS CGI. |

| | |
|---|---|
| **Parameter:** | FOCFIRSTPAGE |
| **Description:** | Assigns a page number to the first page of output. |
| **Syntax:** | `SET FOCFIRSTPAGE = {n\|1/&FOCNEXTPAGE}`<br>where:<br>*n*<br>Is the number to be assigned to the first page of output. Valid values are integers with 1 to 6 characters. The default is 1.<br>`&FOCNEXTPAGE`<br>Is a variable whose value is determined by the last page number used by the last report. Its value is one more than the last page number used in the last report. |

| | |
|---|---|
| **Parameter:** | FOCSTACK |
| **Description:** | Specifies the amount of space, in thousands of bytes, used by FOCUS commands waiting for execution. |
| **Syntax:** | `SET FOCSTACK [SIZE] = {n\|8}`<br>where:<br>*n*<br>Is the maximum amount, in thousands of bytes, that can be used by FOCSTACK. The maximum value depends on your region size.<br>8<br>Allows 8000 bytes to be used by FOCSTACK. This value is the default. |

| | |
|---|---|
| **Parameter:** | FORMULTIPLE |
| **Description:** | Allows you to include the same value of a FOR field—whether as a single tag value, as part of a range, or in a calculation— in multiple rows of the FML matrix. |
| **Syntax:** | `SET FORMULTIPLE = {ON\|OFF}`<br>where:<br>ON<br>Allows you to include a record in multiple FOR categories.<br>OFF<br>Does not allows you to include a record in multiple FOR categories. If a record is used in multiple FOR categories, a period is displayed in place of the data in all but one of the FOR categories. |

| | |
|---|---|
| **Parameter:** | GRAPHEDIT |
| **Description:** | In WebFOCUS, displays a button with your output that accesses the Graph Editor. |
| **Syntax:** | `SET GRAPHEDIT = {ON\|OFF\|SERVER}`<br>where:<br>`ON`<br>Displays a button in the upper-right corner of your window that accesses the Graph Editor.<br>`OFF`<br>Does not display a button that accesses the Graph Editor.<br>`SERVER`<br>Generates a server-side graph. A GIF graph is created on the Web server and the image is returned to the client in your browser. This is the default. |

| | |
|---|---|
| **Parameter:** | GRAPHSERVURL |
| **Description:** | Creates a GIF file of the graph by sending an HTTP request to a servlet. The GIF file is saved on the WebFOCUS Reporting Server, where it can be used as input for an embedded image in a PDF report. It does not require running Java on the WebFOCUS Reporting Server, only on the remote machine that runs the servlet. This will work on any platform that has TCP/IP support, including OS/390. |
| **Syntax:** | `SET GRAPHSERVURL = url`<br>where:<br>*`url`*<br>Is the URL of the servlet that creates the .gif file. |

| | |
|---|---|
| **Parameter:** | GRID |
| **Description:** | Draws a grid on the graph at the horizontal and vertical class marks. |
| **Syntax:** | `SET GRID = {ON\|OFF}`<br>where:<br>`ON`<br>Draws a grid of lines at the horizontal and vertical class marks on the graph.<br>`OFF`<br>Does not draw a grid of lines at the horizontal and vertical class marks on the graph. This is the default. |

| | |
|---|---|
| **Parameter:** | GRMERGE |
| **Description:** | Generates a three-dimensional graph that reflects all output data from a request containing both a BY and ACROSS phrase, or creates separate graphs that collectively reflect all output data. When GRMERGE is OFF, a request containing both a BY and an ACROSS phrase generates two graphs. When it is ON, one graph results.<br>In the Graph Assistant, you can also set GRMERGE to ON by selecting a second horizontal axis in the Across tab, and selecting the *Merge graphs into a single graph* radio button. In OLAP graphs, this parameter can not be controlled in the OLAP Control Panel. |
| **Syntax:** | `SET GRMERGE = {ON\|OFF}`<br>where:<br>`ON`<br>Generates a single three-dimensional graph that reflects all output data. This value is the default.<br>`OFF`<br>Generates separate graphs that collectively reflect all output data. |

| | |
|---|---|
| **Parameter:** | GRWIDTH |
| **Description:** | This parameter applies to OLAP graphs and regular report requests. However, the parameter is not under user control in the OLAP Control Panel.<br><br>Places an output graph in an HTML table of a specified width, in cells. GRWIDTH is applied when you choose to display multiple graphs from the Graph Assistant in columns. |
| **Syntax:** | `SET GRWIDTH = {n\|0}`<br><br>where:<br><br>*n*<br>Is the width, in cells, of an HTML table in which a graph is displayed. Valid values are between 0 and 512. If the parameter is set to 0, a Java applet is generated displaying multiple graphs on one page, one beneath another. The default value is 0. |

| | |
|---|---|
| **Parameter:** | GTREND |
| **Description:** | Specifies the use of basic linear regression to alter the X and Y axis values in a SCATTER graph. Basic linear regression involves the average of the summation of X and Y axis values to determine a linear equation that expresses the trend of the scatter diagram. |
| **Syntax:** | `SET GTREND={ON\|OFF}`<br><br>where:<br><br>`ON`<br>Uses basic linear regression to alter the X and Y axis values.<br><br>`OFF`<br>Does not alter the X and Y axis values. This value is the default. |

| | |
|---|---|
| **Parameter:** | HAUTO |
| **Description:** | Performs automatic scaling of the horizontal axis for the given values. If HAUTO is OFF, the horizontal axis must be scaled using the HMAX and HMIN parameters. (See HMAX and HMIN.) |
| **Syntax:** | `SET HAUTO = {ON\|OFF}`<br>where:<br>`ON`<br>Scales the horizontal axis automatically. This value is the default.<br>`OFF`<br>Does not scale the horizontal axis automatically. The end user must supply values for HMAX and HMIN. |

| | |
|---|---|
| **Parameter:** | HAXIS |
| **Description:** | Specifies the width, in characters, of the horizontal axis. This parameter applies to graphs generated offline. HAXIS is ignored for online displays since WebFOCUS automatically adjusts the width of the graph to the width of the page. |
| **Syntax:** | `SET HAXIS = {n\|770}`<br>where:<br>*n*<br>Is the width, in characters, of the horizontal axis. Acceptable values are integers between 20 and 770. The default value is 770. |

| | |
|---|---|
| **Parameter:** | HCLASS |
| **Description:** | Specifies the horizontal interval mark when AUTOTICK is OFF. See also HTICK. |
| **Syntax:** | `SET HCLASS = {n\|.00}`<br>where:<br>*n*<br>Is a number specifying the horizontal interval mark. The default value is .00. |

| | |
|---|---|
| **Parameter:** | HDAY |
| **Description:** | Specifies the holiday file from which to retrieve dates that are considered holidays. The file must be named HDAY*xxxx*, where *xxxx* is four letters. |
| **Syntax:** | `SET HDAY = xxxx`<br>where:<br>*xxxx*<br>Are the letters in the name of the holiday file, named HDAY*xxxx*, after HDAY. This string must be four characters long. |

| | |
|---|---|
| **Parameter:** | HISTOGRAM |
| **Description:** | Draws a histogram instead of a curve when the values on the horizontal axis are not numeric. |
| **Syntax:** | `SET HISTOGRAM = {ON\|OFF}`<br>where:<br>ON<br>Draws a histogram instead of a curve when the values on the horizontal axis are not numeric.<br>OFF<br>Draws a curve when the values on the horizontal axis are not numeric. |

| | |
|---|---|
| **Parameter:** | HMAX |
| **Description:** | Sets the maximum value on the horizontal axis when automatic scaling is not used (HAUTO=OFF). |
| **Syntax:** | `SET HMAX = {nnn\|.00}`<br>where:<br>*nnn*<br>Is the maximum value on the horizontal axis when HAUTO=OFF. The default value is .00. |

| | |
|---|---|
| **Parameter:** | HMIN |
| **Description:** | Sets the minimum value on the horizontal axis when automatic scaling is not used (HAUTO=OFF). |
| **Syntax:** | `SET HMIN = {nnn\|.00}`<br>where:<br>*nnn*<br>Is the minimum value on the horizontal axis when HAUTO=OFF. The default value is .00. |

| | |
|---|---|
| **Parameter:** | HOLDATTR[S] |
| **Description:** | Includes the TITLE and ACCEPT attributes from the original Master File in the HOLD Master File. This setting does not affect the way fields are named in the HOLD Master File. |
| **Syntax:** | `SET HOLDATTR = {ON\|OFF\|FOCUS}`<br>where:<br>`ON`<br>Includes the TITLE attribute from the original Master File in HOLD Master Files for HOLD files of any format. The ACCEPT attribute is included in the HOLD Master File when the HOLD file is in FOCUS format.<br>`OFF`<br>Does not include the TITLE or ACCEPT attributes from the original Master File in the HOLD Master File.<br>`FOCUS`<br>Includes the TITLE and ACCEPT attributes in HOLD Master Files when the HOLD file is in FOCUS format. This value is the default. |

| | |
|---|---|
| **Parameter:** | HOLDFORMAT |
| **Description:** | Determines the default format for HOLD files. This value can be overridden for an individual HOLD file by issuing the ON HOLD FORMAT command in a request. |
| **Syntax:** | `SET HOLDFORMAT = {BINARY\|ALPHA}`<br>where:<br>`BINARY`<br>Creates HOLD files in binary format. This value is the default.<br>`ALPHA`<br>Creates HOLD fines in ALPHA format. |

| | |
|---|---|
| **Parameter:** | HOLDLIST |
| **Description:** | Determines what fields in a report request are included in a HOLD or PCHOLD file. |
| **Syntax:** | `SET HOLDLIST = {PRINTONLY\|ALL}`<br>where:<br>`PRINTONLY`<br>Includes only those fields in the HOLD file that are specified in the report request.<br>`ALL`<br>Includes all verb object fields referenced in a request in the HOLD file, including both computed fields and fields referenced in a COMPUTE command. This value is the default. (OLD may be used as a synonym for ALL.) |

| | |
|---|---|
| **Parameter:** | HOLDMISS |
| **Description:** | Determines the way missing data is stored in a HOLD file. This allows you to distinguish between missing data and default data in HOLD files.<br><br>By default, a zero or blank is produced in a HOLD file when data is missing. When HOLDMISS is set to ON, a character string is saved in the HOLD file in place of missing data. The character string is determined by the NODATA parameter. |
| **Syntax:** | `SET HOLDMISS = {OFF\|ON}`<br><br>where:<br><br>`OFF`<br>Stores missing data in a HOLD file as a zero or blank.<br><br>`ON`<br>Stores missing data in a HOLD file as a period. |

| | |
|---|---|
| **Parameter:** | HOLDSTAT |
| **Description:** | Includes the comments and DBA information in HOLD and PCHOLD Master Files. This information is found in the HOLDSTAT ERRORS file supplied by Information Builders, or in a user-specified file. |
| **Syntax:** | `SET HOLDSTAT = {ON\|OFF\|name}`<br><br>where:<br><br>`ON`<br>Derives comments and DBA information from the holdstat.mas or errors.mas file in UNIX and NT. In OS/390, this information is derived from the member HOLDSTAT in the PDS allocated to the ddname MASTER or ERRORS in OS/390.<br><br>`OFF`<br>Does not include information from the HOLDSTAT file in the HOLD Master File. This value is the default.<br><br>*name*<br>Specifies a HOLDSTAT file, created by the end user, whose information is included in the HOLD Master File. |

| | |
|---|---|
| **Parameter:** | HSTACK |
| **Description:** | Stacks the bars on a histogram instead of placing them side by side. |
| **Syntax:** | `SET HSTACK = {ON\|OFF}`<br>where:<br>`ON`<br>Stacks the bars on a histogram.<br>`OFF`<br>Places the bars on a histogram side by side. This value is the default. |

| | |
|---|---|
| **Parameter:** | HTICK |
| **Description:** | Sets the horizontal axis interval mark when AUTOTICK is OFF. See also HCLASS. |
| **Syntax:** | `SET HTICK = {nnn\|.00}`<br>where:<br>*nnn*<br>Is the horizontal axis interval mark when AUTOTICK=OFF. The default value is .00. |

| | |
|---|---|
| **Parameter:** | HTMLCSS |
| **Description:** | Creates an internal Cascading Style Sheets command in the HTML display page. |
| **Syntax:** | `SET HTMLCSS = {ON\|OFF}`<br>where:<br>`ON`<br>Creates an internal CSS command in the HTML page that displays the report output.<br>`OFF`<br>Does not create an internal CSS command in the HTML page that displays the report output. This value is the default. |

| | |
|---|---|
| **Parameter:** | INDEX |
| **Description:** | This parameter applies to indexes created in FOCUS data sources using early WebFOCUS releases.<br><br>Determines the indexing scheme used for indexes. Indexes are fields specified with FIELDTYPE=I keywords in the Master Files. |
| **Syntax:** | `SET INDEX = {NEW\|OLD}`<br><br>where:<br><br>`NEW`<br>Creates a binary tree index. This value is the default.<br><br>`OLD`<br>Creates a hash index. |

| | |
|---|---|
| **Parameter:** | JOINOPT |
| **Description:** | Enables you to join two data sources containing different numeric data types. One field is converted to the data type of the other. |
| **Syntax:** | `SET JOINOPT = {NEW\|OLD}`<br><br>where:<br><br>`NEW`<br>Enables data sources containing different numeric data types to be joined.<br><br>`OLD`<br>Requires that numeric fields in a data source be in the same format to be joined. This value is the default. |

| | |
|---|---|
| **Parameter:** | KEEPDEFINES |
| **Description:** | Controls whether a virtual field created for a host or joined structure is retained after a JOIN command is run. This parameter applies when a DEFINE command precedes the JOIN command. |
| **Syntax:** | `SET KEEPDEFINES = {ON\|OFF}`<br>where:<br>`ON`<br>Retains the virtual field after a JOIN command is run.<br>`OFF`<br>Clears the virtual field after a JOIN command is run. This value is the default. |

| | |
|---|---|
| **Parameter:** | LANG[UAGE] |
| **Description:** | Specifies the National Language Support (NLS) environment. |
| **Syntax:** | `SET LANG[UAGE] = {value\|AMENGLISH}`<br>where:<br>*value*<br>Is a language from the following list. The ID, name, or abbreviation can be used to specify the language. The default value is AMENGLISH. |

| ID | Name | Abbreviation | ID | Name | Abbreviation |
|---|---|---|---|---|---|
| 1 | ENGLISH* | AME | 36 | HUNGARIAN | HUN |
| 1 | AMENGLISH* | AME | 354 | ICELANDIC | ICL |
| 20 | ARABIC | ARB | 39 | ITALIAN | ITA |
| 359 | BULGARIAN | BLG | 81 | JAPANESE* | JPN |
| 416 | CANFRENCH | CFR | 10081 | JAPANESE?E* | JPN |
| 34 | CATALAN | CAT | 82 | KOREAN | KOR |
| 85 | S?CHINESE | PRC | 47 | NORWEGIAN | NOR |
| 86 | T?CHINESE | ROC | 48 | POLISH | POL |
| 45 | DANISH | DAN | 35 | PORTUGUESE | POR |

| ID | Name | Abbreviation | ID | Name | Abbreviation |
|---|---|---|---|---|---|
| 31 | DUTCH* | DUT | 7 | RUSSIAN | RUS |
| 358 | FINNISH | FIN | 38 | SLOVENIAN | SLO |
| 32 | FLEMISH | FLM | 34 | SPANISH* | SPA |
| 33 | FRENCH* | FRE | 46 | SWEDISH | SWE |
| 49 | GERMAN* | GER | 66 | THAI | THA |
| 30 | GREEK | GRE | 90 | TURKISH | TUR |
| 972 | HEBREW | HEB | 44 | UKENGLISH | UKE |
| 91 | HINDI | IND | | | |

*When one of these languages is specified, error messages are translated to that language.

**Note:** To specify JAPANESE?E, you can use the ID or the full name, but not the abbreviation JPN.

| | |
|---|---|
| **Parameter:** | LEFTMARGIN |
| **Description:** | Sets the left boundary for report contents on a page. This command applies to PostScript and PDF report formats. |
| **Syntax:** | `SET LEFTMARGIN = {value\|.250}`<br>where:<br>*value*<br>Is the left boundary of report contents on a page. The default is .25 inches. |

| | |
|---|---|
| **Parameter:** | LINES |
| **Description:** | Sets the maximum number of lines that appear on a logical page, from the heading at the top to the footing on the bottom. The value of LINES can range between 1 and 999999; specify 999999 for continuous forms. WebFOCUS will generate HEADING and FOOTING text after each group of lines as defined with the SET parameter. |
| **Syntax:** | `SET LINES = {n\|57}`<br>where:<br>*n*<br>Is the maximum number of lines of output that appear on a logical page. The default value is 57. |

| | |
|---|---|
| **Parameter:** | LINKMODE |
| **Description:** | In Developer Studio, specifies whether the WebFOCUS CGI is accessed or if the iWay Server is access directly when a procedure is run. This setting determines whether your connection from the Developer Studio to the WebFOCUS Reporting Server is persistent or non-persistent. |
| **Syntax:** | `SET LINKMODE = {EDA\|HTTP}`<br>where:<br>`EDA`<br>Accesses the iWay Server directly. This creates a persistent connection.<br>`HTTP`<br>Accesses the iWay Server through the WebFOCUS CGI or HTTP Listener. This creates a non-persistent connection. This value is the default. |

| | |
|---|---|
| **Parameter:** | LOOKGRAPH |
| **Description:** | Specifies a graph style. This can be a number of variations of area, bar, histogram, and pie charts, among other graph styles. |
| **Syntax:** | `SET LOOKGRAPH = option`<br>where:<br>*`option`*<br>Specifies a graph style. An extensive number of graph styles is available for WebFOCUS. For a complete list, see the *Creating Reports With WebFOCUS Language* manual. |

| | |
|---|---|
| **Parameter:** | MAXLRECL |
| **Description:** | This parameter does not apply to FOCUS data sources.<br><br>Specifies the maximum length of a record described with the Master File OCCURS attribute. |
| **Syntax:** | `SET MAXLRECL = {n\|12288}`<br><br>where:<br><br>*n*<br>Is the maximum record length for a record with the Master File OCCURS attribute. The default value is 12288. The maximum value allowed depends on your platform. |

| | |
|---|---|
| **Parameter:** | MESSAGE |
| **Description:** | Controls the display of informational messages in the view source of your Web browser. |
| **Syntax:** | `SET {MESSAGE\|MSG} = {ON\|OFF}`<br><br>where:<br><br>ON<br>Displays informational messages. This value is the default.<br><br>OFF<br>Suppresses both informational messages and carets that appear when WebFOCUS executes commands in procedures. WebFOCUS still displays error messages, and the carets that prompt for input. |

| | |
|---|---|
| **Parameter:** | MULTIPATH |
| **Description:** | Controls whether a parent segment is included in report output when selection tests are done on independent paths. That is, it determines whether WHERE or IF tests on separate paths are treated as if an AND or OR operator connects them. |
| **Syntax:** | `SET MULTIPATH = {SIMPLE\|COMPOUND}`<br>where:<br>`SIMPLE`<br>Treats selection tests as if connected by an OR operator. The parent segment displays if the condition is true for one of the tests. The following message also displays:<br>`Warning. Testing in independent sets of data`<br>`COMPOUND`<br>Treats selection tests as if connected by an AND operator. The parent segment displays only if the conditions are true for all tests. This value is the default. |

| | |
|---|---|
| **Parameter:** | NODATA |
| **Description:** | Determines the character string that indicates missing data in a report. |
| **Syntax:** | `SET {NODATA\|NA} = {`*`string`*`\|.}`<br>where:<br>*`string`*<br>Is the character string that indicates missing data in reports. The default is a period. |

| | |
|---|---|
| **Parameter:** | OFFLINE-FMT |
| **Description:** | Determines the format of printed report output generated from a request. |
| **Syntax:** | `SET OFFLINE-FMT = option`<br>where:<br>*option*<br>Is one of the following:<br>STYLED formats the report according to the active StyleSheet. This value is the default.<br>STANDARD produces the report as un-styled character-based output. |

| | |
|---|---|
| **Parameter:** | ONLINE-FMT |
| **Description:** | Determines the format of report output. |
| **Syntax:** | `SET ONLINE-FMT = option`<br>where:<br>*option*<br>Is one of the following:<br>HTML produces the report as an HTML table. In WebFOCUS, this value is the default.<br>PDF produces output according to the Adobe Acrobat portable document format.<br>POSTSCRIPT and PS display the report, formatted according to the current StyleSheet, as a Postscript file.<br>EXL2K produces output according to Excel 2000 styling. |

| | |
|---|---|
| **Parameter:** | PAGE[-NUM] |
| **Description:** | Controls the numbering of report pages. |
| **Syntax:** | `SET PAGE[-NUM] = {ON\|OFF\|NOPAGE}`<br>where:<br>`ON`<br>Displays the page number on the upper left-hand corner of the report. This value is the default.<br>`OFF`<br>Suppresses page numbering.<br>`NOPAGE`<br>Suppresses page breaks, causing the report to be printed as a continuous logical page. When PAGE is set to NOPAGE, the LINES parameter controls where column headings are printed. You can also use NOLEAD in place of NOPAGE. |

| | |
|---|---|
| **Parameter:** | PCOMMA |
| **Description:** | Enables the retrieval of comma-delimited files created by a PC application or the HOLD FORMAT COM command.<br>A PC application cannot normally retrieve comma-delimited files because when a Master File specifies SUFFIX=COM, incoming alphanumeric values are not enclosed in double quotation marks, and each record is terminated with a comma and dollar sign (,$) character combination. This format does not support retrieval of most comma-delimited files produced by a PC application.<br>PCOMMA enables retrieval of comma-eliminated files that contain alphanumeric data enclosed in double quotation marks and each record is completely contained on one line and terminated with a carriage return and line feed. |

| | |
|---|---|
| **Syntax:** | `SET PCOMMA = {ON\|OFF}`<br>where:<br>`ON`<br>Enables the retrieval of comma-delimited data sources created by a PC application. It indicates that alphanumeric data is enclosed in double quotation marks and each record is completely contained on one line and is terminated with a carriage return and line feed.<br>`OFF`<br>Does not enable the retrieval of comma-delimited data sources created by a PC application. It indicates that alphanumeric data is not enclosed in double quotation marks and each record is terminated with a comma and dollar sign. This value is the default. |

| | |
|---|---|
| **Parameter:** | PRINT |
| **Description:** | In Developer Studio, determines whether report output is sent to your screen or to the printer. Changing the setting for PRINT also resets the values for the WIDTH and LINES parameters.<br>In addition to changing this setting with a SET command, you can enter ONLINE or OFFLINE as a separate command in the Command Console. |
| **Syntax:** | `SET PRINT = {ONLINE\|OFFLINE}`<br>where:<br>`ONLINE`<br>Sends report output to your screen. This value is the default.<br>`OFFLINE`<br>Sends report output to the printer or to a file. Output can also be sent to a file labeled with the offline ddname using the FILEDEF command. |

| | |
|---|---|
| **Parameter:** | PSPAGESETUP |
| **Description:** | This SET parameter applies only to PDF and PostScript files.<br>Enables the PAGESIZE and ORIENTATION parameter settings which determine the size and orientation of the paper. If PSPAGESETUP is set to OFF, the settings for PAGESIZE and ORIENTATION are ignored by WebFOCUS. |

| | |
|---|---|
| **Syntax:** | `SET PSPAGESETUP = {OFF\|ON}`<br>where:<br>`OFF`<br>Does not allow you to select the paper size and orientation. This is the default.<br>`ON`<br>Allows you to select the paper size with the PAGESIZE and ORIENTATION parameters. |

| | |
|---|---|
| **Parameter:** | QUALCHAR |
| **Description:** | Specifies the qualifying character to be used in qualified field names. |
| **Syntax:** | `SET QUALCHAR = {character\|.}`<br>where:<br>*`character`*<br>Is a valid qualifying character. They include:<br>`.` period (hex 4B)<br>`:` colon (hex 7A)<br>`!` exclamation point (hex 5A)<br>`%` percent sign (hex 6C)<br>`¦` broken vertical bar (hex 6A)<br>`\` backslash (hex E0)<br>The period is the default character. The use of the other qualifying characters listed above is restricted; they should not be used with 66-character field names.<br>If the qualifying character is a period, you can use any of the other characters listed above as part of a field name. If you change the default qualifying character to a character other than the period, then you cannot use that character in a field name. |

| | |
|---|---|
| **Parameter:** | QUALTITLES |
| **Description:** | Uses qualified column titles in report output when duplicate field names exist in a Master File. A qualified column title distinguishes between identical field names by including the segment name. |

| | |
|---|---|
| **Syntax:** | `SET QUALTITLES = {ON\|OFF}`<br>where:<br>`ON`<br>Uses qualified column titles when duplicate field names exist and FIELDNAME is set to NEW.<br>`OFF`<br>Disables qualified column titles. This value is the default. |

| | |
|---|---|
| **Parameter:** | RECAP-COUNT |
| **Description:** | Includes lines containing a value created with RECAP when counting the number of lines per logical page. The number of lines per logical page is determined by the LINES parameter. |
| **Syntax:** | `SET RECAP-COUNT {ON\|OFF}`<br>where:<br>`ON`<br>Counts lines containing a value created with RECAP.<br>`OFF`<br>Does not count lines containing a value created with RECAP. This value is the default. |

| | |
|---|---|
| **Parameter:** | RECORDLIMIT |
| **Description:** | Limits the number of records retrieved. |
| **Syntax:** | `SET RECORDLIMIT = {n\|RECORDLIMIT}`<br>where:<br>*n*<br>Is the maximum number of records to be retrieved.<br>`RECORDLIMIT`<br>Respects explicit RECORDLIMIT values only. This is the default. |

| | |
|---|---|
| **Parameter:** | RIGHTMARGIN |

| | |
|---|---|
| **Description:** | This parameter applies to PostScript and PDF report formats.<br>Sets the right boundary for report contents on a page. |
| **Syntax:** | `SET RIGHTMARGIN = {value\|.25}`<br>where:<br>*value*<br>Is the right boundary of report contents on a page. The default value is .25 inches. |

| | |
|---|---|
| **Parameter:** | RPAGESET |
| **Description:** | Controls how the number of lines per logical page are determined when output contains text created with SUBFOOT and a field value created with RECAP. |
| **Syntax:** | `SET RPAGESET = {NEW\|OLD}`<br>where:<br>`NEW`<br>Sets the number of lines per logical page equal to the LINES value plus two plus the number of the highest BY field with a SUBFOOT.<br>`OLD`<br>Sets the number of lines per logical page equal to the value of the LINES parameter. See the LINES parameter for details. |

| | |
|---|---|
| **Parameter:** | SAVEDMASTERS |
| **Description:** | Saves a Master File in memory after it is used in a request. Saving a Master File prevents re-parsing the Master File in every instance that it is referenced in a single procedure. Up to 99 Master Files can be saved to memory.<br>This parameter cannot be set in the ON TABLE SET command. |
| **Syntax:** | `SET SAVEDMASTERS = n`<br>where:<br>*n*<br>Specifies the number of Master Files that can be saved in memory. When set to 0, Master Files are not saved in memory. The value 0 is the default. |

| | |
|---|---|
| **Parameter:** | SAVEMATRIX |
| **Description:** | Saves the matrix from your request to protect it from being overwritten when using Dialogue Manager commands. |
| **Syntax:** | `SET SAVEMATRIX = {ON\|OFF}`<br>where:<br>`ON`<br>Saves the internal matrix from the last report request, preventing it from being overwritten. In Developer Studio, this value is the default.<br>`OFF`<br>Overwrites the internal matrix for each request. In WebFOCUS, this value is the default. |

| | |
|---|---|
| **Parameter:** | SHIFT |
| **Description:** | In Developer Studio, controls the use of “shift" strings. |
| **Syntax:** | `SET SHIFT = {ON\|OFF}`<br>where:<br>`ON`<br>Specifies a shift string for Hebrew or DBCS (double byte character support).<br>`OFF`<br>Does not specify a shift string for Hebrew or DBCS. This value is the default. |

| | |
|---|---|
| **Parameter:** | SPACES |
| **Description:** | Sets the number of spaces between columns in a report.<br>This parameter does not work with HTML, PDF, or un-styled reports. |
| **Syntax:** | `SET SPACES = {AUTO\|n}`<br>where:<br>`AUTO`<br>Automatically places either one or two spaces between columns. This value is the default.<br>*n*<br>Is the number of spaces to place between columns of a report. Valid values are integers between 1 and 8. |

| | |
|---|---|
| **Parameter:** | SQLTOPTTF |
| **Description:** | Enables the SQL Translator to generate TABLEF commands instead of TABLE commands. |
| **Syntax:** | `SET SQLTOPTTF = {ON\|OFF}`<br>where:<br>`ON`<br>Generates TABLEF commands when possible. For example, a TABLEREF command is generated if there is no JOIN or GROUP BY command.<br>`OFF`<br>Always generates TABLE commands. This value is the default. |

| | |
|---|---|
| **Parameter:** | SQUEEZE |
| **Description:** | This parameter applies to the StyleSheet feature.<br><br>Determines the column width in report output. The column width is based on the size of the data value or column title, or on the field format defined in the Master File. |
| **Syntax:** | `SET SQUEEZE = {ON\|OFF}`<br><br>where:<br><br>`ON`<br>Assigns column widths based on the widest data value or widest column title, whichever is longer. This value is the default for reports displayed in HTML format.<br><br>`OFF`<br>Assigns column widths based on the field format specified in the Master File. This value pads the column width to the length of the column title or field format descriptions, whichever is greater. This value is the default for reports displayed in PDF format. |

| | |
|---|---|
| **Parameter:** | %STRICTMATH |
| **Description:** | In WebFOCUS, determines the behavior of a calculation when dividing by zero. |
| **Syntax:** | `SET %STRICTMATH = {'OLD'\|'NEW'}`<br><br>where:<br><br>`OLD`<br>Returns zero when a value is divided by zero. This value must be enclosed in single quotation marks.<br><br>`NEW`<br>Returns the value when the value is divided by zero. This value is the default, and must be enclosed in single quotation marks. |

| | |
|---|---|
| **Parameter:** | STYLEMODE |
| **Description:** | Determines the type of HTML generated for report output.<br><br>**Note:** The SQUEEZE parameter must be set to OFF to align data columns from one page to the next. |

| | |
|---|---|
| **Syntax:** | `SET STYLEMODE = {FULL\|FIXED\|PAGED}`<br>where:<br>`FULL`<br>Displays normal HTML output. This value is the default.<br>`FIXED`<br>Generates <PRE tag in HTML to indicate preformatted text which is not to be treated as HTML.<br>`PAGED`<br>In WebFOCUS, displays report output in multiple HTML tables where each table is a separate report page. These smaller HTML files are retrieved from the Web server quicker than a single large file. |

| | |
|---|---|
| **Parameter:** | STYLE[SHEET] |
| **Description:** | Controls the format of report output by accepting or rejecting StyleSheet parameters. These parameters specify formatting options such as page size, orientation, and margins. |
| **Syntax:** | `SET STYLE[SHEET] = {stylesheet\|ON\|OFF}`<br>where:<br>*stylesheet*<br>Is the name of the StyleSheet file. For UNIX and NT, this is the name of the StyleSheet file without the file extension .sty. For OS/390, this is the member name in the PDS allocated to ddname FOCSTYLE by a DYNAM command.<br>For a PDF or PostScript report, WebFOCUS uses the page layout settings for UNITS, TOPMARGIN, BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN, PAGESIZE, ORIENTATION, and SQUEEZE; the settings for LINES and WIDTH are ignored.<br>ON<br>Creates an HTML table using the default proportional font defined in the end user's browser. This value is the default.<br>For a PDF or PostScript report, WebFOCUS uses the page layout settings for UNITS, TOPMARGIN, BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN, PAGESIZE, ORIENTATION, and SQUEEZE; the settings for LINES and WIDTH are ignored.<br>OFF<br>Creates a preformatted report using the default fixed font defined in the end user's browser.<br>For a PDF or Postscript report, WebFOCUS uses the settings for LINES and WIDTH; the settings for UNITS, TOPMARGIN, BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN, PAGESIZE, ORIENTATION, and SQUEEZE are ignored.<br>This is used when the STYLE parameter is set in an ON TABLE command. It indicates that StyleSheet commands follow and are embedded in the procedure. |

| | |
|---|---|
| **Parameter:** | SUMPREFIX |
| **Description:** | Allows users to choose the answer set display order when using an external sort to perform aggregation of alphanumeric or smart date formats. |
| **Syntax:** | `SET SUMPREFIX = {LST\|FST}`<br>where:<br>`LST`<br>Displays the last value when alphanumeric or smart date data types are aggregated. This value is the default.<br>`FST`<br>Displays the first value when alphanumeric or smart date data types are aggregated. |

| | |
|---|---|
| **Parameter:** | TARGETFRAME |
| **Description:** | Includes the HTML code <BASE TARGET="*framename*"> in the heading of the HTML file that WebFOCUS displays in your browser. All drill-down hyperlinks from the base report or graph are directed to the specified frame unless overridden by the TARGET attribute in the StyleSheet. |
| **Syntax:** | `SET TARGETFRAME = framename`<br>where:<br>*`framename`*<br>Is the frame on the Web page in which the output from the drill-down hyperlink (either a FOCEXEC or URL) is displayed. Possible values include standard HTML frame names such as _blank, _self, _parent, _top or a user-defined name. |

| | |
|---|---|
| **Parameter:** | TEMP |
| **Description:** | In WebFOCUS, assigns temporary files to a specific directory. |
| **Syntax:** | `SET TEMP = directory`<br>where:<br>*`directory`*<br>Is the directory to which all temporary files are assigned. |

| **Parameter:** | TEMPDIR |
|---|---|
| **Description:** | In Developer Studio, assigns temporary files to a specific directory.<br>This parameter is only applicable when the server is not in APP mode. |
| **Syntax:** | `SET TEMPDIR = directory`<br>where:<br>*directory*<br>Is the directory to which all temporary files are assigned. |

| **Parameter:** | TEMPERASE |
|---|---|
| **Description:** | Determines if the temporary files created during a WebFOCUS connection are retained after the connection is closed. Applies to HOLD files and other files that may be created during the session. |
| **Syntax:** | `SET TEMPERASE = {ON\|OFF}`<br>where:<br>OFF<br>Erases any temporary files after the WebFOCUS connection is closed.<br>ON<br>Keeps any temporary files created during a WebFOCUS connection. |

| **Parameter:** | TESTDATE |
|---|---|
| **Description:** | Temporarily alters the system date in order to test a dynamic window. That is, it allows you to simulate clock settings beyond the year 1999 to determine the behavior of your program. |
| **Syntax:** | `SET TESTDATE = {yyyymmdd\|TODAY}`<br>where:<br>*yyyymmdd*<br>Is an 8-digit date in the format YYYYMMDD.<br>TODAY<br>Is the current date. This value is the default. |

| | |
|---|---|
| **Parameter:** | TITLE |
| **Description:** | Uses predefined column titles in the Master File as column titles in report output. |
| **Syntax:** | `SET TITLE[S] = {ON\|OFF}`<br>where:<br>`ON`<br>Uses predefined column titles in the Master File as column titles in report output. This value is the default.<br>`OFF`<br>Uses the field names in the Master File as column titles in report output. |

| | |
|---|---|
| **Parameter:** | TOPMARGIN |
| **Description:** | This parameter applies to PostScript and PDF report formats.<br>Sets the top boundary on a page for report output. |
| **Syntax:** | `SET TOPMARGIN = {value\|.25}`<br>where:<br>*`value`*<br>Is the top boundary on a page for report output. The default value is .25 inches. |

| | |
|---|---|
| **Parameter:** | UNITS |
| **Description:** | This parameter applies to PostScript and PDF report formats.<br>Specifies the unit of measure for page margins, column positions, and column widths. |
| **Syntax:** | `SET UNITS = {INCHES\|CM\|PTS}`<br>where:<br>`INCHES`<br>Uses inches as the unit of measure. This value is the default.<br>`CM`<br>Uses centimeters as the unit of measure.<br>`PTS`<br>Uses points as the unit of measurement. (One inch = 72 points, one cm = 28.35 points) |

| | |
|---|---|
| **Parameter:** | USER |
| **Description:** | In WebFOCUS, enables user access to a data source or stored procedure protected by Information Builders security. |
| **Syntax:** | `SET USER = user`<br>where:<br>*user*<br>Is the user name that enables access to a data source or stored procedure protected by Information Builders security. |

| | |
|---|---|
| **Parameter:** | VAUTO |
| **Description:** | Performs automatic scaling of the vertical axis for the given values. If VAUTO is OFF, the vertical axis must be scaled using the VMAX and VMIN parameters. (See VMAX and VMIN.) |
| **Syntax:** | `SET VAUTO = {ON\|OFF}`<br>where:<br>ON<br>Scales the vertical axis automatically. This value is the default.<br>OFF<br>Does not scale the horizontal axis automatically. The end user must supply values for VMAX and VMIN. |

| | |
|---|---|
| **Parameter:** | VAXIS |
| **Description:** | This parameter applies to graphs generated offline.<br>Specifies the length of the vertical axis, in lines. VAXIS is ignored for online displays since WebFOCUS automatically adjusts the length of the graph to the height of the terminal. |
| **Syntax:** | `SET VAXIS = {n\|405}`<br>where:<br>*n*<br>Is the page length, in lines. Valid values are integers between 20 and 405. The default value is 405. |

| | |
|---|---|
| **Parameter:** | VCLASS |
| **Description:** | Specifies the vertical interval mark when AUTOTICK is OFF. See also VTICK. |
| **Syntax:** | `SET VCLASS = {nnn\|0}`<br>where:<br>*nnn*<br>Is the vertical interval mark.<br>0<br>Specifies zero as the vertical interval mark. This value is the default. |

| | |
|---|---|
| **Parameter:** | VGRID |
| **Description:** | Draws a grid at the horizontal and vertical class marks of the graph. See also GRID. |
| **Syntax:** | `SET VGRID = {ON\|OFF}`<br>where:<br>ON<br>Draws a grid at the horizontal and vertical class marks of the graph.<br>OFF<br>Does not draw a grid. This value is the default. |

| | |
|---|---|
| **Parameter:** | VMAX |
| **Description:** | Sets the maximum value on the vertical axis when automatic scaling is not used (VAUTO=OFF). |
| **Syntax:** | `SET VMAX = {nnn\|.00}`<br>where:<br>*nnn*<br>Is the maximum value on the vertical axis when VAUTO is OFF. The default value is .00. |

| | |
|---|---|
| **Parameter:** | VMIN |
| **Description:** | Sets the minimum value on the vertical axis when automatic scaling is not used (VAUTO=OFF). |
| **Syntax:** | `SET VMIN = {nnn\|.00}`<br>where:<br>*nnn*<br>Is the minimum value on the vertical axis when VAUTO is OFF. The default value is .00. |

| | |
|---|---|
| **Parameter:** | VTICK |
| **Description:** | Sets the vertical axis interval mark when AUTOTICK is OFF. See also VCLASS. |
| **Syntax:** | `SET VTICK = {nnn\|.00}`<br>where:<br>*nnn*<br>Is the vertical axis interval mark. The default value is .00. |

| | |
|---|---|
| **Parameter:** | VZERO |
| **Description:** | Treats missing values on the vertical axis as zeros. When VZERO is off, these values are ignored. |
| **Syntax:** | `SET VZERO = {ON\|OFF}`<br>where:<br>ON<br>Treats missing values as zeros.<br>OFF<br>Ignores missing values. This value is the default. |

| | |
|---|---|
| **Parameter:** | WEBVIEWER |
| **Description:** | Enables on-demand paging, and invokes the WebFOCUS Viewer. The first page of report output is displayed in the WebFOCUS Viewer while the remaining pages remain on the Web server's disk. |
| **Syntax:** | `SET WEBVIEWER = {ON\|OFF}`<br>where:<br>`ON`<br>Enables on-demand paging. The first page of report output is displayed in the WebFOCUS Viewer, while the remaining pages are held on the Web server's disk.<br>`OFF`<br>Sends all report output to your browser. The report output displays in a standard browser window. |

| | |
|---|---|
| **Parameter:** | WEEKFIRST |
| **Description:** | Specifies a day of the week as the start of the week. This is used in week computations by the HDIFF, HNAME, HPART, and HSETPT functions.<br>The WEEKFIRST parameter does not change the day of the month that corresponds to each day of the week, but only specifies which day is considered the start of the week. |
| **Syntax:** | `SET WEEKFIRST = {n\|7}`<br>where:<br>*n*<br>Is a number from one to seven, where one represents Sunday and seven represents Saturday. The default value is seven, which is consistent with the Microsoft SQL Server convention. |

| | |
|---|---|
| **Parameter:** | XRETRIEVAL |
| **Description:** | Controls the retrieval of data when previewing a report. |
| **Syntax:** | `SET XRETRIEVAL = {ON\|OFF}`<br>where:<br>`ON`<br>Retrieves data when previewing a report. This value is the default.<br>`OFF`<br>Does not retrieve data when previewing a report. |

| | |
|---|---|
| **Parameter:** | YRTHRESH |
| **Description:** | Defines the start of a 100-year window globally or on a field-level. Used with DEFCENT, interprets the current century according to the given values. Two-digit years greater than or equal to YRTHRESH assume the value of the default century. Two-digit years less than YRTHRESH assume the value of one more than the default century. (See DEFCENT.)<br>**Note:** This same result can be achieved by including the FDEFCENT and FYRTHRESH attributes in the Master File. |
| **Syntax:** | `SET YRTHRESH = {[-]yy\|0}`<br>where:<br>`yy`<br>Is the year threshold for the window. If you do not supply a value, *yy* defaults to zero (0).<br>If *yy* is a positive number, that number is the start of the 100-year window. Any two-digit years greater than or equal to the threshold assume the value of the default century. Two-digit years less than the threshold assume the value of one more than the default century.<br>If *yy* is a negative number (-*yy*), the start date of the window is derived by subtracting that number from the current year, and the default century is automatically calculated. The start date is automatically incremented by one at the beginning of each successive year. |

CHAPTER 16

# Defining and Allocating WebFOCUS Files

**Topics:**

- WebFOCUS Files Under Windows NT/2000/XP
- Application Files Under Windows NT/2000/XP
- Extract Files Under Windows NT/2000/XP
- Work Files Under Windows NT/2000/XP
- WebFOCUS Files Under MVS
- Application Files Under MVS
- Extract Files Under MVS
- Work Files Under MVS
- Reviewing Attributes of Allocated Files Under MVS
- WebFOCUS Files Under UNIX
- Application Files Under UNIX
- Extract Files Under UNIX
- Work Files Under UNIX

You will come across a variety of files when you use WebFOCUS. These files can be application files, which you create; extract files, which WebFOCUS creates as requested by your application; or work files, which WebFOCUS creates as required by your application.

This topic describes WebFOCUS file allocation requirements and conventions for specific operating systems. It describes how to assign a logical name, or ddname, to a file or device for operating systems that support this feature.

# WebFOCUS Files Under Windows NT/2000/XP

WebFOCUS files fall into three categories: application files, extract files, and work files. This topic contains information about files and ways to define them in the Windows NT/2000/XP environment.

Some files are automatically allocated by WebFOCUS, but there are times when you must explicitly define a file and its location. You can do this by using either the FILEDEF tool or the FILEDEF command. For details on the FILEDEF command see *Assigning a Logical Name Under Windows NT/2000/XP* on page 16-5.

## *Syntax* How to Reference a File Under Windows NT/2000/XP

*`filename.filetype`*

where:

*`filename`*

Is the name of the file.

*`filetype`*

Is the type of file. This is a three-letter extension for an installation of Windows NT/2000/XP based on the File Allocation Table (FAT) file system.

## *Reference* WebFOCUS Files Under Windows NT/2000/XP

The following tables describe the WebFOCUS files that you will use under Windows NT/2000/XP. WebFOCUS uses file extensions to distinguish different file types.

**Application Files**

| Extension | Description |
|---|---|
| `.MAS` | Master File. |
| `.ACX` | Access File. |
| `.FEX` | Procedure (FOCEXEC). |
| `.FOC` | FOCUS data source and external index. |
| `Varied` | Other supported data sources. |
| `.DAT` | Sequential data source |
| `.STY` | WebFOCUS StyleSheet file. |
| `.EXE or .DLL` | Library of functions. |

| Extension | Description |
|---|---|
| `.PRF` | Profile. |
| `.HTM`<br>`.HTML`<br>`.JPEG`<br>`.GIF`<br>`.CSS`<br>`.JS`<br>`.CLASS`<br>`.JAR` | Files displayed in a Web browser. |

**Extract Files**

| File | Description |
|---|---|
| `HOLD` | Contains data saved using the HOLD command. |
| `SAVB` | Contains data saved using the SAVB command. |
| `SAVE` | Contains data saved using the SAVE command. |
| `HOLDMAST` | Temporary Master File for HOLD files. |

| Extension | Description |
|---|---|
| `.FTM` | Contains data saved using the HOLD, SAVB, or SAVE command. |

**Note:** Dialogue Manager output files must be allocated using the FILEDEF command in system or user profiles.

**Work Files**

| File | Description |
|---|---|
| FOCSTACK | Contains resolved Dialogue Manager procedures. |
| FOCSORT | Used during sorting. |
| FOCPOST | Sequential output file saved using the POST command. The PICKUP command reads it back in. |
| FOCSML | Used by Financial Modeling Language (FML). |
| OFFLINE | Used when the SET PRINT parameter is OFFLINE. |
| SYSIN | Directs input. |
| SYSPRINT | Directs output to the screen. |

## Dynamically Defining a File Under Windows NT/2000/XP

You do not have to explicitly define most of your application files prior to referring to them. WebFOCUS dynamically allocates certain application files. Additionally, WebFOCUS dynamically defines all extract files and temporary work files to the operating system during a session.

WebFOCUS defines the following extract and work files:

- HOLD, SAVB, and SAVE files.
- FOCSORT files.
- SYSIN to input from keyboard. It is assigned to TERM. This is equivalent to the command FILEDEF SYSIN TERM.
- SYSPRINT to output to be displayed on the screen. It is assigned to TERM. This is equivalent to the command FILEDEF SYSPRINT TERM.
- OFFLINE to output sent to the printer. It is assigned to PRINTER. This is equivalent to the command FILEDEF OFFLINE PRINTER.
- FOCSTACK to the file FOCSTACK.FTM. This is equivalent to the command FILEDEF FOCSTACK DISK FOCSTACK.FTM. For more information on FOCSTACK, see Chapter 10, *Managing Flow of Control in an Application*.

## Assigning a Logical Name Under Windows NT/2000/XP

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 FILEDEF command. A FILEDEF command can be issued in a procedure and lasts for a single request.

It is recommended that instead of including a FILEDEF 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 allows you to make changes to your FILEDEF commands globally instead of changing the FILEDEF 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 Filedef Tool

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), then click the *Filedef* button on the Component Connector toolbar.

   The Filedef tool opens.

3. Click *New*.

   The cursor appears in the Name field.

4. Enter the logical name you wish to assign to the file in the Name field.

   The name can be from 1 to 8 characters. If you are identifying a data source, the name must be the same as the name of the Master File.

5. Select the device for the logical name in the Device section. The options are:

   *Disk* associates the logical name with a file.

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

   *Terminal* indicates that the keyboard and monitor are the input source and output destination for the file. If you select terminal, a Lower Case button displays which, when selected, retains the case.

   *Clear* clears the FILEDEF assigned to a file name.

6. Enter the file name in the Filename field. The name can include the drive letter and directory.

   or

   Click *Browse*. A dialog box opens. Choose the file from the dialog box.

7. Optionally, select *Append* to append the records to the end of the file. If you do not select this option, the file is overwritten.

8. Optionally, if your file has a fixed record length, specify the length:

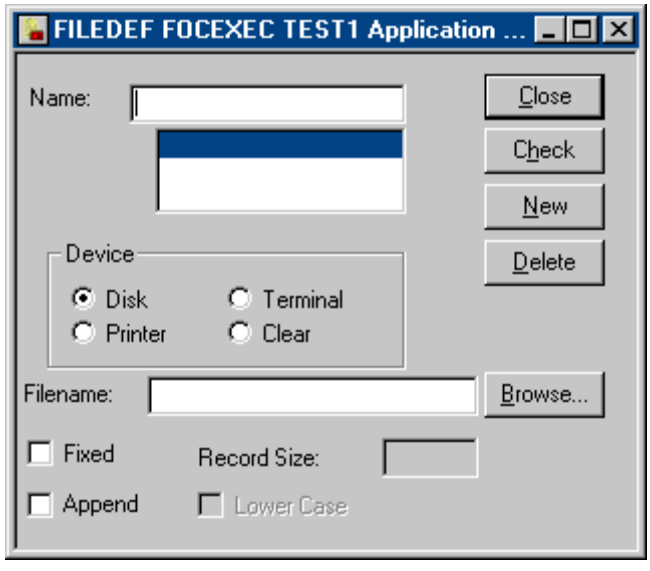
   **a.** Select the *Fixed* check box.

   **b.** Enter the record length in the Record Size field.

9. Click *Close*.

   You are prompted to save your changes.

## *Reference* FILEDEF Dialog Box




The FILEDEF dialog box has the following entry fields and buttons:

**Name**

Shows the logical (defined) name of a file or device (1-8 characters).

**Device**

Identifies the type of device to associate with the logical name. The options are:

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

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

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

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

**Filename**

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.

**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.

**Close**

Closes the Filedef tool.

**Check**

Displays the Filedef phrase in code, and specifies any errors.

**New**

Moves the cursor into the Name box for your entry.

**Delete**

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

*Syntax*

## How to Assign a Logical Name With Code Under Windows NT/2000/XP

```
FILEDEF ddname DISK [filename] [(APPEND] [LRECL n] [RECFM F]
```

or

```
FILEDEF SYSIN TERM [LOWER]
```

or

```
FILEDEF ddname PRINTER
```

where:

*ddname*

Is the logical name for the file, input, or output. The *ddname* can be from 1 to 8 characters. When used to associate a data source with a Master File, the *ddname* must match the name of the Master File.

DISK

Associates the specified *ddname* with a file.

*filename*

Is the complete file name. It may include a drive and directory specification.

APPEND

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

`LRECL n`

Specifies the record length; *n* is an integer.

You must specify an LRECL value if the file is a SAVB file or fixed-format transaction file used in data maintenance (FIXFORM files) that contains binary values.

`RECFM F`

Specifies fixed length records.

You must specify an RECFM value if the file is a SAVB file or fixed-format transaction file used in data maintenance (FIXFORM files) that contains binary values.

`TERM`

Specifies the keyboard and monitor as the input source and output destination.

`LOWER`

Sends keyboard input to WebFOCUS as entered.

`PRINTER`

Specifies the printer as the output destination.

### *Example* Assigning a Logical Name to a File Located on a Server

Assign a logical name to a file located on a server using the Universal Naming Convention:

```
FILEDEF DATFILE DISK \\SERVER2\DISK1\MAYSPLES.DAT
```

### *Example* Redirecting Output

To send output to the LPT1 port (provided that the machine is configured properly):

```
FILEDEF OFFLINE DISK LPT1
```

### *Example* Setting a Search Path

To search all directories on the search path for the NEW_EMPS.DAT file:

```
FILEDEF SYSIN DISK *:NEW_EMPS.DAT
```

### *Example* Appending a Report Extract to Other Content

To append a report extract from the LIBRARY data source to the current content of the file LIB03.FTM:

```
FILEDEF SAVE DISK C:\LIBRARY\LIB03.FTM (APPEND
```

### *Example* Reading a Standard Text Editor Files With LRECL

You can specify an LRECL equal to or greater than the implied LRECL for the request. For example, if the length of the longest line in the text file is 7 characters, issue this command:

```
FILEDEF BIGLINE DISK BIGLINE.FTM (LRECL 7
```

## Displaying Current ddnames Assigned With FILEDEF

The ? FILEDEF command displays the ddnames assigned for various files, input and output.

### *Syntax* How to Display Current ddnames

```
? FILEDEF
```

### *Example* Displaying Current ddnames

Issuing the command

```
? FILEDEF
```

produces information similar to the following:

```
Lname   Device  Lrecl Recfm Append Expl Filename
=================================================
HOLD2   DISK     0   V     N    Y C:\VM\SMALL\HOLD2.FTM
```

## Clearing FILEDEF Assignments

You can clear FILEDEF assignments.

### *Procedure* How to Clear a FILEDEF Assignment With the Filedef Tool

1. Select the logical name in the list below the Name box.
2. Select the *Clear* radio button under Device.

### *Syntax* How to Clear a Logical Name With Code

```
FILEDEF ddname CLEAR
```

where:

*ddname*

Is the logical name. It may contain 1 to 8 alphanumeric characters.

CLEAR

Clears the specified ddname.

# Application Files Under Windows NT/2000/XP

You can use WebFOCUS to reference and search for your application files.

When you do not change the default file type, you can prepare a report without issuing a FILEDEF command or other communication. When you change the default file type of a FOCUS data source, or the data source does not reside in a standard search path, you must issue the USE command. The USE command is discussed in Chapter 14, *Accessing a FOCUS Data Source*.

## Master Files Under Windows NT/2000/XP

A Master File contains metadata about a data source, and is identified with a file name and extension, for example, *filename*.mas. It consists of attributes that describe a data source. A Master File and the data source that it describes usually have the same name. A Master File must consist of 80-byte, fixed length records.

The description of a data source and all files it cross-references must be available whenever you refer to the data source.

For more information about how to generate Master Files, see the *Describing Data With WebFOCUS Language* manual.

## Locating a Master File Under Windows NT/2000/XP

WebFOCUS searches for a Master File on the path of the WebFOCUS Reporting Sever, using either EDAPATH or APP PATH logic. For details on search path logic, see *Storing and Searching for Application Files* in Chapter 2, *WebFOCUS Application Logic*.

### *Reference* Naming a Master File Under Windows NT/2000/XP

The following rules apply:

- The file name can be up to 64 characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are therefore not recommended or supported.
- The file extension must be mas for a Master File, since WebFOCUS searches for a Master File with that extension by default.

## Access Files Under Windows NT/2000/XP

For some data sources, an Access File supplements a Master File. An Access File includes additional information that completes the description of the data source. For example, it includes the full data source name and location. You need one Master File–and, for some data sources, one Access File–to describe a data source.

WebFOCUS searches for an Access File on the path of the WebFOCUS Reporting Server using either EDAPATH or APP PATH logic.

### *Reference* Naming an Access File Under Windows NT/2000/XP

The following rules apply:

- The file name can be up to eight characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are therefore not recommended or supported.
- The file extension must be acx since WebFOCUS searches for an Access File with that extension by default.
- The file name should have the same name as the corresponding Master File.

## Procedures Under Windows NT/2000/XP

A procedure is an executable text file that contains your report request. It can have up to 3024 characters per line. A procedure is identified with a file name and extension, for example, *filename*.fex.

WebFOCUS searches for a procedure on the path of the WebFOCUS Reporting Server, using either EDAPATH or APP PATH logic.

### *Reference* Naming a Procedure Under Windows NT/2000/XP

The following rules apply:

- The file name can be up to 64 characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are therefore not recommended or supported.
- The file extension must be fex since WebFOCUS searches for a procedure with that extension by default.

## FOCUS Data Sources Under Windows NT/2000/XP

FOCUS data sources contain data written in FOCUS format. All FOCUS data sources have a record length of 4096 and a fixed length record format. The maximum size of a FOCUS data source is 1G or 256K pages.

A FOCUS data source is identified with a file name and extension, for example, *filename*.foc. The name of a FOCUS data source usually matches the name of the corresponding Master File. For example, if the Master File is ledger.mas, then the FOCUS data source is ledger.foc. You can override this default with the USE command, described in Chapter 14, *Accessing a FOCUS Data Source*.

WebFOCUS searches for a FOCUS data source on the platform with the WebFOCUS Reporting Server or subserver.

### *Reference* Naming a FOCUS Data Source Under Windows NT/2000/XP

The following rules apply:

- The file name can be up to 64 characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are therefore not recommended or supported.
- The file extension must be foc since WebFOCUS searches for a FOCUS data source with that extension by default.

## External Indexes for FOCUS Data Sources Under Windows NT/2000/XP

An external index is a FOCUS data source that contains index, field, and segment information for one or more specified FOCUS data sources. The external index is independent of its associated FOCUS data source and is used to improve retrieval performance. In Windows NT/2000/XP, the external index is automatically allocated as a permanent file when it is created using REBUILD.

For more information see the iWay documentation for Windows NT/2000/XP.

## Other Supported Data Sources Under Windows NT/2000/XP

WebFOCUS supports many data source types. See the *iWay Server Administration* manual for details.

WebFOCUS searches for a data source on the platform with the WebFOCUS Reporting Server or subserver.

### *Reference* Naming a Supported Data Source Under Windows NT/2000/XP

WebFOCUS supports standard data source naming conventions.

## Sequential Data Sources Under Windows NT/2000/XP

WebFOCUS recognizes sequential data sources formatted in the following ways:

- Fixed format, in which each field occupies a predefined position in the record.
- Free-format, also known as comma-delimited, in which fields can occupy any position in a record.

For details see the *Describing Data With WebFOCUS Language* manual.

## WebFOCUS StyleSheet Files Under Windows NT/2000/XP

WebFOCUS StyleSheets enable you to style and produce reports that highlight key information. With StyleSheets, you specify various stylistic characteristics of a report and style report components individually. You can also use StyleSheets to define a hyperlink from any reporting object.

You can create StyleSheets externally or within a report request. For details, see the *Creating Reports With WebFOCUS Language* manual.

WebFOCUS searches for a WebFOCUS StyleSheet on the path of the WebFOCUS Reporting Server, using either EDAPATH or APP PATH logic.

### *Reference* Naming a StyleSheet File Under Windows NT/2000/XP

The following rules apply:

- The file name can be up to 64 characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are therefore not recommended or supported.
- The file extension must be sty since WebFOCUS searches for a StyleSheet with that extension by default.

## Library of Functions Under Windows NT/2000/XP

External function libraries are located outside of WebFOCUS. For more information on function libraries see the *Using Functions* manual.

## Profiles Under Windows NT/2000/XP

WebFOCUS supports the following levels of profiles to provide flexibility in designing and running applications.

- A **global profile** is a startup file that is automatically created during the installation and configuration of the server. It contains the default environment settings that are required for the proper operation of the server.
- A **user profile** is available based on a user ID. A user profile is a file used by the server to specify settings that apply to a server environment, but only for a specific user ID.

For details on profiles see the *iWay Server Administration* manual.

## Web Pages Under WebFOCUS Under Windows NT/2000/XP

Web pages and related files provide the presentation logic for an application. They include files displayed for the user in a browser, such as HTML files, graphical images, Java class and archive files, JavaScript files, and cascading StyleSheets.

The Web Server searches for a Web page or related files in the Web server home directory or Web server alias. An HTML file called by -HTMLFORM must reside in a path defined by the WebFOCUS Reporting Server.

### *Reference* Naming a Web Page Under Windows NT/2000/XP

WebFOCUS supports standard file naming conventions. Web pages must have an extension of .HTM.

# Extract Files Under Windows NT/2000/XP

WebFOCUS creates all extract files in a temporary directory on the WebFOCUS Reporting Server, or in a user-defined location on the WebFOCUS Reporting Server.

*Reference* **Locating Extract Files Under Windows NT/2000/XP**

WebFOCUS creates all extract files in a temporary directory under the directory specified by the EDATEMP environment variable in EDASERVE.CFG.

## HOLD Files Under Windows NT/2000/XP

A HOLD extract file is a sequential data source that contains the results of a report request. It may have a corresponding HOLD Master File. The extension for an extract file is ftm unless you specify the FORMAT option. If a Master File is created, it has the same name as the extract file, with the extension mas.

If you specify the FORMAT FOCUS option with HOLD, WebFOCUS creates an extract file and a Master File, each with the file name FOC$HOLD. These files are then used as input to the procedure that creates the final FOCUS file. The new FOCUS and Master Files are created in the user temporary directory.

*Syntax* **How to Create a HOLD File Under Windows NT/2000/XP**

```
ON TABLE HOLD [AS filename]
```

where:

*filename*

Is a name for the HOLD file. If you do not specify a file name, HOLD is used as the default name. Since each subsequent HOLD command overwrites the previous HOLD file, it is useful practice to code a distinct file name in each request to direct the extracted data to a separate file, preventing it from being overwritten.

For the complete syntax with all options, see the *Creating Reports With WebFOCUS Language* manual.

## SAVB Files Under Windows NT/2000/XP

A SAVB file is an extract file containing the results of a report request in internal format. That is, all numeric fields are stored in binary, and all character fields are padded with spaces to a multiple of four bytes. The file cannot be printed.

When a SAVB file is created, WebFOCUS does not create a Master File.

### *Syntax* How to Create a SAVB File Under Windows NT/2000/XP

```
ON TABLE SAVB [AS filename]
```

where:

*filename*

Is the name of the file. The default is SAVB. The default extension is FTM.

For the complete syntax with all options, see the *Creating Reports With WebFOCUS Language* manual.

## SAVE Files Under Windows NT/2000/XP

A SAVE file is an extract file that saves the data of a report, but does not save headings or subtotals, or create a Master File. It is a simple sequential character data file that can be used by other programs, or merged into another data file using a data maintenance request. The default format is simple character, although you can specify formats compatible with many other software products. WebFOCUS saves all columns in the report in printable, character format with no spaces between columns.

A SAVE file contains the external character format equivalent to SAVB. The command syntax and allocations are the same as SAVB. However, the numbers are printable Extended Binary Coded Decimal Interchange Code (EBCDIC) characters and no padding takes place.

### *Syntax* How to Create a SAVE File Under Windows NT/2000/XP

```
ON TABLE SAVE [AS filename]
```

where:

*filename*

Is the name of the file. The default is SAVE. The default extension is FTM.

## HOLDMAST Files Under Windows NT/2000/XP

A HOLDMAST file is a temporary Master File. It is created with the FILEDEF HOLDMAST command, which specifies its location.

*Syntax*

### How to Specify the Location of a Temporary Master File Under Windows NT/2000/XP

```
FILEDEF HOLDMAST DIR directory_name
```

where:

*directory_name*

Is a valid directory path.

*Example*

### Specifying the Location of a Temporary Master File Under Windows NT/2000/XP

The following example specifies a location for a temporary Master File, and assigns a logical name to a physical data source.

```
1. FILEDEF HOLDMAST DIR C:\TMP
2. FILEDEF MKTSALES DISK C:\TMP\MKTSALES.FTM
    TABLE FILE GGSALES
    SUM DOLLARS BY REGION
    ON TABLE HOLD AS MKTSALES FORMAT ALPHA
    END
```

The numbers listed next to the code correspond with the following notes:

1. Specifies the location of the Master File in the \tmp directory. WebFOCUS uses the same logical name, MKTSALES, specified in the second line, to name the temporary Master File.
2. Specifies MKTSALES as the logical name for the physical data source. The HOLD command creates the temporary Master File mktsales.mas and the data source mktsales.ftm.

## Work Files Under Windows NT/2000/XP

WebFOCUS creates work files as required by your application. WebFOCUS creates all extract files in a temporary directory on the WebFOCUS Reporting Server, or in a user-defined location on the WebFOCUS Reporting Server.

The available work files are:

- **FOCSTACK** files contain resolved Dialogue Manager procedures.
- **FOCSORT**, allocated as FOCSORT.FOC, may be required for sorting with the TABLE, TABLEF, and MATCH commands. The maximum size of a FOCSORT file is 1G of 256K pages.
- **FOCPOST** files hold sequential output, which is saved by the POST command and read by the PICKUP command.
- **FOCSML** files are used by the Financial Modeling Language (FML) facility.
- **OFFLINE** sends output to a file when the  to send output to a file when the PRINT parameter is set to OFFLINE in WebFOCUS Developers Studio. Allocate the file to ddname OFFLINE using a FILEDEF command after the offline has been closed with the OFFLINE CLOSE command.

*Example* **Sending Output to a File With OFFLINE Under Windows NT/2000/XP**

The following request sends report output to the file EMPL.OUT, which is stored on the C: drive in the \tmp directory:

```
OFFLINE
FILEDEF OFFLINE DISK C:\TMP\EMPL.OUT

TABLE FILE MOVIES
PRINT TITLE BY DIRECTOR
WHERE CATEGORY IS 'ACTION'
END
```

# WebFOCUS Files Under MVS

You will come across a variety of files in WebFOCUS. These files fall into three categories: application files, extract files, and work files. This topic contains information about files and ways to define them in the MVS environment.

Some files are automatically allocated by WebFOCUS, but there are times when you must explicitly define a file and its location to WebFOCUS. You can do this by using the DYNAM and ALLOCATE commands. For details on these commands see the *iWay Procedures Reference Manual*.

## Referencing Files Under MVS

Reference files by ddnames rather than by their fully qualified data set names. Data set names are arbitrary but must conform to your installation's naming standards.

*Reference* **WebFOCUS File Types Under MVS**

The following tables describe the WebFOCUS files that you will use under MVS. WebFOCUS uses ddnames to distinguish different file types as follows:

**Application Files**

| File | Description |
|---|---|
| `MASTER` | Master File. |
| `ACCESS` | Access file. |
| `FOCEXEC` | Procedure. |
| `ddname` | FOCUS data source and external index. |
| `ddname` | Other supported data sources. |
| `FOCSTYLE` | WebFOCUS StyleSheet file. |

**Extract Files**

| File | Description |
|---|---|
| `*HOLD` | Contains data saved using the HOLD command. |
| `*SAVB` | Contains data saved using the SAVB command. |
| `*SAVE` | Contains data saved using the SAVE command. |
| `HOLDMAST` | Temporary Master File for HOLD files. |
| `HOLDSTAT Files` | Documentation and/or DBA information for extract files. |

**Note:** There are also extract files that you must allocate if used: LOG, POST, and Dialogue Manager output files.

**Work Files**

| File | Description |
|---|---|
| `FOCSTACK` | Contains resolved Dialogue Manager procedures. |
| `FOCSORT` | Used during sorting. |
| `*FOCPOST` | Holds sequential output. |
| `FOCSML` | Used by Financial Modeling Language (FML). |

*The AS phrase renames asterisked files to allow more than one file of that type in a single session.

### Dynamically Defining Files Under MVS

You do not have to explicitly define most of your files. WebFOCUS automatically allocates many application files on an as-needed basis as long as they follow the data set naming conventions. WebFOCUS allocates five primary cylinders and five secondary cylinders for dynamically allocating files. WebFOCUS dynamically allocates work files, but does not automatically define application files. You need to define these files.

WebFOCUS will allocate some or all of the following output or work files as temporary data sets:

- HOLD, SAVB, and SAVE files.
- FOCUS data sources created by the CREATE FILE command.
- FOCSTACK, FOCSORT, and HOLDMAST files.
- FOCSML files.
- OFFLINE, SYSIN, and SYSPRINT files.

You can explicitly allocate files in the server JCL. For more information about allocating files see your iWay Server documentation.

**Note:** Work files should not be allocated in the server JCL.

## Application Files Under MVS

This topic describes how WebFOCUS references and searches for your application files.

### Master Files Under MVS

Master Files are comma-delimited files that contain metadata about data sources. When WebFOCUS needs a Master File, it searches for the ddname MASTER. If it does not find the member there, it next searches the ddname HOLDMAST, followed by EDASYNM. If it does not find the member in HOLDMAST either, WebFOCUS returns an error message.

The DCB attributes of the MASTER are: RECFM=FB, LRECL=80, with BLKSIZE a multiple of LRECL.

The partitioned data set (PDS) member names must correspond to the WebFOCUS names of Master Files. For example, the command TABLE FILE CAR requires a member CAR in the MASTER file PDS.

Several partitioned data sets of Master Files can be concatenated under the ddname MASTER, provided that they have identical LRECL and RECFM attributes.

To create and maintain Master Files under TSO, use the ISPF editor.

### *Example* Allocating a Master File Under MVS

A typical allocation for a Master File is:

```
//MASTER DD DSN=qualif.EDAMFD.DATA,DISP=SHR
```

The entire PDS (and not a particular member) must be referenced in the allocation.

## Access Files Under MVS

For some data sources, an Access File supplements a Master File. An Access File includes additional information that completes the description of the data source. For example, it includes the full data source name and location. You need one Master File–and, for some data sources, one Access File–to describe a data source.

An Access File is allocated to ddname ACCESS. The DCB attributes of the ACCESS PDS are: RECFM=FB, LRECL=80, with BLKSIZE a multiple of LRECL.

The PDS member names must correspond to the name of the Master File. For example, the FOCUS command TABLE FILE CAR requires a member CAR in the ACCESS file PDS.

### *Example* Allocating an Access File Under MVS

A typical allocation for an ACCESS file is:

```
//ACCESS DD DSN=prefix.ACCESS.DATA,DISP=SHR
```

The entire PDS must be referenced in the allocation, not a particular member.

## Procedures Under MVS

A procedure can be stored in one or more partitioned data sets allocated to ddname FOCEXEC. The name of the procedure corresponds to the member name.

The DCB attributes: RECFM=FB, LRECL=80, with BLKSIZE a multiple of LRECL.

In TSO, you can also create a procedure using the ISPF editor. In batch mode, you can create procedures with the utility programs IEBPTPCH and IEBUPDTE.

Several partitioned data sets of EDARPC procedures may be concatenated under the ddname of FOCEXEC.

### *Example* Allocating a Procedure Under MVS

A sample allocation for a procedure is:

```
//FOCEXEC DD DSN=qualif.EDARPC.DATA,DISP=SHR
```

## FOCUS Data Sources Under MVS

FOCUS databases contain data written in FOCUS format. The maximum size of a FOCUS database is 2G. Each data source is allocated to a ddname that matches the member name of the file's Master File in the PDS allocated to ddname MASTER. For example, if the data source's Master File has the member name LEDGER, then the data source is allocated to ddname LEDGER. (If you want to override this default, you can issue the USE command, or use the a DATASET attribute in the Master File or Access File. For more information see Chapter 14, *Accessing a FOCUS Data Source*.

FOCUS data sources are formatted in 4096-byte pages. We strongly recommend that you allocate FOCUS data sources in cylinders, if the file is big enough to justify this.

### *Example* Allocating a FOCUS Data Source

The following example applies to a new FOCUS data source:

```
//CAR   DD DSNAME=CAR.FOCUS,UNIT=SYSDA,DISP=(NEW,CTLG),
//   VOL=SER=MYVOL,SPACE=(CYL,(5,5)), LRECL=4096, RECFM=F, BLKSIZE=4096
```

## Using FOCUS Data Sources Under MVS

Existing FOCUS data sources can always be allocated as SHR even when being modified. Concurrent destructive updates are prevented by WebFOCUS itself, by reserving exclusive use of the data source, for update purposes, for the duration of the MODIFY command. At the conclusion of the MODIFY, the user which had possession of the data source relinquishes control, making the data source available to the other users.

While only one user is allowed to update the data source, multiple users may have the data source open for read-only purposes, concurrently. This scheme offers the same protection as exclusive allocation (DISP=OLD), but it is more flexible, because with DISP=OLD, the data source is reserved for the duration of the allocation. (For a batch job, this is the entire time from initiation to termination.)

If a user attempts to modify a FOCUS data source currently being modified by another user or batch job, or if the data source is controlled by a sink machine, the result is a message stating that the data source is in use elsewhere and the MODIFY command flushes to the END statement allowing you to perform other tasks. On the other hand, if a batch job encounters the same situation, it will wait until the data source is free (that is, until the MODIFY currently in control ends or the sink machine is terminated).

When TABLE and TABLEF commands are run for the purpose of locating cross-referenced segments (segment types KU and KM), you must allocate the data source with DISP=OLD or NEW. This is so their addresses may be written into the data source from which they are being cross-referenced. This process, called pointer resolution, does not take place if the disposition of the data source is SHR or if the data source resides on a sink machine.

## External Indexes for FOCUS Data Sources Under MVS

An external index is a FOCUS data source that contains index, field, and segment information for one or more specified FOCUS data sources. The external index is independent of its associated FOCUS data source and is used to improve retrieval performance.

the external index is automatically allocated as a temporary file when it is created using REBUILD. To create a permanent external index, you must allocate it before you issue the REBUILD command.

SORTOUT is a work file that is used during the process to create an external index. You must allocate the work file with the proper amount of space, minus DCB parameters. To estimate the amount of space, use this formula:

```
bytes = (field_length + 20) * number_of_occurrences
```

## Other Supported Data Sources Under MVS

You can use the FOCUS query language to read external files that are not in FOCUS format. WebFOCUS can read: QSAM, ISAM, and VSAM files; IMS, CA-IDMS/DB, ADABAS, TOTAL, SYSTEM 2000, MODEL 204, Millennium, and Supra database files; DB2, SQL/DS, CA-DATACOM/DB, Teradata, and Oracle tables.

The ddname under which you should allocate the external file depends on the type of file. For more information about external files, see your iWay Server documentation.

## WebFOCUS StyleSheet Files Under MVS

WebFOCUS StyleSheets enable you to style and produce reports that highlight key information. With StyleSheets, you specify various stylistic characteristics of a report and style report components individually. You can also use StyleSheets to define a hyperlink from any reporting object.

You can create StyleSheets externally or within a report request. For details, see the *Creating Reports With WebFOCUS Language* manual.

The FOCSTYLE data set stores any external StyleSheets used by FOCUS procedure StyleSheet declarations.

### *Example* Allocating Space for ddname FOCSTYLE Under MVS

The following example shows how to allocate space for the FOCSTYLE data set:

```
//FOCSTYLE DD DSN=qualif.STY.DATA,DISP=SHR
```

where:

*qualif*

Is your site's high-level qualifier.

## Profiles Under MVS

WebFOCUS supports the following levels of profiles to provide flexibility in designing and running applications.

- A **global profile** is a startup file that is automatically created during the installation and configuration of the server. It contains the default environment settings that are required for the proper operation of the server.
- A **group profile** is a file used by the server to specify settings that apply to a server environment, but only for a user of a specific security group.
- A **user profile** is available based on a user ID. A user profile is a file used by the server to specify settings that apply to a server environment, but only for a specific user ID.

For details on profiles see your iWay Server documentation.

# Extract Files Under MVS

Extract files save lines of output during a WebFOCUS request. They contain data taken from existing data sources.

## HOLD Files Under MVS

A HOLD extract file is a sequential data source that contains the results of a report request. It may have a corresponding HOLD Master File. The ddname is HOLD unless you specify the FORMAT option. WebFOCUS provides the DCB parameters in accordance with the record length of the report it is about to store. The DCB BLKSIZE parameter is automatically calculated.

If you specify the FORMAT FOCUS option with HOLD, WebFOCUS creates an extract file and a Master File, each with the file name FOC$HOLD. These files are then used as input to the procedure that creates the final FOCUS file. The new FOCUS and Master Files are created in the user temporary directory.

*Syntax* **How to Create a HOLD File Under MVS**

```
ON TABLE HOLD [AS ddname]
```

where:

*ddname*

Is a name for the HOLD file. If you do not specify a file name, HOLD is used as the default name. Since each subsequent HOLD command overwrites the previous HOLD file, it is useful practice to code a distinct file name in each request to direct the extracted data to a separate file, preventing it from being overwritten.

## SAVB Files Under MVS

A SAVB file is an extract file containing the results of a report request in internal format. That is, all numeric fields are stored in binary, and all character fields are padded with spaces to a multiple of four bytes. The file cannot be printed.

The server dynamically allocates a temporary sequential data set under ddname SAVB or a ddname you supply with the AS ddname option. WebFOCUS allocates five tracks each for primary and secondary space. Record format is variable blocked with record length and blocksize dependent on the record size. To keep the file, use the DYNAM COPY command.

*Syntax* **How to Create a SAVB File Under MVS**

```
ON TABLE SAVB [AS ddname]
```

where:

*ddname*

Is name under which WebFOCUS allocates a temporary sequential data set.

## SAVE Files Under MVS

A SAVE file is an extract file that saves the data of a report, but does not save headings or subtotals, or create a Master File. It is a simple sequential character data file that can be used by other programs, or merged into another data file using a data maintenance request. The default format is simple character, although you can specify formats compatible with many other software products. WebFOCUS saves all columns in the report in printable, character format with no spaces between columns.

A SAVE file contains the external character format equivalent to SAVB. The command syntax and allocations are the same as SAVB. However, the numbers are printable Extended Binary Coded Decimal Interchange Code (EBCDIC) characters and no padding takes place.

*Syntax* **How to Create a SAVE File Under MVS**

```
ON TABLE SAVE [AS filename]
```

where:

AS *filename*

Specifies a file name for your SAVE file. If you do not specify a file name, it defaults to SAVE. If you do not specify a file name, subsequent creations of HOLD files will overwrite each other.

## HOLDMAST Files: Temporary Master Files Under MVS

When HOLD files are created either under the default name HOLD, or a specified name (for example, HOLD AS MYNAME), the description is written into the PDS whose ddname is HOLDMAST. This PDS is exactly like MASTER except that WebFOCUS creates the members. It is usually a temporary file.

The data control block (DCB) attributes of the HOLDMAST are RECFM=FB, LRECL=80, with BLKSIZE a multiple of LRECL. DCB parameters must not be supplied by the user. The server will create the HOLDMAST file with the current DCB.

If HOLDMAST is not allocated when a HOLD file is created, the server will allocate HOLDMAST as a temporary data set with five primary and five secondary tracks.

Whenever WebFOCUS needs the description of a file, it first searches the ddname MASTER. If the member is not found, it then searches the ddname HOLDMAST.

If you want to retain the HOLDMAST file, give it a name and a DISP parameter.

## HOLDSTAT Files Under MVS

HOLDSTAT files enable you to include DBA information and environmental comments in HOLD and PCHOLD Master Files. Member HOLDSTAT in the distributed library EDAMSG.DATA is the default. Alternately, you may create your own HOLDSTAT or another user-specified member in your ERRORS or MASTER PDSs.

The contents of a HOLDSTAT file are included in HOLD and PCHOLD Master Files when the SET HOLDSTAT command is specified to ON or to a member name. For information about the SET HOLDSTAT command, see the *Creating Reports With WebFOCUS Language* manual.

The HOLDSTAT file may exist as a member in the ERRORS or the MASTER library.

A HOLDSTAT file may contain environmental comments like a file header, or the DBA attribute, or both. The supplied HOLDSTAT member contains the following file header with Dialogue Manager system variables:

```
$=================================================================$
$      HOLD file created on &DATE at &TOD by FOCUS &FOCREL       $
$              Database records retrieved= &RECORDS               $
$              Records in the HOLD file = &LINES                  $
$=================================================================$
```

In the HOLD Master File, the comments appear after the FILE and SUFFIX attributes and the DBA information is appended to the end.

### *Reference* HOLDSTAT File Rules Under MVS

If you create your own HOLDSTAT file, the following rules apply:

- Each line of comments must begin with a dollar sign ($) in column 1.
- Comments may not include user-defined variables.
- List the DBA information after any comments. On separate lines, specify the keywords $BOTTOM and END, beginning in column 1, followed by the DBA attribute. The syntax is:

  ```
  $BOTTOM
  END
  DBA=attribute,$
  ```

  where:

  *attribute*
  Is an attribute such as password, user, access, restrict, name, or value.

## Work Files Under MVS

This topic describes characteristics of temporary work files used in the MVS environment.

The available work files are:

- **FOCSTACK** files contain resolved Dialogue Manager procedures.
- **FOCSORT**, allocated as FOCSORT.FOC, may be required for sorting with the TABLE, TABLEF, and MATCH commands. The maximum size of a FOCSORT file is 1G of 256K pages.
- **FOCPOST** files hold sequential output, which is saved by the POST command and read by the PICKUP command.
- **FOCSML** files are used by the Financial Modeling Language (FML) facility.

# Reviewing Attributes of Allocated Files Under MVS

You can check on existing file allocations using the command ? TSO DDNAME. This can be done in the following ways:

| Command | Function |
|---|---|
| `? TSO DDNAME` | Lists allocated files. Produces a listing on file SYSPRINT even if you issue the OFFLINE command. |
| `? TSO DDNAME ddname` | Lists file attributes. Produces a listing on file SYSPRINT even if you issue the OFFLINE command. |
| `-? TSO DDNAME ddname` | Places file attribute information into Dialogue Manager variables. Attributes are returned as values for Dialogue Manager variables. |

## Displaying Allocated Files With ? TSO DDNAME Under MVS

The ? TSO DDNAME command lists all the currently allocated files by ddname, shows the number of occurrences for each and lists their corresponding data set names. It displays currently allocated files.

*Syntax* **How to Display Allocated Files With ? TSO DDNAME**

```
? TSO DDNAME
```

*Example* **Displaying Allocated Files With ? TSO DDNAME**

The following is sample output from a ? TSO DDNAME command.

```
DDNAME              OCCURRENCES        DSNAME

STEPLIB                   2         EDA.V3R2M01.EDALIB.LOAD
                                    INPDQ.PROC.PROCLIB
MASTER                    1         EDA.V3R2M01.EDAMFD.DATA
FOCEXEC                   1         EDA.V3R2M01.EDAMFD.DATA
ERRORS                    1         EDA.V3R2M01.EDAMSG.DATA
FOCSORT                   1         SYS88229.TO95540.RA000.INPDQ.R0000002
OFFLINE                   1         TSOINFOC.TSOINFOC.OFFLINE
SYSUT1                    1         SYS88229.TO95058.RA000.INPDQ.SYSUT1
SYSEDIT                   1         SYS88229.T095058.RA000.INPDO.EDIT
HOLD                      1         SYS88229.TO95058.RA000.INPDQ.R0000003
HOLDMAST                  1         SYS88229.T095044.RA000.INPDQ.R0000004
CAR                       1         EDA.V3R2M01.CAR.FOCUS
```

The DDNAME column lists all allocated file names (ddnames). The OCCURRENCES column shows how many data sets are allocated to the corresponding ddname. This number will be one unless two or more data sets are concatenated under the ddname. The DSNAME column shows the fully qualified data set name (DSN) corresponding to the ddname. For concatenated data sets, all the data set names are shown.

## Displaying File Attributes With ? TSO DDNAME ddname Under MVS

The ? TSO DDNAME ddname command displays attributes of the queried file name (ddname). If the specified ddname was not allocated, the command displays the attributes as either blanks or zeroes.

### *Syntax* How to Display File Attributes With ? TSO DDNAME ddname

```
? TSO DDNAME ddname
```

where:

*ddname*

Is the ddname whose attributes you want to display. You may specify up to eight characters. You may also create a general list of allocated files by ddname by specifying a wildcard character (* or ?) with part of the ddname.

## Displaying File Attributes With ? TSO DDNAME ddname

The following is sample output from a ? TSO DDNAME *ddname* command.

```
DDNAME          =          CAR
DSNAME          =          EDA.VSR2M00.CAR.FOCUS
DISP            =          OLD
DEVICE          =          DISK
VOLSER          =          TSOPAK
DSORG           =          PS
RECFM           =          F
SECONDARY       =            1
ALLOCATION      =          TRACKS
BLKSIZE         =          4096
LRECL           =          4096
TRKTOT          =            2
EXTENTSUSED     =            1
BLKSPERTRK      =           10
TRKSPERCYL      =           15
CYLSPERDISK     =          886
BLKSWRITTEN     =           20
FOCUSPAGES      =            8
```

## Placing File Attributes in Dialogue Manager Commands

The -? TSO DDNAME ddname command places file attribute information in Dialogue Manager variables instead of displaying the information on the terminal. These Dialogue Manager TSO system variables have the same names as the attributes returned by the ? TSO DDNAME ddname command. For a complete list of these variables see *Dialogue Manager TSO System Variables* on page 16-32.

If there is no information for a variable, the variable will contain blanks if it is an alphanumeric variable, or zeroes, if it is a numeric variable.

### *Reference* Dialogue Manager TSO System Variables

| TSO System Variable | Description |
|---|---|
| `&DDNAME` | Queried file name (ddname). |
| `&DSNAME` | Fully qualified data set name. For concatenated data sets, only the first data set name is returned. |
| `&DISP` | Disposition: OLD, NEW, MOD, or SHR. |
| `&DEVICE` | DISK, TAPE, TERM, or READER-PRINTER. |
| `&VOLSER` | Disk or tape volume serial number. |
| `&DSORG` | Data set organization: PS (sequential); IS (indexed sequential); PO (partitioned); U (undefined); DA (direct). |
| `&RECFM` | Record format, for example, F, FB, V, VB. |
| `&SECONDARY` | Quantity specified for secondary allocations. |
| `&ALLOCATION` | Unit of secondary allocation: TRACKS, BLOCKS, or CYLINDER. |
| `&BLKSIZE` | Blocksize. |
| `&LRECL` | Record length, in bytes. |
| `&TRKTOT` | Total number of currently allocated tracks, spanning both primary and secondary allocation. |
| `&EXTENTSUSED` | Total number of currently allocated extents (max 16). |
| `&BLKSPERTRK` | The number of blocks (of BLKSIZE bytes) that can be written onto 1 track of the device. |
| `&TRKSPERCYL` | The number of tracks per cylinder for the device. |

| TSO System Variable | Description |
|---|---|
| &CYLSPERDISK | The number of cylinders per disk for the device. |
| &BLKSWRITTEN | The total number of blocks in the data set, assuming that all blocks are BLKSIZE long. |
| &FOCUSPAGES | Posted only for FOCUS data sources. This number is the total number of 4096-byte pages in the data source. This is the same as the total number of pages shown by the ? FILE file name command. It is also the same as the highest page number shown by the ? FDT file name command. It does not include SHADOW pages and directory. |

***Example*** **Using TSO ddname Variables**

```
1. -? TSO DDNAME &DD.ENTER DDNAME.
2. -IF &DSNAME EQ ' ' GOTO ALLOCATE;
   -TYPE DATASET &DSNAME ALLOCATED TO &DD
   -EXIT
   -ALLOCATE
    .
    .
    .
```

The process is as follows:

1. The command prompts you for the ddname. The information that is entered from the terminal is then placed in the variable &DD. Depending on the value of &DD, the system supplies the information for the variable &DSNAME. If no data set is allocated to the ddname that was supplied, the variable &DSNAME will contain a blank since it is alphanumeric.
2. The command tests if the variable &DSNAME is a blank. If it is blank (that is, no data set was found for the ddname entered by the operator), the procedure jumps to the label -ALLOCATE and continues. If it is not blank (that is, a data set was found for the ddname entered by the operator), the procedure prints a message stating the name of the data set and exits.

For detailed information on Dialogue Manager commands and variables, see Chapter 10, *Managing Flow of Control in an Application*.

## Determining If a Data Source Exists Under MVS

Since you can logically erase an existing data set by issuing a CREATE command against it, it is necessary to have a means of testing for the existence of an OS data set before issuing the CREATE command.

### *Syntax* How to Test for a Data Set

```
? {MVS|TSO} DSNAME datasetname
```

where:

*datasetname*

Is the data set name. If you supply only the unqualified data set name, iWay will take the prefix from the profile to create a fully qualified data set name. Do not specify member names.

### *Syntax* How to Test for a Data Set in Dialogue Manager

```
-? TSO DSNAME dsname
```

where:

*dsname*

Is the data set name.

In this case, there is no output message. The system variable &RETCODE is set and must be tested for the outcome. The possible results follow:

| &RETCODE Value | Equivalent FOCUS Code/Message |
|---|---|
| 0 | (FOC488) Dataset is in catalog: |
| 4 | (FOC489) Dataset is in catalog, but not on volume indicated: |
| 8 | (FOC490) Dataset is not in catalog: |

## Estimating Data Set Sizes to Determine Available Space Under MVS

You can use the file attribute information returned by the ? TSO ddname command to determine the number of records in the data set and to estimate how much available space is left on the currently allocated tracks.

**Note:** The estimates obtained from the formulas which follow are only approximations and do not take into account space that was reused after it was logically vacated by deleted segments.

- FOCUS data source formulas:

  ```
  Available pages (without additional extents) = BLKSWRITTEN -
  FOCUSPAGES
  ```

  ```
  BLKSWRITTEN = BLKSPERTRK x TRKTOT
  ```

- Fixed-block data set formulas:

  ```
  RECSPERBLK = BLKSIZE/LRECL
  ```

  ```
  No. of records = BLKSWRITTEN x RECSPERBLK
  ```

  ```
  No. of free blocks = (TRKTOT x BLKSPERTRK) - BLKSWRITTEN
  ```

- Variable-length blocked data sets:

  The formulas for fixed-block data sets (that is, SAVE files) usually apply to variable-length blocked (VB) data sets as well. This is true because VB data sets usually have the same length blocks even though the description implies otherwise.

- If SET SHADOW=ON, use the following formula:

  ```
  (FOCUSPAGES x 2) + 3
  ```

# WebFOCUS Files Under UNIX

WebFOCUS files fall into three categories: application files, extract files, and work files. This topic contains information about files and ways to define them in the UNIX environment.

Some files are automatically allocated by WebFOCUS, but there are times when you must explicitly define a file and its location. You can do this by using the FILEDEF command. For details on the FILEDEF command see *Assigning a Logical Name Under UNIX* on page 16-38.

*Reference*

## WebFOCUS Files Under UNIX

The following tables describe the WebFOCUS files that you will use under UNIX. WebFOCUS uses file extensions to distinguish different file types.

### Application Files

| Extension | Description |
|---|---|
| `.mas` | Master File. |
| `.acx` | Access File. |
| `.fex` | Procedure. |
| `.foc` | FOCUS data source and external indexes. |
| `.dat` | Sequential data source. |
| `.sty` | WebFOCUS StyleSheet file. |
| `.err` | Error messages file or help text. |
| `.prf` | Profile. |
| `.htm`<br>`.html`<br>`.jpeg`<br>`.gif`<br>`.css`<br>`.js`<br>`.class`<br>`.jar` | Files displayed in a Web browser. |

**Extract Files**

| File | Description |
|---|---|
| `*HOLD` | Contains data saved using the HOLD command. |
| `*SAVB` | Contains data saved using the SAVB command. |
| `*SAVE` | Contains data saved using the SAVE command. |
| `HOLDMAST` | Temporary Master File for HOLD files. |

| Extension | Description |
|---|---|
| `.ftm` | Extract or temporary file. |

**Work Files**

| File | Description |
|---|---|
| `FOCSTACK` | Contains resolved Dialogue Manager procedures. |
| `FOCSORT` | Used during sorting. |
| `FOCPOST` | Holds sequential output. |
| `FOCSML` | Used by the Financial Modeling Language (FML). |
| `OFFLINE` | Used when the SET PRINT parameter is OFFLINE. |

## Dynamically Defining Files Under UNIX

WebFOCUS will automatically allocate some or all of the following output or work files:

- HOLD, SAVB, and SAVE files.
- FOCUS data sources.
- Work files such as FOCSORT.
- OFFLINE, SYSIN, and SYSPRINT files.

## Assigning a Logical Name Under UNIX

WebFOCUS files fall into three categories: application files, extract files, and work files. This topic contains information about files and ways to define them in the UNIX environment.

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 FILEDEF command. A FILEDEF command can be issued in a procedure and lasts for a single request.

It is recommended that instead of including a FILEDEF 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 allows you to make changes to your FILEDEF commands globally instead of changing the FILEDEF information in each procedure.

### *Syntax* How to Assign a Logical Name With FILEDEF Under UNIX

```
FILEDEF ddname devicetype fileid [[LRECL n] [RECFM fm] [APPEND]]
```

where:

*ddname*

Is the logical name. It may contain 1 to 8 alphanumeric characters.

*devicetype*

Identifies the type of device with which to interact. Specify DISK for a file that resides on disk.

*fileid*

Is the physical name of the file.

LRECL *n*

Specifies the record length, *n*, in bytes. This parameter is optional. If you omit it, the default is 80 bytes. The left parenthesis preceding the optional parameters is required.

RECFM *fm*

Describes the record format. Specify F for fixed format, V for variable format. This parameter is optional. If you omit it, the default is fixed format. The left parenthesis preceding the optional parameters is required.

`APPEND`

Opens the specified file and adds new material at the end. This parameter is optional. If you omit it and the specified file exists, it will be overwritten. The left parenthesis preceding the optional parameters is required.

FOCUS data sources (files with the .foc extension) that do not conform to the default naming convention are identified with the USE command, not FILEDEF. For information on the USE command, see Chapter 14, *Accessing a FOCUS Data Source*.

## Displaying Current ddnames Assigned With FILEDEF

The ? FILEDEF command displays the ddnames assigned for various files, input and output.

Developer Studio can create many FILEDEFs. Four of the system-created FILEDEFs are apparent using the ? FILEDEF command.

### *Syntax* How to Display Current ddnames

```
? FILEDEF
```

## Clearing FILEDEF Assignments

You can clear FILEDEF assignments.

### *Syntax* How to Clear a Logical Name

```
FILEDEF ddname CLEAR
```

where:

*ddname*

Is the logical name. It may contain 1 to 8 alphanumeric characters.

`CLEAR`

Clears the specified ddname.

# Application Files Under UNIX

This topic describes how WebFOCUS references and searches for application files.

When you do not change the default file type, you can prepare a report without issuing a FILEDEF command or other communication. When you change the default file type of a FOCUS data source, or the data source does not reside in a standard search path, you must issue the USE command. The USE command is discussed in Chapter 14, *Accessing a FOCUS Data Source*.

## Master Files Under UNIX

A Master File contains metadata about a data source, and is identified with a file name and extension, for example, *filename*.mas. It consists of attributes that describe a data source. A Master File and the data source that it describes usually have the same name. A Master File must consist of 80-byte, fixed length records.

The description of a data source and all files it cross-references must be available whenever you refer to the data source.

For more information about how to generate Master Files, see the *Describing Data With WebFOCUS Language* manual.

WebFOCUS searches for a Master File on the path of the WebFOCUS Reporting Sever, using either EDAPATH or APP PATH logic. For details on search path logic, see *Storing and Searching for Application Files* in Chapter 2, *WebFOCUS Application Logic*.

### *Reference* Naming a Master File Under UNIX

The following rules apply:

- The file name can be up to 64 characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are therefore not recommended or supported.
- The file extension must be mas for a Master File, since WebFOCUS searches for a Master File with that extension by default.

## Access Files Under UNIX

For some data sources, an Access File supplements a Master File. An Access File includes additional information that completes the description of the data source. For example, it includes the full data source name and location. You need one Master File–and, for some data sources, one Access File–to describe a data source.

WebFOCUS searches for an Access File on the path of the WebFOCUS Reporting Server using either EDAPATH or APP PATH logic.

### *Reference* Naming an Access File Under UNIX

The following rules apply:

- The file name can be up to 64 characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are therefore not recommended or supported.
- The file extension must be acx since WebFOCUS searches for a FOCUS data source with that extension by default.

## Procedures Under UNIX

A procedure contains your report requests. A procedure can contain up to 80 bytes on a single line and cannot have control characters. A procedure is identified with a file name and extension, for example, *filename*.fex.

A procedures can be simultaneously accessed and executed by all users. The device and directory must precede the procedure file name and extension for the server to locate the procedure. If you specify a disk and a directory, you must include both the file name and the file type (.fex) when calling the procedure.

WebFOCUS searches for a procedure on the path of the WebFOCUS Reporting Server, using either EDAPATH or APP PATH logic.

### *Reference* Naming a Procedure Under UNIX

The following rules apply:

- File names can be up to 64 characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are not recommended or supported.
- The file extension must be fex, since WebFOCUS searches for a procedure with that extension by default.

## FOCUS Data Sources Under UNIX

FOCUS data sources contain data written in FOCUS format. All FOCUS data sources have a record length of 4096 and a fixed length record format. The maximum size of a FOCUS data source is 1G or 256K pages.

A FOCUS data source is identified with a file name and extension, for example, *filename*.foc. The name of a FOCUS data source usually matches the name of the corresponding Master File. For example, if the Master File is ledger.mas, then the FOCUS data source is ledger.foc. You can override this default with the USE command, described in Chapter 14, *Accessing a FOCUS Data Source*.

WebFOCUS searches for a FOCUS data source on the platform with the WebFOCUS Reporting Server or subserver.

### *Reference* Naming a FOCUS Data Source Under UNIX

The following rules apply:

- The file name can be up to 64 characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are therefore not recommended or supported.
- The file extension must be foc since WebFOCUS searches for an Access File with that extension by default.

## External Indexes for FOCUS Data Sources Under Windows UNIX

An external index is a FOCUS data source that contains index, field, and segment information for one or more specified FOCUS data sources. The external index is independent of its associated FOCUS data source and is used to improve retrieval performance. In UNIX, the external index is automatically allocated as a permanent file when it is created using REBUILD.

For more information, see the EDA Server documentation for UNIX.

## Sequential Data Sources Under UNIX

WebFOCUS recognizes sequential data sources formatted in the following ways:

- Fixed format, in which each field occupies a predefined position in the record.
- Free-format, also known as comma-delimited, in which fields can occupy any position in a record.

For details see the *Describing Data With WebFOCUS Language* manual.

## WebFOCUS StyleSheets Under UNIX

WebFOCUS StyleSheets enable you to style and produce reports that highlight key information. With StyleSheets, you specify various stylistic characteristics of a report and style report components individually. You can also use StyleSheets to define a hyperlink from any reporting object.

You can create StyleSheets externally or within a report request. For details, see the *Creating Reports With WebFOCUS Language* manual.

WebFOCUS searches for a WebFOCUS StyleSheet on the path of the WebFOCUS Reporting Server, using either EDAPATH or APP PATH logic.

*Reference*

### Naming a StyleSheet Under UNIX

The following rules apply:

- File names can be up to eight characters long. The first character should be a letter, and the remaining characters can be any combination of letters, numbers, and underscores. Other special characters may be used, but could cause problems and are not recommended or supported.
- The file extension must be sty since WebFOCUS searches for a StyleSheet with that extension by default.

## Profiles Under UNIX

WebFOCUS supports the following levels of profiles to provide flexibility in designing and running applications.

- A **global profile** is a startup file that is automatically created during the installation and configuration of the server. It contains the default environment settings that are required for the proper operation of the server.
- A **group profile** is a file used by the server to specify settings that apply to a server environment, but only for a user of a specific security group.
- A **user profile** is available based on a user ID. A user profile is a file used by the server to specify settings that apply to a server environment, but only for a specific user ID.

For details on profiles see the *EDA Server Administration* manual.

# Extract Files Under UNIX

WebFOCUS creates all extract files in a temporary directory on the WebFOCUS Reporting Server, or in a user-defined location on the WebFOCUS Reporting Server.

## HOLD Files Under UNIX

A HOLD extract file is a sequential data source that contains the results of a report request. It may have a corresponding HOLD Master File. The extension for a HOLD file is ftm unless you specify the FORMAT option. If a Master File is created, it has the same name as the extract file, with the extension mas.

If you specify the FORMAT FOCUS option with HOLD, WebFOCUS creates an extract file and a Master File, each with the file name FOC$HOLD. These files are then used as input to the procedure that creates the final FOCUS file. The new FOCUS and Master Files are created in the user temporary directory.

### *Syntax* How to Create a HOLD File Under UNIX

```
ON TABLE HOLD [AS filename]
```

where:

*filename*

Is a name for the HOLD file. If you do not specify a file name, HOLD is used as the default name. Since each subsequent HOLD command overwrites the previous HOLD file, it is useful practice to code a distinct file name in each request to direct the extracted data to a separate file, preventing it from being overwritten.

For the complete syntax with all options, see the *Creating Reports With WebFOCUS Language* manual.

## SAVB Files Under UNIX

A SAVB file is an extract file containing the results of a report request in internal format. That is, all numeric fields are stored in binary, and all character fields are padded with spaces to a multiple of four bytes. The file cannot be printed.

### *Syntax* How to Create a SAVB File Under UNIX

```
ON TABLE SAVB [AS filename]
```

where:

*filename*

Is the name of the file. The default is SAVB. The default extension is FTM.

## SAVE Files Under UNIX

A SAVE file is an extract file that saves the data of a report, but does not save headings, or subtotals, or create a Master File. It is a simple sequential character data file and can be used by other programs, or merged into another data file using a data maintenance request. The default format is simple character, although you can specify formats compatible with many other software products. WebFOCUS saves all columns in the report in printable, character format with no spaces between columns.

A SAVE file contains the external character format equivalent to SAVB. The command format and allocations are the same as SAVB. However, the numbers are printable Extended Binary Coded Decimal Interchange Code (EBCDIC) characters and no padding takes place.

### *Syntax* How to Create a SAVE File Under UNIX

```
ON TABLE SAVE [AS filename]
```

where:

*filename*

Is the name of the file. The default is SAVE. The default extension is FTM.

## HOLDMAST Files Under UNIX

A HOLDMAST file is a temporary Master File. It is created with the FILEDEF HOLDMAST command, which specifies its location.

If the HOLD file was created under the default name HOLD, the FILEDEF command can be used in conjunction with the FILEDEF HOLDMAST syntax to designate a logical name to the HOLD file and respective Master File.

### *Syntax* How to Specify the Location for a HOLDMAST File Under UNIX

```
FILEDEF HOLDMAST DIR /directory_name
```

where:

*directory_name*

Is a valid directory path.

### *Example* Allocating a Logical Name and Location for a HOLDMAST File Under UNIX

The following example sets a logical name with FILEDEF and specifies a location for a HOLDMAST file:

```
1. FILEDEF HOLDMAST DIR /tmp
2. FILEDEF SALES DISK /tmp/sales.ftm
    TABLE FILE GGSALES
    PRINT GGSALES
    ON TABLE HOLD AS SALES
    END
```

The numbers listed next to the code correspond with the following notes:

1. Specifies the location of the Master File in the \tmp directory. WebFOCUS uses the same logical name, SALES, specified in the second line, to name the temporary Master File.
2. Specifies SALES as the logical name for the physical data source. The HOLD command creates the temporary Master File sales.mas and the data source sales.ftm.

## Work Files Under UNIX

WebFOCUS creates work files as required by your application. WebFOCUS creates all extract files in a temporary directory on the WebFOCUS Reporting Server, or in a user-defined location on the WebFOCUS Reporting Server.

The available work files are:

- **FOCSTACK** files contain resolved Dialogue Manager procedures.
- **FOCSORT**, allocated as FOCSORT.FOC, may be required for sorting with the TABLE, TABLEF, and MATCH commands. The maximum size of a FOCSORT file is 1G of 256K pages.
- **FOCPOST** files hold sequential output, which is saved by the POST command and read by the PICKUP command.
- **FOCSML** files are used by the Financial Modeling Language (FML) facility.
- **OFFLINE** sends output to a file when the  to send output to a file when the PRINT parameter is set to OFFLINE in WebFOCUS Developers Studio. Allocate the file to ddname OFFLINE using a FILEDEF command after the offline has been closed with the OFFLINE CLOSE command.

### *Example* Sending Output With the OFFLINE Command Under UNIX

The following command sends further report output to the file EMPL OUTPUT A with a fixed length record format and a record length of 80 bytes:

```
OFFLINE CLOSE
FILEDEF OFFLINE PRINTER EMPL OUTPUT A (RECFM F LRECL 80
```

CHAPTER 17

# Euro Currency Support

**Topics:**

- Integrating the Euro Currency
- Converting Currencies
- Creating the Currency Data Source
- Identifying Fields That Contain Currency Data
- Activating the Currency Data Source
- Processing Currency Data
- Querying the Currency Data Source in Effect
- Punctuating Numbers
- Using the Extended Currency Symbol Format

This topic describes how to create and use a currency data source to convert to and from the new euro currency.

# Integrating the Euro Currency

With the introduction of the euro currency, businesses need to maintain books in two currencies, add new fields to their data source designs, and perform new types of currency conversions. You can perform currency conversions according to the rules specified by the European Union. To do this:

1. Create a currency data source with the currency IDs and exchange rates you will use. See *Creating the Currency Data Source* on page 17-4.
2. Identify fields in your data sources that represent currency data. See *Identifying Fields That Contain Currency Data* on page 17-7.
3. Activate your currency data source. See *Activating the Currency Data Source* on page 17-9.
4. Perform currency conversions. See *Processing Currency Data* on page 17-10.

**Note:** As the euro symbol becomes available to operating systems, Information Builders will support it.

# Converting Currencies

Euro currency was introduced in Euroland on January 1, 2002, and on July 1, 2002 it became the only legal tender. All monetary transactions now occur in euro currency.

The European Union has set fixed exchange rates between the euro and the traditional national currency in each of the 12 adopting member nations. Although 12 or more currencies in the European Union use the euro, more than 100 currencies have a recognized status worldwide. In addition, you may need to define custom currencies for other applications.

While the exchange rates within Euroland remain fixed, exchange rates between the euro and non-euro countries continue to vary freely and, in fact, several rates may be in use at one time (for example, actual and budgeted rates).

You identify your currency codes and rates by creating a currency data source. For more information, see *Creating the Currency Data Source* on page 17-4.

## *Reference* Currency Conversion Rules

The European Union has established the following rules for currency conversions:

- The exchange rate must be specified as a decimal value, *r*, with six significant digits. This rate will establish the following relationship between the euro and the particular national currency:

  1 euro = *r* national units

- To convert from the euro to the national unit, multiply by *r* and round the result to two decimal places.
- To convert from the national currency to the euro, divide by *r* and round the result to two decimal places.
- To convert from one national currency to another, first convert from one national unit to the euro, rounding the result to three decimal places (your application rounds to exactly three decimal places). Then convert from the euro to the second national unit, rounding the result to two decimal places. This two-step conversion process is *triangulation*.

## *Example* Performing Triangulation

The following example illustrates triangulation. In this case, 10 US dollars (USD) are converted to French francs (FRF). The exchange rate for USD to euros (EUR) is 1.17249. The exchange rate for FRF to euros is 6.55957.

- The 10 USD are converted to EUR by dividing the 10 USD by the EUR exchange rate of 0.8840:

  ```
  EUR = 10 / 0.8840
  ```

  ```
  This results in 11.3122 euros.
  ```

- The euros are converted to FRF by multiplying the above result by the exchange rate of FRF for euros (6.55957):

  ```
  FRF = 11.3122 * 6.55957
  ```

  The result is 74.26. FRF. This means 74.26 FRF are equivalent to 10 USD.

## Creating the Currency Data Source

For each type of currency you need, you must supply the following values in your currency data source:

- A three-character code to identify the currency, such as USD for US dollars or BEF for Belgian francs. (For a partial list of recognized currency codes, see*Sample Currency Codes* on page 17-6.)
- One or more exchange rates for the currency.

There is no limit to the number of currencies you can add to your currency data source, and the currencies you can define are not limited to official currencies and. Therefore, the currency data source can be fully customized for your applications.

The currency data source can be any type of data source your application can access (for example, FOCUS, FIX, DB2, or VSAM). The currency Master File must have one field that identifies each currency ID you will use and one or more fields to specify the exchange rates.

We strongly recommend that you create a separate data source for the currency data rather than adding the currency fields to another data source. A separate currency data source enhances performance and minimizes resource utilization because the currency data source is loaded into memory before you perform currency conversions.

## *Syntax* How to Create a Currency Data Source

```
FILE = name, SUFFIX = suffix,$
FIELD  = CURRENCY_ID,, FORMAT =  A3,                    [ACTUAL = A3 ,]$
FIELD  = rate_1,,      FORMAT = {D12.6|numeric_format1}, [ACTUAL = A12,]$
   .
   .
   .
FIELD  = rate_n,,      FORMAT = {D12.6|numeric_formatn}, [ACTUAL = A12,]$
```

where:

*name*

Is the name of the currency data source.

*suffix*

Is the suffix of the currency data source. The currency data source can be any type of data source your application can access.

CURRENCY_ID

Is the required field name. The values stored in this field are the three-character codes that identify each currency, such as USD for U.S. dollars. Each currency ID can be a universally recognized code or a user-defined code.

**Note:** The code EUR is automatically recognized; you should *not* store this code in your currency data source. See *Sample Currency Codes* on page 17-6 for a list of common currency codes.

*rate_1*...*rate_n*

Are types of rates (such as BUDGET, FASB, ACTUAL) to be used in currency conversions. Each rate is the number of national units that represent one euro.

*numeric_format1*...*numeric_formatn*

Are the display formats for the exchange rates. Each format must be numeric. The recommended format, D12.6, ensures that the rate is expressed with six significant digits as required by the European Union conversion rules. Do not use Integer format (I).

ACTUAL A*n*

Is required only for non-FOCUS data sources.

**Note:** The maximum number of fields in the currency data source must not exceed 255 (that is, the CURRENCY_ID field plus 254 currency conversion fields).

## *Reference* Sample Currency Codes

On January 1, 1999, Euroland set exchange rates between the euro and other currencies. Countries included in Euroland as of that date are marked with an asterisk (*). Their rates are fixed and will not change; the rates for other countries change over time.

| Currency Name | Currency Code | Rate |
|---|---|---|
| American dollar | USD | .974298 |
| Austrian schilling | ATS | 13.7603 |
| Belgian franc* | BEF | 40.3399 |
| British pound | GBP | .625152 |
| Canadian dollar | CAD | 1.54504 |
| Danish krone | DKK | 7.42659 |
| Dutch guilder* | NLG | 2.20371 |
| Dutsche mark* | DEM | 1.95583 |
| Euro | EUR | 1 |
| Finnish markka | FIM | 5.94573 |
| French franc* | FRF | 6.55957 |
| Greek drachma* | GRD | 340.750 |
| Irish pound* | IEP | 0.787564 |
| Italian lira* | ITL | 1936.27 |
| Japanese yen | JPY | 118.377 |
| Luxembourg franc* | LUF | 40.3399 |
| Norwegian kroner | NOK | 7.34864 |
| Portuguese escudo* | PTE | 200.482 |
| Spanish peseta* | ESP | 166.386 |
| Swedish krona | SEK | 9.20906 |
| Swiss franc | CHF | 1/4634 |

### *Example* Specifying Currency Codes and Rates in a Master File

The following Master File for a comma-delimited currency data source specifies two rates for each currency, ACTUAL and BUDGET:

```
FILE = CURRCODE, SUFFIX = COM,$
FIELD = CURRENCY_ID,,    FORMAT = A3,    ACTUAL = A3 ,$
FIELD = ACTUAL, ALIAS =, FORMAT = D12.6, ACTUAL = A12 ,$
FIELD = BUDGET, ALIAS =, FORMAT = D12.6, ACTUAL = A12 ,$
```

The following is sample data for the currency data source defined by this Master File:

```
FRF,  6.55957,  6.50000,$
USD,  0.974298,  1.00000,$
BEF,  40.3399, 41.00000,$
```

## Identifying Fields That Contain Currency Data

Once you have created your currency data source, you must identify the fields in your data sources that represent currency values. To designate a field as a currency-denominated value (a value that represents a number of units in a specific type of currency) add the CURRENCY attribute to one of the following:

- The FIELD specification in the Master File.
- The left side of a DEFINE or COMPUTE.

### *Syntax* How to Identify a Currency Value

Use the following syntax to identify a currency-denominated value

**In a Master File**

```
FIELD = currfield,, FORMAT = numeric_format, ..., CURR =
{curr_id|codefield} ,$
```

**In a DEFINE in the Master File**

```
DEFINE currfield/numeric_format CURR curr_id = expression ;$
```

**In a DEFINE FILE command**

```
DEFINE FILE filename
currfield/numeric_format CURR curr_id = expression ;
END
```

### In a COMPUTE command

```
COMPUTE currfield/numeric_format CURR curr_id = expression ;
```

where:

*currfield*

Is the name of the currency-denominated field.

*numeric_format*

Is a numeric format. Depending on the currency denomination involved, the recommended number of decimal places is either two or zero. Do not use I or F format.

CURR

Indicates that the field value represents a currency-denominated value. CURR is an abbreviation of CURRENCY, which is the full attribute name.

*curr_id*

Is the three-character currency ID associated with the field. In order to perform currency conversions, this ID must either be the value EUR or match a CURRENCY_ID value in your currency data source.

*codefield*

Is the name of a field, qualified if necessary, that contains the currency ID associated with *currfield*. The code field should have format A3 or longer and is interpreted as containing the currency ID value in its first three bytes. For example:

```
FIELD = PRICE,, FORMAT = P12.2C, ..., CURR = TABLE.FLD1,$
    .
    .
    .
FIELD = FLD1,, FORMAT = A3, ...,$
```

The field named FLD1 contains the currency ID for the field named PRICE.

*filename*

Is the name of the file for which this field is defined.

*expression*

Is a valid expression.

### *Example* Identifying a Currency-Denominated Field

The following Master File contains the description of a field named PRICE that is denominated in U. S. dollars.

```
FILE=CURRDATA,SUFFIX=COM,$
FIELD=PRICE, FORMAT=P17.2 , ACTUAL=A5, CURR=USD,$
.
.
.
```

## Activating the Currency Data Source

Before you can perform currency conversions, you must specify the currency data source by setting the EUROFILE parameter with the SET command. By default, the EUROFILE parameter is not set.

Issue the SET command in a procedure or any supported profile. In Developer Studio, it can also be issued from the Command Console. It cannot be set within a report request.

Once a data source is activated, you can access a different currency data source by reissuing the SET command.

**Note:** The EUROFILE parameter must be set alone. For example, appending an additional SET parameter will cause the additional parameter setting to be lost.

### *Syntax* How to Activate Your Currency Data Source

```
SET EUROFILE = {ddname|OFF}
```

where:

*ddname*

Is the name of the Master File for the currency data source. The ddname must refer to a data source known to and accessible by your application in read-only mode.

OFF

Deactivates the currency data source and removes it from memory.

### *Reference* EUROFILE Error Messages and Notes

- Issuing the SET EUROFILE command when the currency data source Master File does not exist generates the following error message:

  `(FOC205) THE DESCRIPTION CANNOT BE FOUND FOR FILE NAMED: ddname`

- Issuing the SET EUROFILE command when the currency Master File specifies a FOCUS data source and the associated FOCUS data source does not exist generates the following error message:

  `(FOC036) NO DATA FOUND FOR THE FOCUS FILE NAMED: name`

## Processing Currency Data

After you have created your currency data source, identified the currency-denominated fields in your data sources, and activated your currency data source, you can perform currency conversions.

Each currency ID in your currency data source generates a virtual conversion function whose name is the same as its currency ID. For example, if you added BEF to your currency data source, a virtual BEF currency conversion function will be generated.

The euro function, EUR, is supplied automatically with your application. You do not need to add the EUR currency ID to your currency data source.

The result of a conversion is calculated with very high precision, 31 to 36 significant digits, depending on platform. The precision of the final result is always rounded to two decimal places. In order to display the result to the proper precision, its format must allow at least two decimal places.

## Syntax How to Process Currency Data

### In a procedure

```
DEFINE FILE filename
result/format [CURR curr_id] = curr_id(infield, rate1 [,rate2]);
END
```

or

```
COMPUTE result/format [CURR curr_id] = curr_id(infield, rate1 [,rate2]);
```

### In a Master File

```
DEFINE result/format [CURR curr_id] = curr_id(infield, rate1 [,rate2]);$
```

where:

*filename*

Is the name of the file for which this field is defined.

*result*

Is the converted currency value.

*format*

Is a numeric format. Depending on the currency denomination involved, the recommended number of decimal places is either two or zero. The result will always be rounded to two decimal places, which will display if the format allows at least two decimal places. Do not use an Integer or Floating Point format.

*curr_id*

Is the currency ID of the result field. This ID must be the value EUR or match a currency ID in your currency data source; any other value generates the following message:

```
(FOC263)  EXTERNAL FUNCTION OR LOAD MODULE NOT FOUND: curr_id
```

**Note:** The CURR attribute on the left side of the DEFINE or COMPUTE identifies the result field as a currency-denominated value which can be passed as an argument to a currency function in subsequent currency calculations. Adding this attribute to the left side of the DEFINE or COMPUTE does not invoke any format or value conversion on the calculated result.

*infield*

Is a currency-denominated value. This input value will be converted from its original currency to the *curr_id* denomination. If the *infield* and *result* currencies are the same, no calculation is performed and the *result* value is the same as the *infield* value.

*rate1*

Is the name of a rate field from the currency data source. The *infield* value is divided by its currency's *rate1* value to produce the equivalent number of euros.

If *rate2* is not specified in the currency calculation and triangulation is required, this intermediate result is then multiplied by the *result* currency's *rate1* value to complete the conversion.

In certain cases, you may need to provide different rates for special purposes. In these situations you can specify any field or numeric constant for *rate1* as long as it indicates the number of units of the *infield* currency denomination that equals one euro.

*rate2*

Is the name of a rate field from the currency data source. This argument is only used for those cases of triangulation in which you need to specify different rate fields for the *infield* and *result* currencies. It is ignored if the euro is one of the currencies involved in the calculation.

The number of euros that was derived using *rate1* is multiplied by the *result* currency's *rate2* value to complete the conversion.

In certain cases, you may need to provide different rates for special purposes. In these situations you can specify any field or numeric constant for *rate2* as long as it indicates the number of units of the *result* currency denomination that equals one euro.

## *Reference* Currency Calculation Error Messages

Issuing a report request against a Master File that specifies a currency code not listed in the active currency data source generates the following message:

```
(FOC1911) CURRENCY IN FILE DESCRIPTION NOT FOUND IN DATA
```

A syntax error or undefined field name in a currency conversion expression generates the following message:

```
(FOC1912)  ERROR IN PARSING CURRENCY STATEMENT
```

## *Example* Using the Currency Conversion Function

Assume that the currency data source contains the currency IDs USD and BEF, and that PRICE is denominated in Belgian francs as follows:

```
FIELD = PRICE, ALIAS=, FORMAT = P17.2, CURR=BEF,$
```

- The following example converts PRICE to euros and stores the result in PRICE2 using the BUDGET conversion rate for the BEF currency ID:

  ```
  COMPUTE PRICE2/P17.2 CURR EUR = EUR(PRICE, BUDGET);
  ```

- This example converts PRICE from Belgian francs to US dollars using the triangulation rule:

  ```
  DEFINE PRICE3/P17.2 CURR USD = USD(PRICE, ACTUAL);$
  ```

  First PRICE is divided by the ACTUAL rate for Belgian francs to derive the number of euros rounded to three decimal places. Then this intermediate value is multiplied by the ACTUAL rate for US dollars and rounded to two decimal places.

- The following example uses a numeric constant for the conversion rate:

  ```
  DEFINE PRICE4/P17.2 CURR EUR = EUR(PRICE,5);$
  ```

- The next example uses the ACTUAL rate for Belgian francs in the division and the BUDGET rate for US dollars in the multiplication:

  ```
  DEFINE PRICE5/P17.2 CURR USD = USD(PRICE, ACTUAL, BUDGET);$
  ```

## *Example* Converting U.S. Dollars to Euros, French Francs, and Belgian Francs

### First

Create a currency data source that identifies the currency and one or more exchange rates. (See *Creating the Currency Data Source* on page 17-4 for details.) The following sample data source is named CURRCODE:

```
FILE = CURRCODE, SUFFIX = COM,$
FIELD = CURRENCY_ID,,    FORMAT = A3,    ACTUAL = A3 ,$
FIELD = ACTUAL, ALIAS =, FORMAT = D12.6, ACTUAL = A12 ,$
FIELD = BUDGET, ALIAS =, FORMAT = D12.6, ACTUAL = A12 ,$
```

### Second

Create a data source that contains the values to be converted. (See *Identifying Fields That Contain Currency Data* on page 17-7 for details.) The following sample data source is named CURRDATA:

```
FILE=CURRDATA,SUFFIX=COM,$
FIELD=PRICE, FORMAT=P17.2 , ACTUAL=A5, CURR=USD,$
```

**Third**

Create a request that uses the currency data source to convert the currency values contained in the data source containing these values. The following procedure converts PRICE to euros, French francs, and Belgian francs. The numbers on the left correspond to the notes explaining the code.

```
   -* THE FOLLOWING ALLOCATIONS ARE FOR RUNNING UNDER OS/390
1. -* DYNAM ALLOC FILE CURRCODE DA USER1.FOCEXEC.DATA(CURRCODE) SHR REU
2. -* DYNAM ALLOC FILE CURRDATA DA USER1.FOCEXEC.DATA(CURRDATA) SHR REU

    -* THE FOLLOWING ALLOCATIONS ARE FOR RUNNING UNDER WINDOWS NT
1.     FILEDEF CURRCODE DISC C:\IBI\APPS\GGDEMO\CURRCODE.COM
2.    FILEDEF CURRCODE DISC C:\IBI\APPS\GGDEMO\CURRDATA.COM

3. SET EUROFILE = CURRCODE

   DEFINE FILE CURRDATA
4. PRICEEUR/P17.2 CURR EUR = EUR(PRICE, ACTUAL);
    END

    TABLE FILE CURRDATA
    PRINT PRICE PRICEEUR AND COMPUTE
5. PRICEFRF/P17.2 CURR FRF = FRF(PRICE, ACTUAL);
    PRICEBEF/P17.2 CURR BEF = BEF(PRICE, ACTUAL);
    END
```

The report request executes as follows:

1. The FILEDEF or DYNAM command informs the operating system of the location of the CURRCODE data source.
2. The FILEDEF command informs the operating system of the location of the CURRDATA data source.
3. The SET command specifies the currency data source as CURRCODE.
4. This line calls the EUR function, which converts U. S. dollars to euros.
5. The next two lines are the conversion functions that convert euros into their equivalent in French and Belgian Francs.

The output is:

You cannot use the derived euro value PRICEEUR in a conversion from USD to BEF. PRICEEUR has two decimal places (P17.2), not three, as the triangulation rules require.

## Querying the Currency Data Source in Effect

You can issue a query to determine what currency data source is in effect. To do this, issue the ? SET ALL query command or the ? EUROFILE query command.

*Syntax* **How to Determine the Currency Data Source in Effect**

```
? SET EUROFILE
```

*Example* **Determining the Currency Data Source in Effect**

Assume the currency data source is named CURRCODE.

If you issue the following commands:

```
SET EUROFILE = CURRCODE
? SET EUROFILE
```

the output is:

# Punctuating Numbers

Countries differ in how they punctuate numbers, and you can reflect these differences in your reports using Continental Decimal Notation (CDN) which is specified with the CDN SET parameter. The CDN SET allows you to choose to punctuate numbers with a combination of commas, decimals, spaces, and single quotation marks.

The CDN SET parameter can be used in a report request but is not supported in DEFINE or COMPUTE commands.

**Note:** The punctuation specified by the CDN parameter also determines the punctuation used in numbers affected by the CENT-ZERO SET parameter.

## *Syntax* How to Determine the Punctuation of Large Numbers

```
SET CDN = option
```

where:

*option*

Determines the punctuation used in numerical notation. The options are:

ON uses CDN. For example, the number 3,045,000.76 is represented as 3.045.000,76.

OFF turns CDN off. For example, the number 3,045,000.76 is represented as 3,045,000.76. This value is the default.

SPACE separates groups of three significant digits with a space instead of a comma, and marks a decimal position with a comma instead of a period. For example, the number 3,045,000.76 is represented as 3 045 000,76.

QUOTE separates groups of three significant digits with a single quotation mark instead of a comma, and marks a decimal position with a comma instead of a period. For example, the number 3,045,000.76 is represented as 3’045’000,76.

## *Example* Displaying Numbers Using Continental Decimal Notation

The following table shows how 1234.56 is displayed depending on the setting of CDN.

| CDN Setting | Result |
|---|---|
| OFF | 1,234.56 |
| ON | 1.234,56 |
| SPACE | 1 234,56 |
| QUOTE | 1’234,56 |

### *Example* Determining the Punctuation of Large Numbers

In the following request, CDN is set to ON which punctuates numbers using a period to separate thousands, and a comma to separate decimals.

```
SET CDN = ON
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME SALARY
END
```

The output is:

| LAST_NAME | FIRST_NAME | SALARY |
|---|---|---|
| STEVENS | ALFRED | $11.000,00 |
| SMITH | MARY | $13.200,00 |
| JONES | DIANE | $18.480,00 |
| JONES | DIANE | $17.750,00 |
| BANNING | JOHN | $29.700,00 |
| IRVING | JOAN | $26.862,00 |
| IRVING | JOAN | $24.420,00 |
| ROMANS | ANTHONY | $21.120,00 |
| MCCOY | JOHN | $18.480,00 |
| BLACKWOOD | ROSEMARIE | $21.780,00 |
| MCKNIGHT | ROGER | $16.100,00 |
| MCKNIGHT | ROGER | $15.000,00 |
| CROSS | BARBARA | $27.062,00 |
| CROSS | BARBARA | $25.775,00 |

## Using the Extended Currency Symbol Format

You can select a currency symbol to display in your report output regardless of the default currency symbol configured in the NLS (National Language Support) configuration. You can do this by using the extended currency symbol format in place of the floating dollar (M) or non-floating dollar (N) display options.

When using the floating dollar (M) or non-floating dollar (N) display options, FOCUS prints the currency symbol associated with the default currency symbol. For example, when using a United States code page, the dollar sign is displayed. However, the extended currency symbol format allows you to display a selected currency symbol regardless of the default currency associated with your application. Using the extended currency symbol format, you can display the symbol for a United States dollar, a British pound, a Japanese yen, or the euro.

The extended currency support symbol formats are available for numeric formats (I, D, F, and P).

## *Reference* Usage Notes for Extended Currency Symbol Formats

The following rules apply:

- Format specifications cannot be longer than eight characters.
- The extended currency option must be the last option in the format.
- The extended currency symbol format cannot be specified in a format that includes the floating (M) or non-floating display (N) option.
- The selected extended currency symbols must be supported by the fonts accessible to your emulation software or printer.
- A non-floating currency symbol displays only on the first line of each report page. If you use field-based reformatting to display multiple currency symbols in one report column, only the symbol associated with the first row will display. In this situation, you should use floating currency symbols.
- Lower case letters are transmitted as upper case letters by the terminal I/O procedures. Therefore, the fixed extended currency symbols can only be specified in a procedure.

## *Syntax* How To Display an Extended Currency Symbol

*numericformat*`!`*option*

where:

*numericformat*

Is any valid numeric format (data types I, D, F, and P).

`!`

Is required.

*option*

Determines the currency symbol to display, and specifies whether the symbol is floating or non-floating. The options are

`d` displays a non-floating dollar sign.

`D` displays a floating dollar sign.

`e` displays a non-floating euro symbol.

`E` displays a floating euro symbol.

`l` displays a non-floating British pound symbol.

`L` displays a floating British pound symbol.

`y` displays a non-floating Japanese yen symbol.

`Y` displays a floating Japanese yen symbol.

## *Example* Displaying Extended Currency Symbols

The following request displays the Japanese yen on the report row that represents Japan, the British pound on the row that represents England, and the euro on the row that represents Italy.

```
DEFINE FILE CAR
CFORMAT/A8 = DECODE COUNTRY('ENGLAND' 'F12.1C!L' 'JAPAN' 'D12!Y'
             ELSE 'D12.2!E');
END

TABLE FILE CAR
PRINT SALES/CFORMAT DEALER_COST/CFORMAT
BY COUNTRY
  WHERE COUNTRY EQ 'ENGLAND' OR 'JAPAN' OR 'ITALY'
  WHERE SALES GT 0
END
```

The output is:

| COUNTRY | SALES | DEALER_COST |
|---|---|---|
| ENGLAND | £12,000.0 | £11,194.0 |
| ITALY | _4,800.00 | _4,915.00 |
| | _12,400.00 | _5,660.00 |
| | _13,000.00 | _5,660.00 |
| JAPAN | ¥43,000 | ¥2,626 |
| | ¥35,030 | ¥2,886 |

CHAPTER 18

# Editing Application Components as Text in Developer Studio

**Topics:**

- Text Editor
- The Other 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

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.

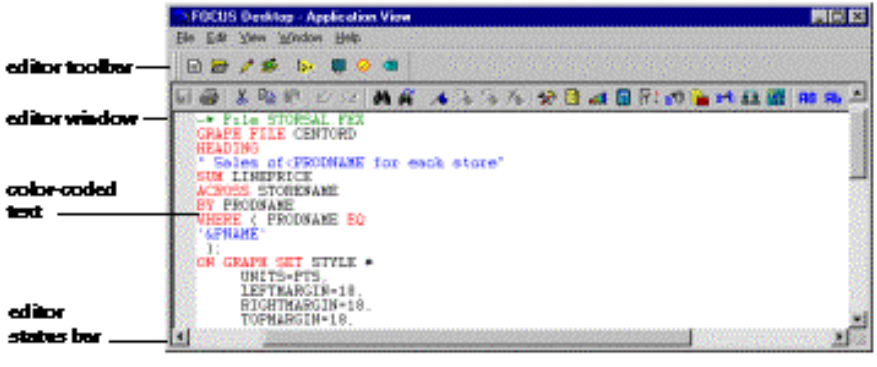
Since the text editor is fully integrated with the graphical toolset, changes you make to source code are immediately reflected in the graphical tools, and vice versa.

# Text Editor

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, Let, Filedef, 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 18-14.
- 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 18-8.

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 18-14 for details.

# 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 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 Chapter 10, *Managing Flow of Control in an Application*.

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 FOCEXEC, the text appears as a Comment component (see *How to Create a Comment Component* on page 18-4). However, if you type Dialogue Manager code, it appears in a component called Other. For related information, see Chapter 4, *Creating a Reporting Procedure*.

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.

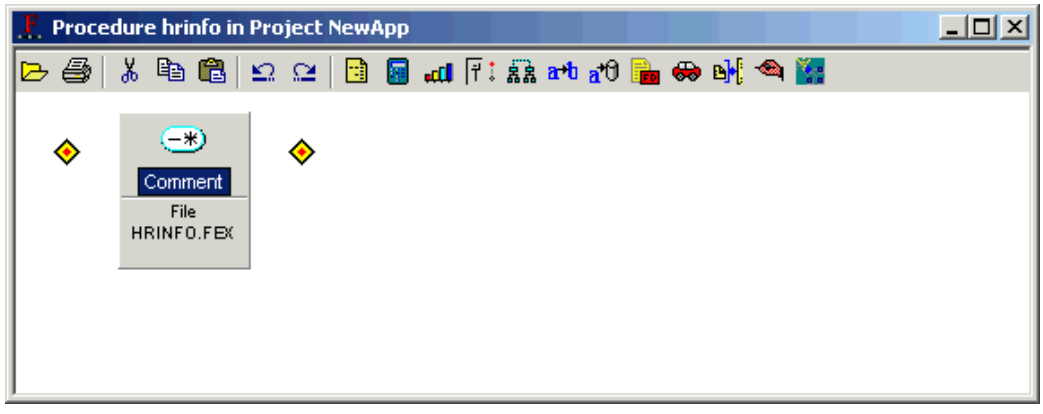
***Procedure*** **How to Use the Other Component**

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.

## Creating a Text File or a Procedure Component as Text

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 18-13.

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 Filename 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.

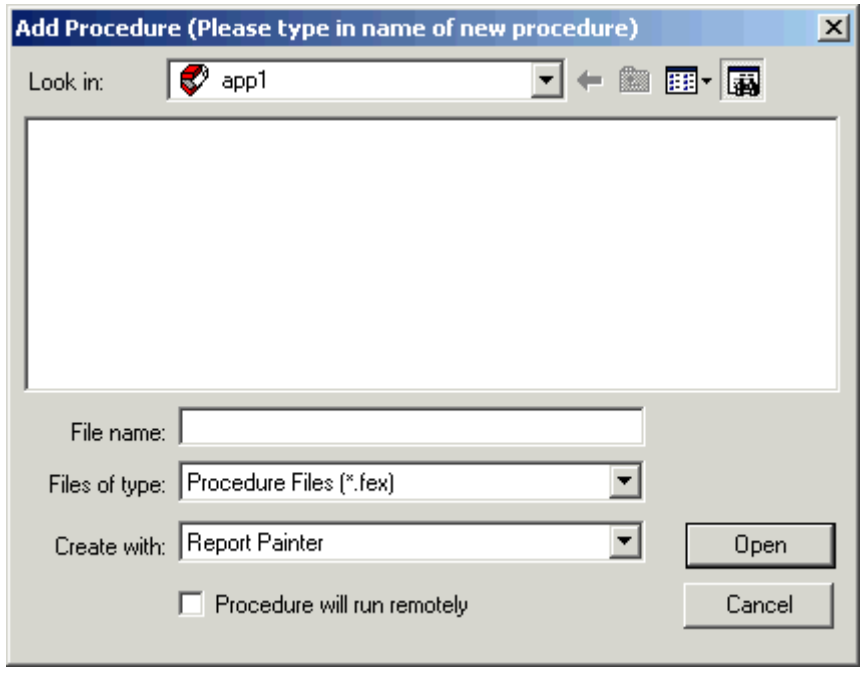
The new file is saved in the selected folder with the appropriate extension.

### *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

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.

   or

1. Highlight a file, such as a procedure or Master File, in an Explorer folder.
2. Select *Edit as Text* from the File menu.

   or

1. Choose *TED* from the Command menu. The Open dialog box displays.
2. Specify the type of file you want to view or edit. FOCEXEC is the default.
3. 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 18-12.

## Changing Text Color and Case Size

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.

| Syntax Coloring Options | Description |
|---|---|
| Text | Regular or default text in the Text Editor. |
| Text Selection | Text that has been highlighted in the Text Editor. |
| Number | Numbers in the text editor. |
| Operator | Operators in the text editor. |
| Comment | Any text that follows a hyphen (-) and an asterisk (*). For Master Files (.MAS), any text that follows a dollar sign ($). |
| Keyword | Words in the FOCUS language. |
| End of Line | Text designating the last line of syntax. |

| Syntax Coloring Options | Description |
|---|---|
| Double Quoted String | Any text between double quotation marks. |
| Single Quoted String | Any text between single quotation marks. |
| Quotes | Any text between a single or double quotation marks. |
| Default Text | All other text. |
| Background | The space that all text displays on. |

*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 Turn On Case Sensitivity in the Text Editor

1. With the text editor open, select *Options* from the View menu.

   The Editor Options dialog box opens.

2. Select *Mixed Case*.
3. Click *OK*.

## *Procedure* How to Change Text to Use Upper Case 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 Lower Case 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.

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 paged. 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

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

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. A report or graph's results display 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. Select *Run* from the File menu.

   or

   Click the *Run* icon on the toolbar. The results display in the appropriate viewer.

### *Procedure* How to Run a Procedure Component as Code

1. Select the procedure, right-click it and select Edit as 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—Report, Define, Join, Set, etc.).

### *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, Filedef, Use, Let, 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 Open 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, Filedef, Use, Let, 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.

CHAPTER 19

# Using the Command Console

**Topics:**

- Command Console Window
- Working in the Command Console

The Command Console enables you send commands directly to a server from Developer Studio. This benefits users who have worked with Information Builders' products and are already familiar with the command syntax.

The command Console sends output to your Web browser. Output can include error messages, responses to query commands, and report output. If you open the Command Console before you run a procedure, it will also serve as a troubleshooting and debugging tool.

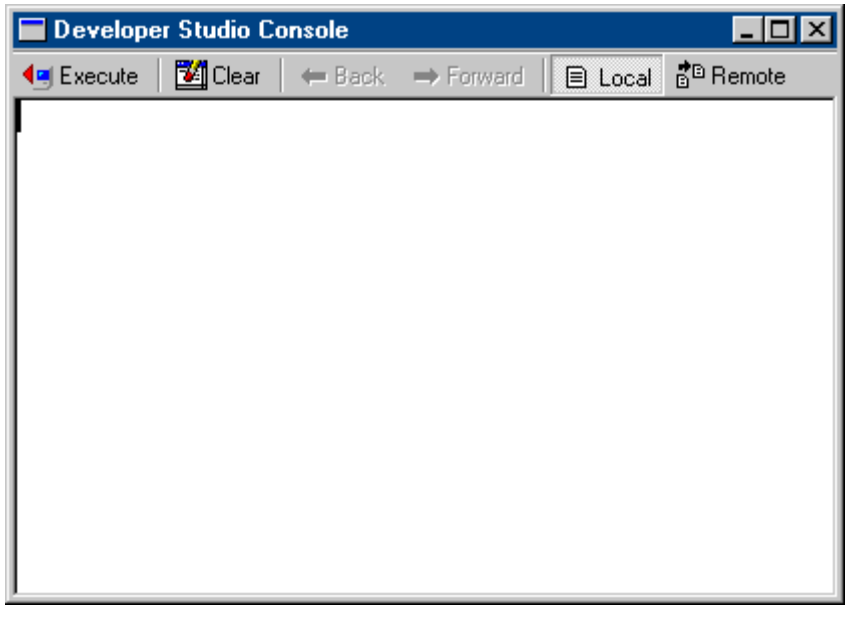
# Command Console Window

The Command Console is a command entry window that allows you to send commands directly to the WebFOCUS Reporting Server without creating a new procedure. When you send commands to the server from the Command Console, the results appear in your Web browser.

For detailed information on syntax and commands, see the following WebFOCUS manuals:

- *Creating Reports With WebFOCUS Language*.
- *Describing Data With WebFOCUS Language*.

The Console window displays as the following:



## *Reference* Console Toolbar

The following table describes the tool buttons on the Console toolbar.

| Button | Operation |
|---|---|
| Execute | Executes a command. |
| Clear | Clears the Output window. |
| Back | Displays the last command executed from the Command Console. |
| Forward | Displays the command next in the list of those executed from the Command Console. |
| Local | Processes commands against the server located on your computer. |
| Remote | Processes commands against a server located remotely. |

## *Procedure* How to Access the Command Console

Select *Command Console* from the Command menu.

or

Click the button on the toolbar.

# Working in the Command Console

You can enter commands and control command processing from the Command Console. You can also change elements in the Command Console window to suit your requirements and preferences.

*Reference* **Rules for Entering Commands in the Command Console**

The following applies when entering commands in the command console.

- The Console is case sensitive. Therefore, you must enter text in the case required for processing. For example, keywords must be entered in uppercase characters.
- The Command Console is not interactive. Therefore, you must enter all required commands in sequence and execute them as an entire process. For example, you must enter all steps required to execute the FOCUS REBUILD command; the Console does not prompt you for information.
- You can copy and cut text from the Command Console or an outside window or and paste it in the Command Console.
- You can drag and drop command syntax within the Command Console.

*Procedure* **How to Enter Commands**

Click in the Console window, then type and execute commands.

*Procedure* **How to Copy, Cut, and Paste Commands**

1. Highlight the text you wan to paste to the Console window.
2. Select *Copy* or *Cut* from the Edit menu.
3. Position the cursor in the Console window, and select *Paste* from the Edit menu.

*Procedure* **How to Drag and Drop Command Syntax**

To drag and drop:

1. Highlight the text you want to drop into the Command Console.
2. Using the mouse, place the cursor on the selected text, hold down the left mouse button, and drag the cursor to where you wish to place the text.
3. Release the mouse button. The text is placed in the new position.

## *Procedure* How to Execute a Command

Click *Execute* on the Console window toolbar.

or

Press Ctrl+Enter.

**Tip:** To execute a command remotely, press Shift+Ctrl+Enter.

## *Procedure* How to Clear the Output Window

Click the *Clear* button on the Console Window toolbar.

or

Press Ctrl+Alt+C.

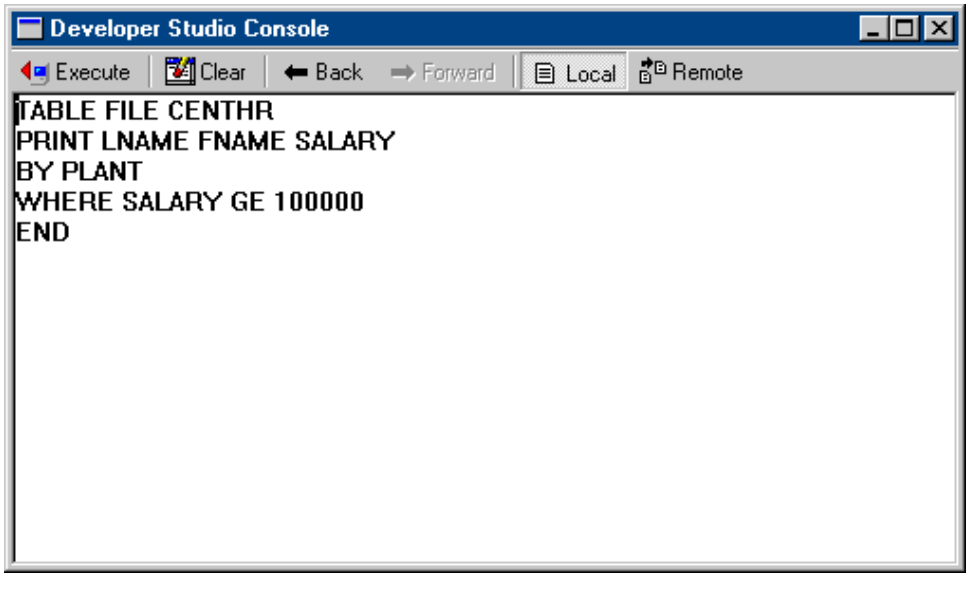
or

Select *Clear Output Window* from the View menu.

## *Example* Executing a Report Request in the Console Window

Consider this simple report request executed from the Command Console.

1. Enter the following commands in the Command Console window:



**2.** Click *Execute* or press Ctrl+Enter to execute the report request.

The Command Entry window is cleared and the report is then displayed in the browser window:

| PAGE 1 | | | |
|---|---|---|---|
| Plant Location | Last Name | First Name | Salary |
| DAL | MERCER | TIM | 100,800.00 |
| LA | MITCHELL | ORVILLE | 110,616.00 |
| | JOHNSON | GREG | 100,800.00 |
| ORL | GARVER | PAUL | 100,800.00 |
| | ROHDE | RICHARD | 115,200.00 |

CHAPTER 20

# Creating an Update Application With Update Assist

**Topics:**

- Using Update Assist
- Calling an Update Assist Procedure From a WebFOCUS Report

The 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.

# Using Update Assist

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.

*Procedure* **How to Create an Update Application With the Update Assist**

1. Create a project. For information on creating a project, see *How to Create a Project Directory* on page 3-5.
2. Add the Master File that you want to base your Update Assist application on to your project. See *How to Add a Master File to a Project* on page 3-13 for details on adding a data source to your project.
3. Right-click the data source, and click *Update Assist*….

   The Update Assist (Step 1 of 6) - Select Segment(s) for Auto Update window opens. For details, see *Update Assist (Step 1 of 6) - Select Segment(s) for Auto Update Window* on page 20-4.
4. In the Select lowest segment or hierarchy you would like to auto update... section, select the segment you want to include that is lowest on the list.

   The segment you select and all segments above it appear in the Segments section.
5. In the Segments section, select the segment that contain the fields you want to update, and click *Next* >.

   **Note:** Update Assist will create a separate procedure for each segment that you want to modify. These procedures are, by default, named after the segment.

   The Update Assist (Step 2 or 6) - Segment & Field Options window opens. For details, see *Update Assist (Step 2 of 6) - Segment & Field Options Window* on page 20-5.
6. In the Update Segment Options section, for each segment, select the capabilities you want your update application to have. The options are:

   *Add* allows your application to add fields to the segment.

   *Update* allows your application to update fields in the segment.

   *Delete* allows your application to delete fields from the segment.

   *Partial Key* allows the user to search for a specific record.

**7.** In the Field View Options section, for each field, select the options you want for each field:

*Visible* makes the field visible to the user.

*Changeable* allows the user to change the field. This option is available only if *Add* or *Update* was selected in the Update Segment Options section.

*Validation* applies a validation technique to the field. The options are:

- *Automatic* validates the user's entry against the field format defined in the Master File.
- *Range* allows you to define a numeric range between which data is valid for the field. When *Range* is selected, the Field Validation - Range dialog box opens. See *Field Validation - Range Dialog Box* on page 20-7 for details.
- *Static List* allows you to supply a list of valid values for the field. When *Static List* is selected, the Field Validation - List dialog box opens. See *Field Validation - List Dialog Box* on page 20-8 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.
- *None* does not perform a validation.

**8.** Click *Next* >.

The Update Assist (Step 3 of 6) - Navigation Options window opens. For details, see *Update Assist (Step 3 of 6) - Navigation Options Window* on page 20-9.

**9.** Select the look of your user interface from the supplied examples, or click *Do not generate any user interface for this project* to create an application without a user interface. Then click *Next* >.

The Update Assist (Step 4 of 6) - Template and Style Options window opens..

**10.** Select a color scheme from the supplied examples, and click *Next* >.

The Update Assist (Step 5 of 6) - Output File Options window opens. For details, see *Update Assist (Step 5 of 6) - Output File Options Window* on page 20-10.

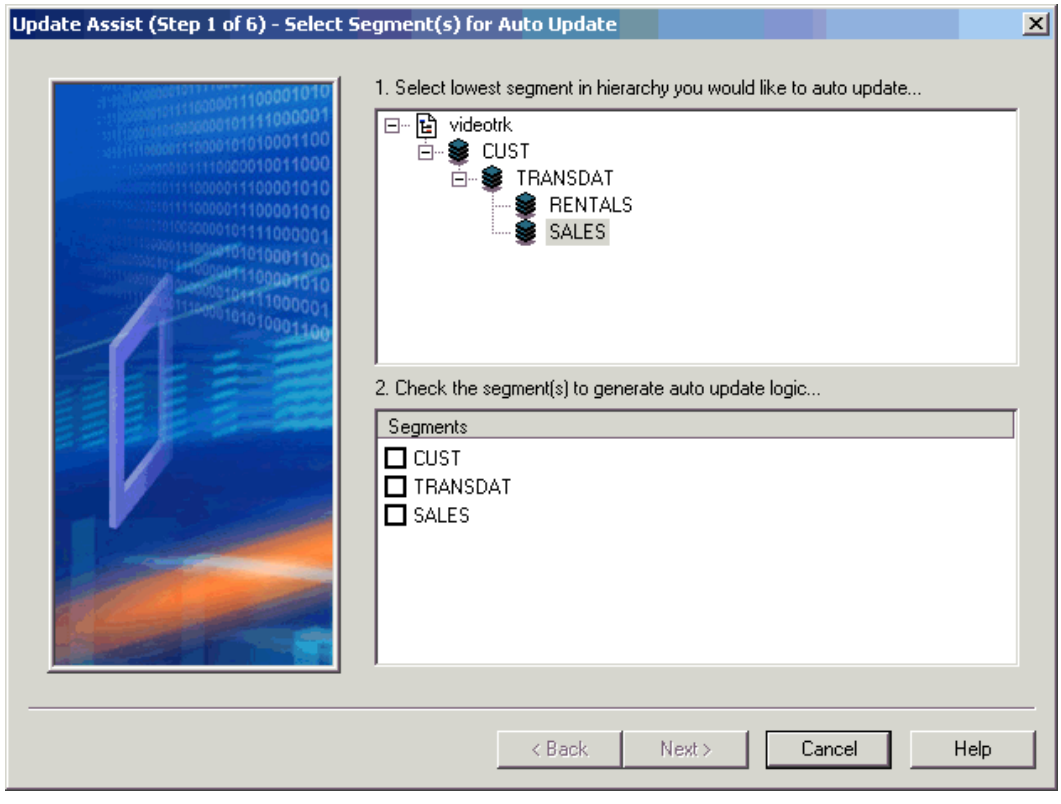
**11.** Review the list of files that your application will update, and rename these files if desired, then click *Next* >.

The Update Assist (Step 6 of 6) - Summary window opens. For details, see *Update Assist (Step 6 of 6) - Summary Window* on page 20-11.

**12.** Review the summary of options you have chosen.

**13.** Click *Finish*.

## *Reference* Update Assist (Step 1 of 6) - Select Segment(s) for Auto Update Window




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 here, so you can select it.

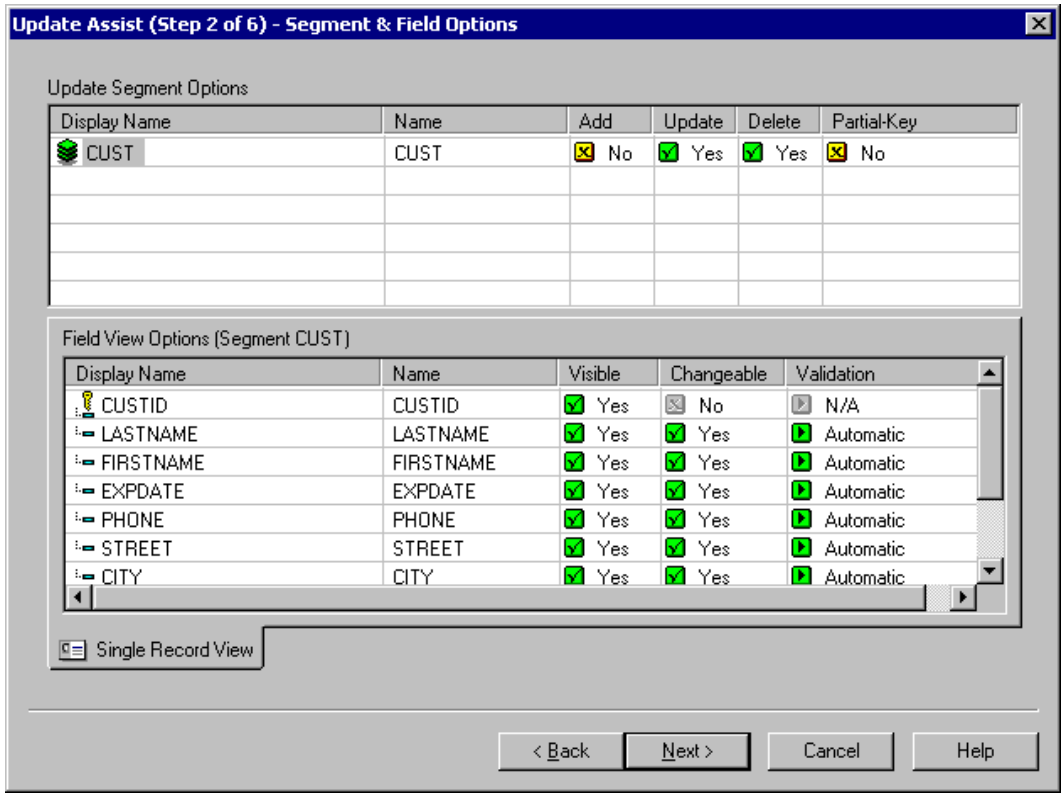
If this is a hierarchical file or a joined view, you will see more than one segment. Select the *lowest* segment that contains fields that you want to update (by lowest, we mean the one farthest down in the list).

**2. Check the segment(s) to generate auto update logic...**

Contains the selected segment, plus any segments that are above it. Fields contained in a segment selected in this box display in the next window (Step 2 of 6).

- If this is a flat file or a relational file, only one segment will appear. Select the segment.
- For a hierarchical file, there may be more than one segment. Select the segments that contain the fields you want to update.

## *Reference* Update Assist (Step 2 of 6) - Segment & Field Options Window



This window contains the following fields/options:

**Update Segment Options**

Contains a list of the segments you selected in the previous window.

**Display Name**

Is the segment name that is displayed to the user. By default, this box contains the name of the segment, but you can rename the field by right-clicking the name and selecting *Rename* from the pop-up menu.

**Name**

Is the name of the segment.

**Add**

Determines whether your application can add fields to this segment.

**Update**

Determines whether your application can change fields in this segment.

**Delete**

Determines whether your application can delete fields from this segment.

**Partial Key**

Allows the user to search for a specific record.

**Field View Options**

Contains the fields in the segment that you selected in Update Segment Options section.

**Display Name**

Is the field name that is displayed to the user. By default, this box contains the name of the field, but you can rename the field by right-clicking the name and selecting *Rename* from the pop-up menu.

**Name**

Is the name of the field.

**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.

**Validation**

Applies a validation technique which verifies the value an user enters for the field. This option is available only if **Changeable** is set to **Yes**. The options for **Validation** are:

**Automatic** 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. When **Range** is selected, the Field Validation - Range dialog box opens. See *Field Validation - Range Dialog Box* on page 20-7 for details. This option is best used for numeric fields.

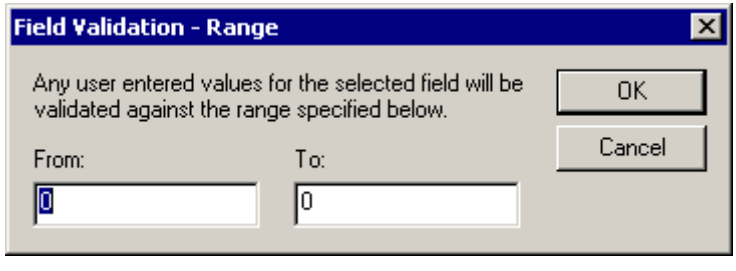
**Static List** allows you to supply a list of valid values from which the user selects at run time. When **Static List** is selected, the Field Validation - List dialog box opens. See *Field Validation - List Dialog Box* on page 20-8 for details.

**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.

**None** does not perform a validation.

**Required field** specifies that the user must supply a value for the field.

## *Reference* Field Validation - Range Dialog Box




This dialog box contains the following fields/options:

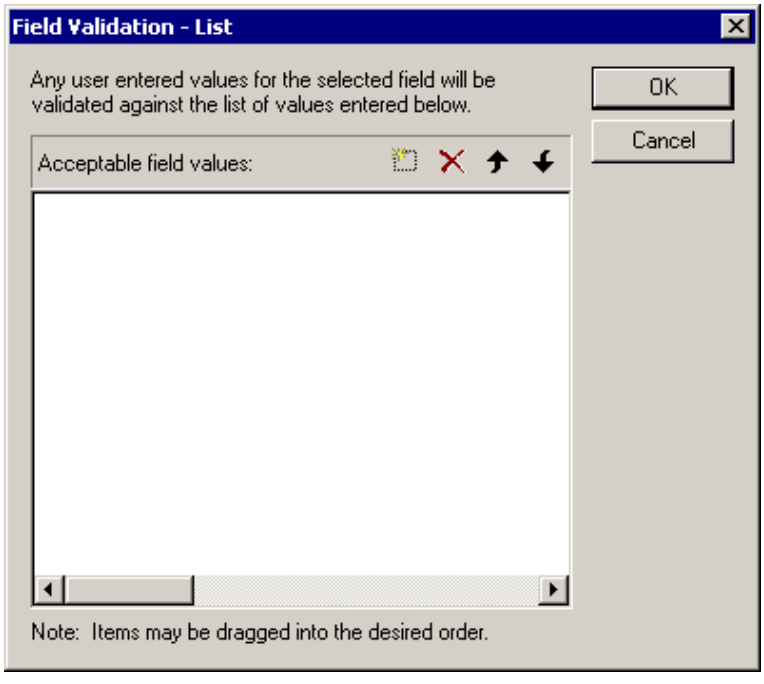
**From:**

Is the value that marks the beginning of the acceptable range of values.

**To:**

Is the value that marks the end of the acceptable range of values.

## *Reference* Field Validation - List Dialog Box




This dialog box has the following fields/options:

**Acceptable field values**

Contains the list of values you have supplied for the user to select from.

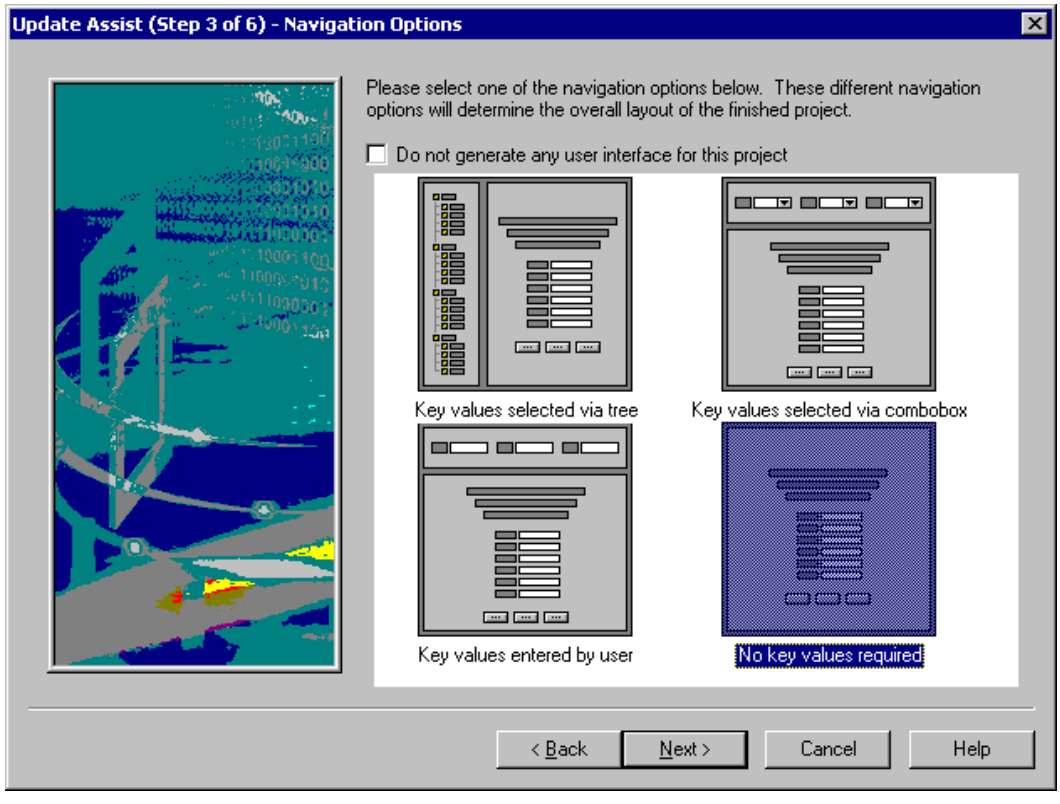
Adds a new value to the list of acceptable field values. Type to enter some text and press Enter.

Deletes a selected value.

Moves a selected list value up in the list

Moves a selected list value down in the list.

## *Reference* Update Assist (Step 3 of 6) - Navigation Options Window




This window contains the following fields/options:

**Do not generate any user interface for this project**

Generates a project with no user interface.

**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.

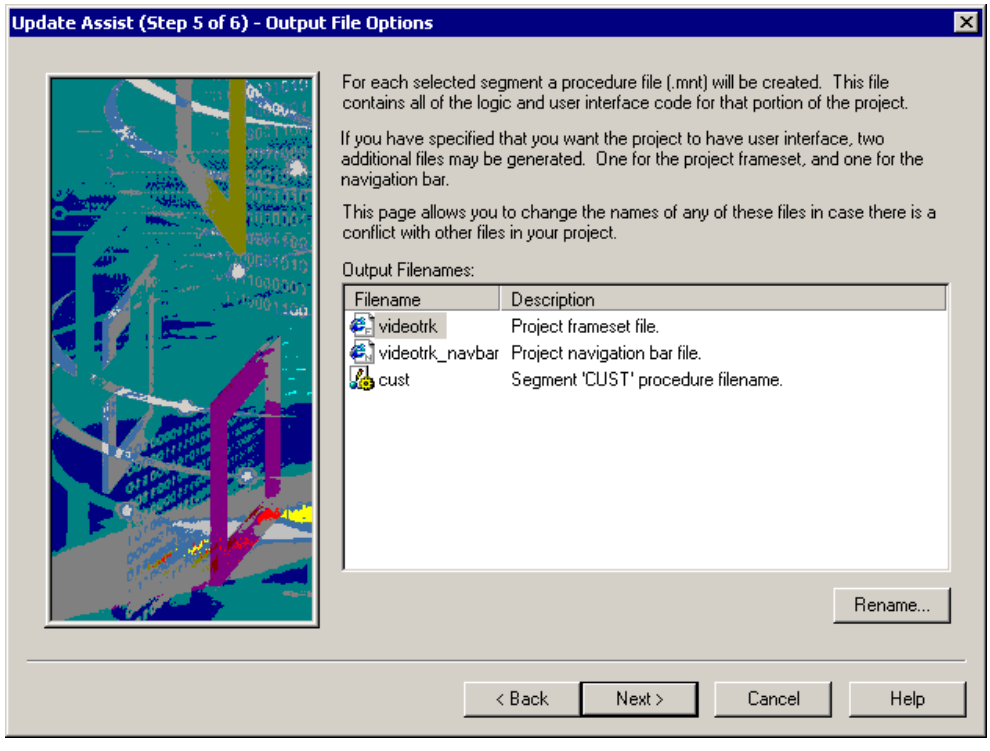
**Key values entered by user**

Generates a form in which the user selects records by entering key values.

**No key values required**

Generates a Maintain procedure that is expecting 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 will call this type of Update Assist project, see *Calling an Update Assist Procedure From a WebFOCUS Report* on page 20-12.

## *Reference* Update Assist (Step 5 of 6) - Output File Options Window




This window contains the following fields/options:

**Output Filenames:**

Lists the files that will be created by the update application.

**Filename**

Is the file name that will be created. This name can be changed.

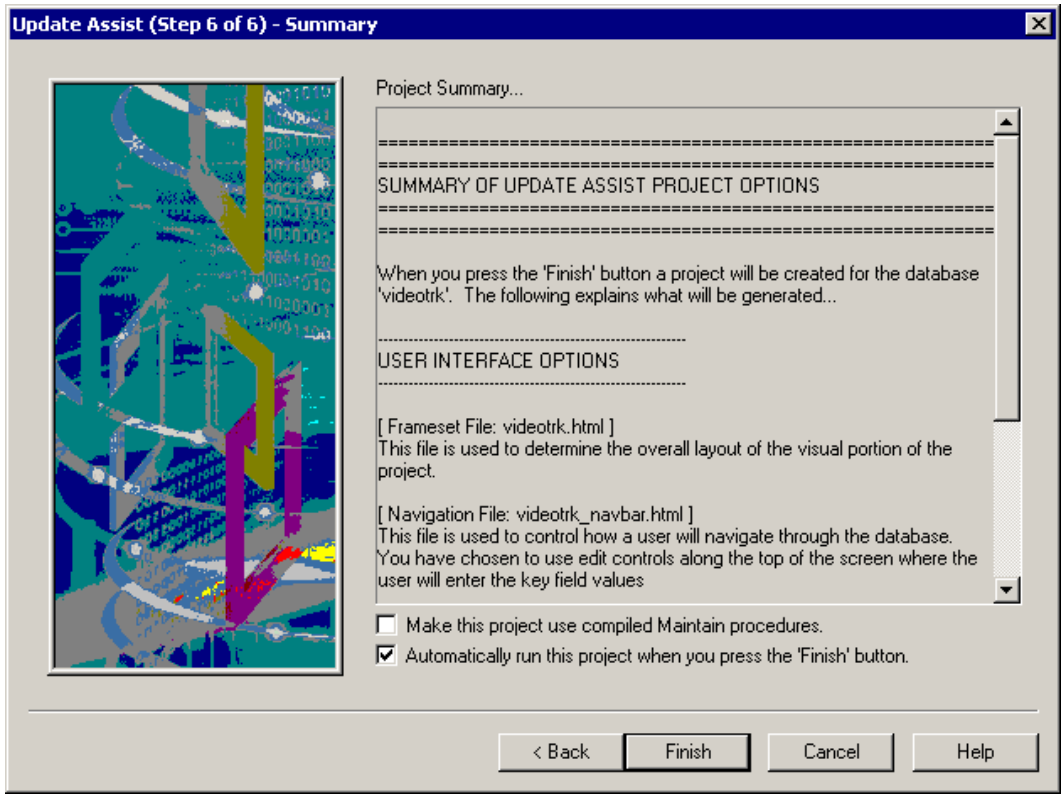
**Description**

Is a description of the file name that will be created.

**Rename**

Allows you to change the selected file name.

## *Reference* Update Assist (Step 6 of 6) - Summary Window




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 that contains 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. You would deselect this if you wanted to perform further editing on your project files.

# Calling an Update Assist Procedure From a WebFOCUS Report

One of the ways to use an Update Assist procedure is to call it from a WebFOCUS report. You can set up the WebFOCUS report so that an user can click on a row of 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.

   For complete instructions on creating an Update Assist procedure, see *How to Create an Update Application With the Update Assist* on page 20-2.

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 don't want to view them in the results, you can make them invisible).

3. Select the column you want to be "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 an 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:

      `fieldname_s_Edit`

      where:

      `fieldname` is the name of the key field (a key field uniquely identifies a record in a segment).

      `_s_Edit` Must be typed *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.
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 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.

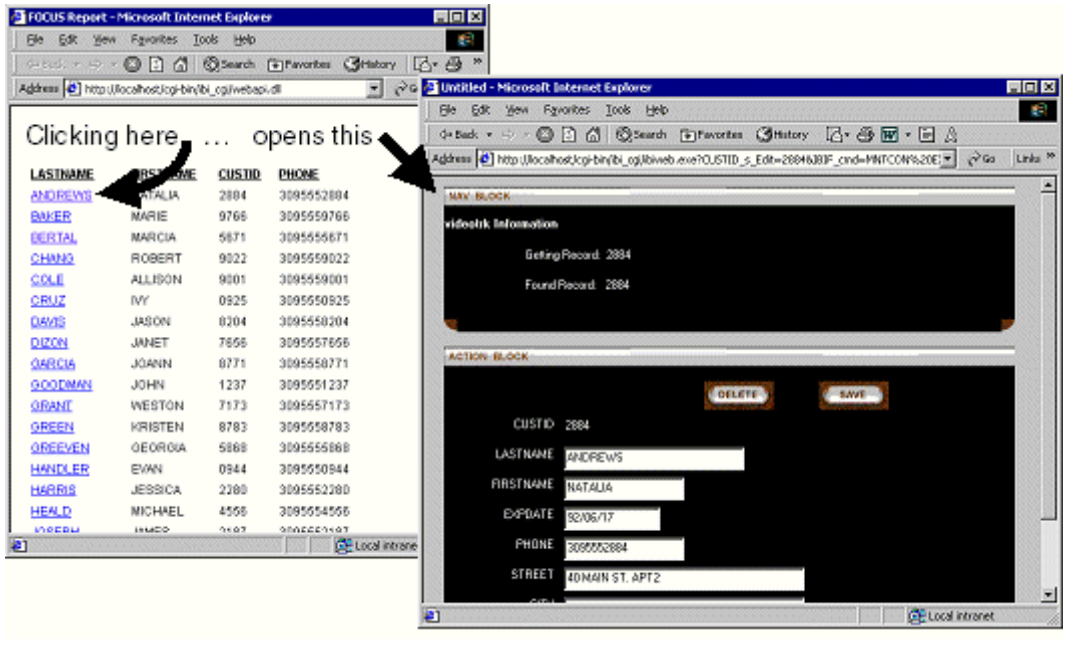
## Calling an Update Assist Project From a WebFOCUS Report Example

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 *Creating an Update Assist Project For the Videotrk Data Source* on page 20-15.
2. Create a report in the Report Painter that contains a simple list of the customers listed in the Videotrk data source. See *Creating a Report With the Videotrk Data Source* on page 20-18.

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 illustrated by the following:

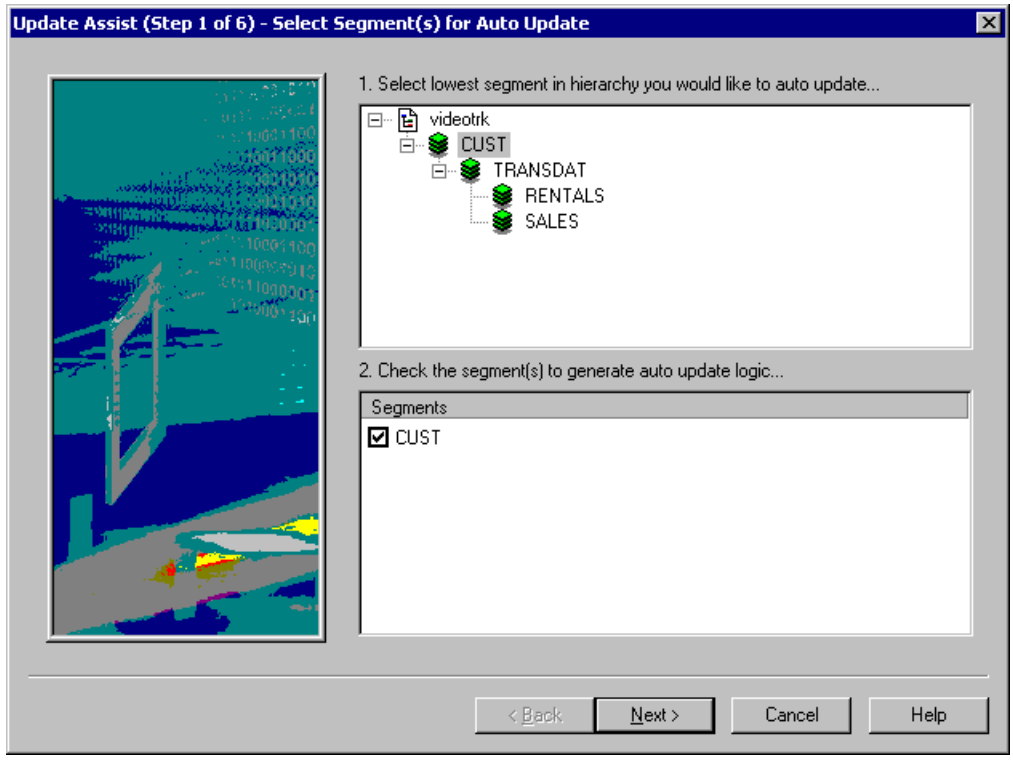


## *Example* Creating an Update Assist Project For the Videotrk Data Source

1. Create a project. For information on creating a project, see *How to Create a Project Directory* on page 3-5.
2. Add the IBISAMP directory to the viewable directories for the project.
3. Add the *Videotrk* data source to your project. See *How to Add a Master File to a Project* on page 3-13 for details on adding a data source to your project.
4. Right-click the Videotrk data source 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) - Select Segment(s) for Auto Update Window* on page 20-4.
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:

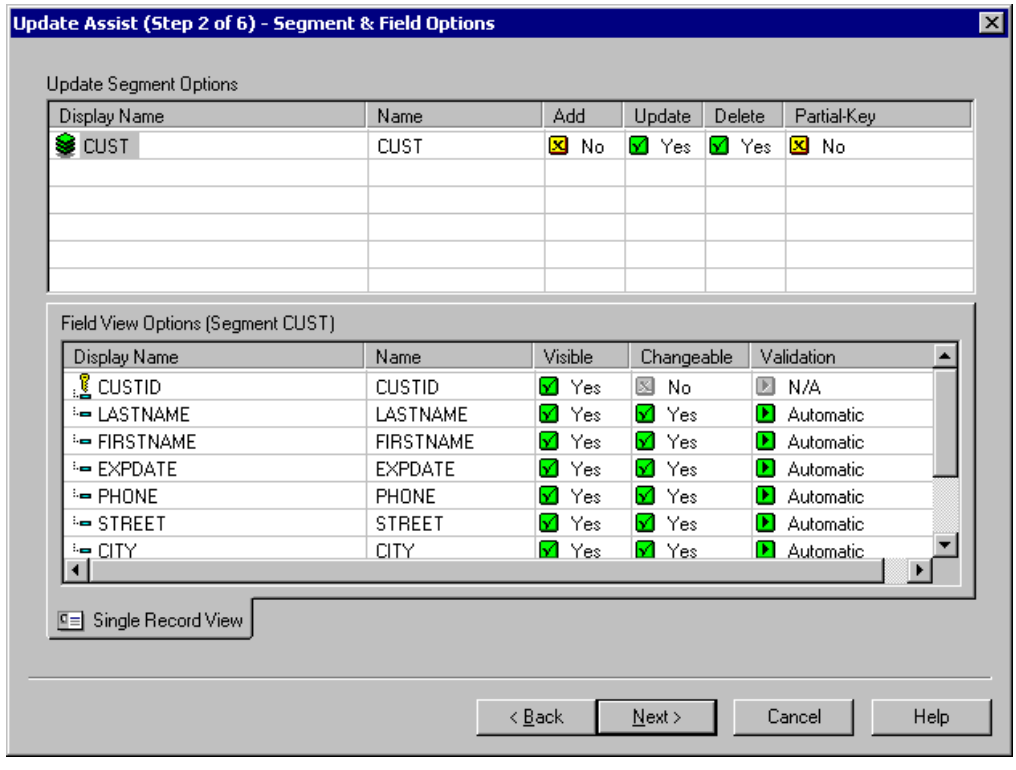


**7.** Click *Next* >.

The Update Assist (Step 2 or 6) - Segment & Field Options window opens. For details, see *Update Assist (Step 2 of 6) - Segment & Field Options Window* on page 20-5.

**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:




**10.** Click *Next* >.

The Update Assist (Step 3 of 6) - Navigation Options window opens. For details, see *Update Assist (Step 3 of 6) - Navigation Options Window* on page 20-9.

**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. For details, see *Update Assist (Step 5 of 6) - Output File Options Window* on page 20-10.

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. For details, see *Update Assist (Step 6 of 6) - Summary Window* on page 20-11.

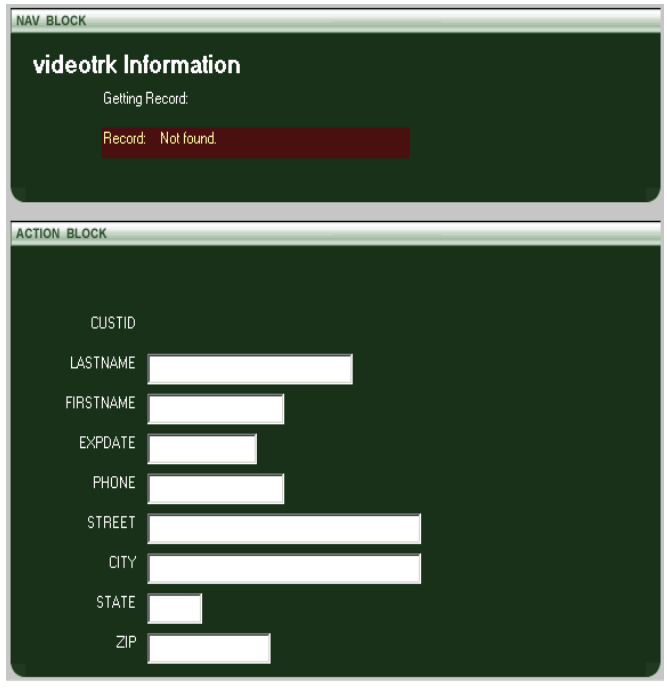
14. Review the summary of options you have chosen, and ensure *Automatically run this project when you press the 'Finish'* is selected.

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 a parameter containing a value for CUSTID (the key field).



### *Example* 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.

   **c.** Select *Videotrk*, and click *Open*.

   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.

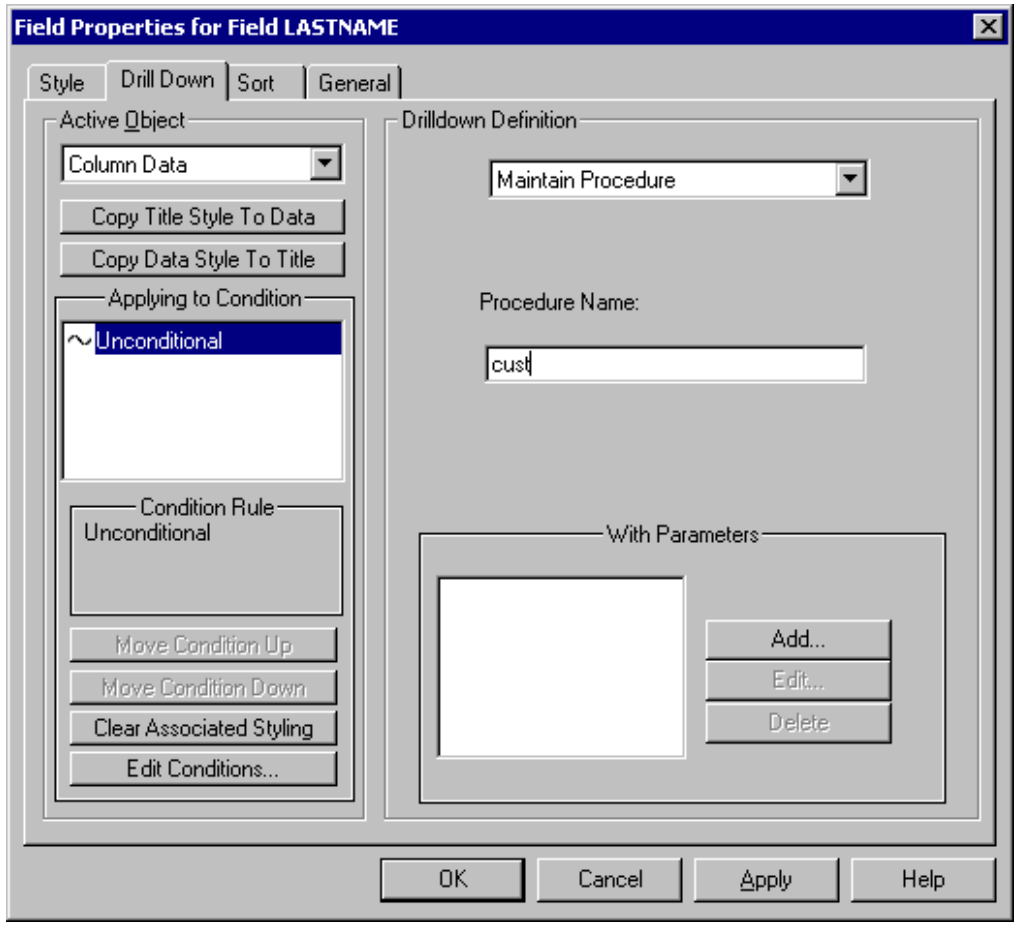
4. Click the *Drill Down* tab.

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 Drilldown 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:

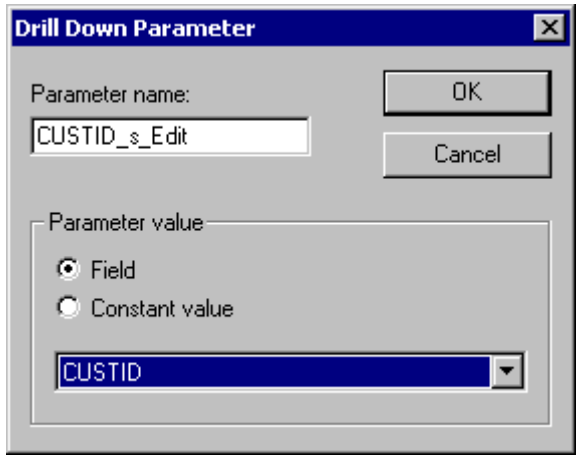
**a.** Click *Add…* in the With Parameters section.

The Drill Down Parameter dialog box opens.

**b.** Enter *CUSTID_s_Edit* in the Parameter name box.

**c.** Select *CUSTID* from the drop-down list in the Parameter value section.

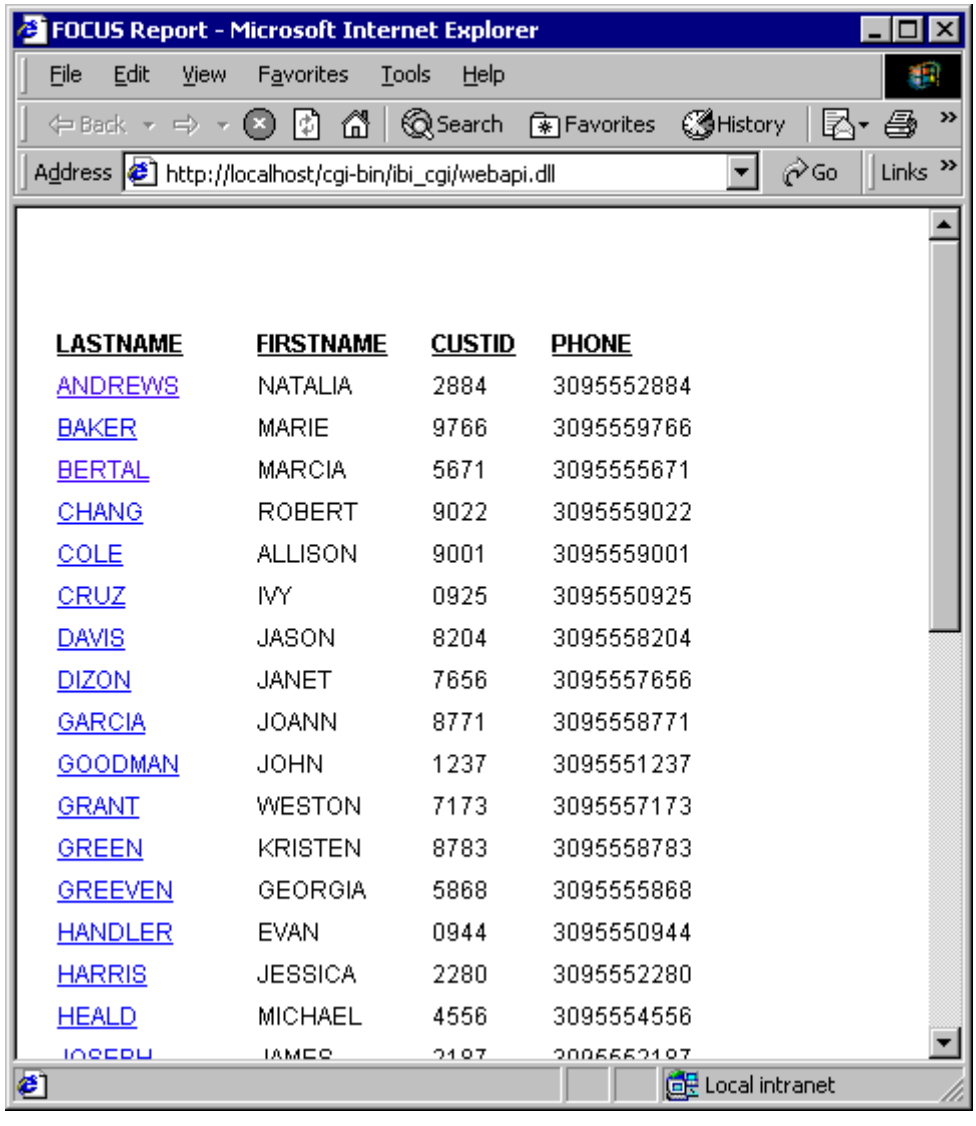
The window appears as follows:




**d.** Click *OK*.

**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 seen in the following:




| LASTNAME | FIRSTNAME | CUSTID | PHONE |
|---|---|---|---|
| ANDREWS | NATALIA | 2884 | 3095552884 |
| BAKER | MARIE | 9766 | 3095559766 |
| BERTAL | MARCIA | 5671 | 3095555671 |
| CHANG | ROBERT | 9022 | 3095559022 |
| COLE | ALLISON | 9001 | 3095559001 |
| CRUZ | IVY | 0925 | 3095550925 |
| DAVIS | JASON | 8204 | 3095558204 |
| DIZON | JANET | 7656 | 3095557656 |
| GARCIA | JOANN | 8771 | 3095558771 |
| GOODMAN | JOHN | 1237 | 3095551237 |
| GRANT | WESTON | 7173 | 3095557173 |
| GREEN | KRISTEN | 8783 | 3095558783 |
| GREEVEN | GEORGIA | 5868 | 3095555868 |
| HANDLER | EVAN | 0944 | 3095550944 |
| HARRIS | JESSICA | 2280 | 3095552280 |
| HEALD | MICHAEL | 4556 | 3095554556 |

Clicking any name on the report opens the Update Assist form with the information for that name already filled in.

# Index

## Symbols


%STRICTMATH parameter 15-11

&ECHO variable 10-69 to 10-70

&RETCODE variable 10-69 to 10-70

&STACK variable 10-69 to 10-70

-* command 10-71

.EVAL suffix 10-23

.EXIST suffix 10-23

.GFA control file 3-5
    deleting 3-27

.LENGTH suffix 10-23

.TYPE suffix 10-23

? && command 11-30

? COMBINE command 11-4

-? command 10-71

? DEFINE command 11-5 to 11-6

? FDT command 11-8 to 11-9

? FILE command 11-11

? FILEDEF command 11-13 to 11-14, 16-10, 16-39

? FUNCTION command 11-14

? HOLD command 11-14 to 11-15

? JOIN command 11-15 to 11-16

? n command 11-18

? NLS command 11-17

? PATH command 11-19

? RELEASE command 11-20

? REMOTE command 11-20 to 11-21

? SET command 11-21 to 11-23

? SET EUROFILE command 17-15

? SET GRAPH command 11-24 to 11-25

? SET NOT command 11-23 to 11-24

? STAT command 11-25 to 11-26

? STYLE command 11-26 to 11-27

? SU command 11-29

? TSO DDNAME command 16-30

? TSO DDNAME ddname command 16-31 to 16-33

? USE command 11-29, 14-20

?F command 11-7

?FF command 11-9 to 11-10


## Numerics


3D parameter 15-14, 15-20


## A


Access Files 2-4, 3-12, 16-12
    identifying 11-30, 11-32
    MVS 16-23
    UNIX 16-41
    WHENCE command 11-30, 11-32
    Windows NT/2000 16-12

ACCESSIBLE parameter 2-14, 15-18, 15-20

accessing FOCUS data sources 14-3, 14-6 to 14-7

ad hoc reports 5-9, 8-24
    enabling 8-24 to 8-25

adding environments 13-2

adding objects to HTML display 6-19

adding objects to HTML display pages 6-8
frames 6-14 to 6-15
graphs 6-9 to 6-10
group boxes 6-24
HTML templates 6-51
hyperlinks 6-23
images 6-19
lines 6-17
push buttons 6-27 to 6-28
reports 6-9 to 6-10
text 6-21
themes 6-51
WebFOCUS forms 6-30

adding objects to launch pages 6-8
group boxes 6-24
HTML templates 6-51
hyperlinks 6-23
images 6-19
lines 6-8, 6-17
push buttons 6-27 to 6-28
text 6-21
themes 6-51
WebFOCUS forms 6-30

adding themes 6-51
Microsoft FrontPage 6-51

ALL parameter 15-16, 15-20

allocating files in MVS 16-30
Access Files 16-23
data sources 16-24 to 16-25
displaying 16-30
external indexes 16-25
Master Files 16-22 to 16-23
WebFOCUS StyleSheets 16-25

ALLOWCVTERR parameter 15-16, 15-22

alternate file specifications 14-8 to 14-10
USE command 14-8 to 14-9
Use tool 14-8

APP PATH logic 2-12, 8-3 to 8-4

APPADMIN variable 2-12

application components 18-1
editing 18-1
viewing as text 18-7

application files 16-11
Access Files 16-12, 16-23, 16-41
data sources 16-13 to 16-14, 16-24, 16-42
external indexes 16-13, 16-25, 16-42
function libraries 16-15
Master Files 16-11, 16-22, 16-40
MVS 16-20, 16-22
procedures 16-12, 16-23, 16-41
profiles 16-15, 16-26, 16-43
UNIX 16-36, 16-40
Web pages 16-15
WebFOCUS StyleSheets 16-14, 16-25, 16-43
Windows NT/2000 16-2, 16-11

application logic 2-1 to 2-2

application partitioning 2-2 to 2-3

application processing 2-5

applications 2-1, 7-10
controlling 7-38
controlling memory 15-15
creating 2-9, 7-10, 7-35
debugging 2-9
deploying 2-6, 2-10
managing flow of control 10-1
optimizing 15-15
partitioning 2-2 to 2-3, 2-6, 2-10
processing 2-5
searching for 2-12
storing 2-12
update applications 20-2
WebFOCUS components 2-6

APPROOT variable 2-12

ASNAMES parameter 15-11, 15-16, 15-23

AUTODRILL parameter 15-18, 15-24

AUTOINDEX parameter 15-15, 15-25

AUTOPATH parameter 15-15, 15-25

AUTOSTRATEGY parameter 15-15, 15-26

AUTOTICK parameter 15-14, 15-26

## B

BARNUMB parameter 15-14, 15-26

BASEURL parameter 15-13, 15-27

BINS parameter 15-15, 15-27

bookmarks 18-12
    text editor 18-12

BOTTOMMARGIN parameter 15-18, 15-27

branching procedures 10-45 to 10-46
    conditional 10-46
    unconditional 10-45

BSTACK parameter 15-14, 15-18

BUSDAYS parameter 15-12, 15-16, 15-28

BYPANEL parameter 15-28

## C

cache 9-32
    clearing 9-32

CACHE parameter 15-15, 15-29

calculations 15-11
    controlling with SET parameters 15-11

CARTESIAN parameter 15-16, 15-30

Cascading Style Sheets (CSS) 9-33

case sensitivity 18-8, 18-10

case sensitivity in Command Console 19-4

CDN (Continental Decimal Notation) 17-16 to 17-17
    punctuating large numbers 17-16

CDN parameter 15-11, 15-16, 15-30

CENT-ZERO parameter 15-16, 15-31

CGI 8-7, 8-14
    calling from forms 8-11 to 8-12, 8-14
    calling from hyperlinks 8-7 to 8-8, 8-10

Check Box Properties window 6-39 to 6-40

check boxes 7-4
    Check Box Properties window 6-39 to 6-40
    supplying parameter values 6-39

clearing output in Command Console 19-5

-CLOSE command 10-21, 10-72

-CMS command 10-66

Command Console 19-1 to 19-5
    accessing 19-3
    case sensitivity in 19-4
    toolbar 19-3

Command Console display options 19-3

Command Console toolbar 19-3

Command Entry window 19-2

command statistics 11-25
    ? STAT 11-25

commands 10-1, 19-2
    dragging and dropping 19-4
    editing 19-4
    entering 19-4 to 19-5
    EXEC 10-58
    executing 19-1 to 19-3, 19-5
    executing remotely 19-3, 19-5
    in Dialogue Manager 10-1, 10-3
    operating system 10-66 to 10-67
    query commands 11-2
    stacked 10-4
    testing results 10-69 to 10-70

Comment component 4-17, 18-4

comments 10-8, 18-4
    adding 3-14, 3-21, 10-8
    Comment component 4-13
    creating 18-4

Component Connector toolbar 4-13, 4-18, 4-23
adding objects to procedures 4-13

components 4-13
deleting 3-12
renaming 3-12

Components toolbar 6-7

compound -IF tests 10-49

COMPUTE parameter 15-11, 15-15 to 15-16, 15-31

concatenating FOCUS data sources 14-14
USE command 14-14, 14-16 to 14-19
Use tool 14-14 to 14-19

concatenating variables 10-34
-SET command 10-34

conditional branching 10-46
compound -IF tests 10-49
-IF ... GOTO command 10-46 to 10-49
screening values 10-23

controlling calculations 15-11

controlling the environment 15-20

controls 5-7
adding to forms 5-7

converting currencies 17-2 to 17-3, 17-11, 17-13
currency data source 17-2
error messages 17-12

cookies 5-2, 5-5

copying commands in Command Console 19-4

creating HTML display pages 6-2
Report Layout Painter 6-2

creating launch pages 5-8
Microsoft FrontPage 2000 7-1
Report Layout Painter 6-2

creating main pages 7-38

CSS (Cascading Style Sheets) 9-33

CSSURL parameter 15-18, 15-31

currencies 17-2
converting 17-2 to 17-3

CURRENCY attribute 17-7, 17-9

currency conversions 17-2

currency data source 17-2
activating 17-9
creating 17-4 to 17-5, 17-7
CURRENCY attribute 17-7, 17-9
currency codes 17-4, 17-6 to 17-7
querying 17-15

currency symbol 17-17
extended currency symbol 17-17 to 17-18

CURRENCY[SYM] parameter 15-18, 15-32

Customer Support Service 1-vi
How to contact 1-vi
Information required 1-vi

customizing HTML pages 6-56

## D

data 15-11
processing 15-11
storing 15-11

data access logic 2-2

data sets 16-34
estimating size 16-35
testing for 16-34

data sources 16-13 to 16-14, 16-24 to 16-25
? FILE command 11-11
? USE command 11-29
accessing 15-19
identifying 11-30 to 11-31
MVS 16-25, 16-34
statistics 11-11
WHENCE command 11-30 to 11-31
Windows NT/2000 16-13

DATEDISPLAY parameter 15-12, 15-16, 15-32

DATEFNS parameter 15-12, 15-16, 15-32

DATEFORMAT parameter 15-11

dates 15-12
    displaying 15-12
    manipulating 15-12

DATETIME parameter 15-12, 15-16, 15-33

ddnames 16-5, 16-10
    clearing 16-39
    displaying 16-10, 16-39

ddnames (logical names) 11-13
    displaying 11-13 to 11-14

debugging procedures 11-2
    query commands 11-2

-DEFAULT command 10-12, 10-72

default pages 8-36

default variable values 10-12
    -DEFAULT command and 10-12

DEFCENT parameter 15-12, 15-16, 15-33

defined functions 11-14
    displaying 11-14

DEFINES parameter 15-11, 15-34

defining environment properties 13-4

defining files 16-1
    UNIX 16-38
    Windows NT/2000 16-5

deleting commands in Command Console 19-4

Deploy Wizard 2-6, 2-10, 8-2, 9-2 to 9-3
    deploying applications to the Web 9-4

deployed applications 2-6, 2-10, 12-2

deployed projects 12-2

deploying applications 2-11

deploying files 12-2

deployment environment tasks 15-13

deployment scenarios 3-14, 12-2
    selecting 3-14, 3-16

deployment steps 12-2

designing user interfaces 5-1
    coding 8-1
    Microsoft FrontPage 2000 7-1
    Report Layout Painter 6-2

Developer Studio 2-6, 2-9, 9-2

Developer Studio text editor 18-1

Developer Studio text editor (TED) 8-2

development environment 13-12
    setting 13-12

development environment tasks 15-13

Dialogue Manager 4-24, 8-1, 10-1 to 10-2
    calling procedures 10-53
    comments 10-8
    creating procedures 10-7
    debugging procedures 10-69
    looping procedures 10-50
    messages 10-68
    operating system commands 10-66 to 10-67
    passwords 10-67
    reading files 10-16
    stacked commands 10-4, 10-42
    testing for data sets 16-34
    variables 10-30
    Web pages and 10-58
    writing files 10-16

Dialogue Manager commands 10-1, 10-3
    -* command 10-71
    -? command 10-71
    -CLOSE 10-21, 10-72
    -CMS 10-66
    -DEFAULT 10-72
    -DOS 10-66, 10-72
    -EXIT 10-41 to 10-42, 10-73
    -GOTO 10-45, 10-73

Dialogue Manager commands (*continued*)
-HTMLFORM 10-58 to 10-60, 10-62, 10-74
-IF 10-75
-IF ... GOTO 10-46
-INCLUDE 10-53 to 10-56, 10-76
-label 10-76
-MVS 10-66, 10-77
operating system 10-66
-PASS 10-67, 10-77
-QUIT 10-41, 10-43, 10-77
-QUIT FOCUS 10-41, 10-43, 10-78
-READ 10-16 to 10-19, 10-79
-REPEAT 10-80
-RUN 10-41, 10-82
-SET 10-82
-TSO 10-66, 10-83
-TYPE 10-68, 10-83
-UNIX 10-66, 10-83
-VMS 10-66, 10-84
-WINNT 10-66, 10-84
-WRITE 10-16, 10-18 to 10-19, 10-85

Dialogue Manager procedures 10-4
processing commands and variables 10-4

Dialogue Manager variables 10-30 to 10-31, 16-32 to 16-33
evaluating 10-27 to 10-28
global 10-30 to 10-31
indexed 10-29
local 10-30 to 10-31
naming 10-30, 10-32
positional 10-14, 10-16
statistical 10-30
supplying dates 10-22
supplying values 10-9 to 10-10, 10-12 to 10-14, 10-21

directory paths 2-12
modifying 3-14, 3-18

display formats 9-1, 9-5

displaying dates 15-12

displaying fields 11-4
? COMBINE 11-4
? DEFINE command 11-5
?F 11-7

displaying reports 8-43
multiple reports on an existing HTML page 8-48
multiple reports on an inline HTML page 8-51
multiple reports on existing HTML page 8-57
on existing HTML page 8-43 to 8-45, 8-53 to 8-54
on inline HTML page 8-46

distributed processing 2-5

documentation roadmap 1-1 to 1-2

-DOS command 10-66, 10-72

dragging and dropping text in Command Console 19-4

drill-down reports 5-9, 9-23 to 9-24

Drop-Down List Properties window 6-34 to 6-35

drop-down lists 8-19, 8-21
Drop-Down List Properties window 6-34 to 6-35
dynamic 8-19
supplying parameter values 6-34

drop-down menus 7-5
adding to launch pages 7-21, 7-24

DTSTRICT parameter 15-11, 15-34

dynamic drop-down lists 8-19

dynamic reports 5-9

dynamically defining files 16-4
MVS 16-22
UNIX 16-37
Windows NT/2000 16-4

## E


EDAPATH logic 2-12, 8-3

edaserve.cfg file 2-12

EDASPROF.PRF file 2-12

editing application components 18-1

editing commands in Command Console 19-4

editing environment propterties 13-4

EMPTYREPORT parameter 15-16, 15-35

entering commands 19-4 to 19-5

Environment Properties dialog box 13-4

environments 13-1
- adding 13-2
- Managed Reporting/ReportCaster properties 13-9
- project development properties 13-11
- properties 13-4
- refreshing status 13-13
- removing 13-13
- setting development 13-12
- Web server properties 13-7
- WebFOCUS Client properties 13-8
- WebFOCUS Server properties 13-10

Environments List dialog box 13-2, 13-13

error messages 11-18
- ? n 11-18
- displaying explanations 11-18

ERROROUT parameter 15-16, 15-35

ESTRECORDS parameter 15-15 to 15-16, 15-35

euro currency 17-1 to 17-2
- converting 17-2 to 17-3, 17-11, 17-13

EUROFILE parameter 15-11, 15-36, 17-9
- error messages 17-9 to 17-10
- restrictions 17-9

evaluating variables 10-27 to 10-28

Excel 2000 9-4

EXEC command 10-14 to 10-15, 10-57 to 10-58
- calling procedures 10-57

Execute component 4-22 to 4-23, 10-14, 10-57

executing commands 19-1, 19-5

-EXIT command 10-41 to 10-42, 10-73

EXPIRE_REPORTS variable 9-41

EXTAGGR parameter 15-16, 15-36

external files 10-16
- closing 10-21
- reading from 10-16 to 10-18
- writing to 10-16

external indexes for FOCUS data sources 16-13
- MVS 16-25
- UNIX 16-42
- Windows NT/2000 16-13

EXTHOLD parameter 15-16, 15-37

extract files 16-16
- HOLD 16-16, 16-26, 16-44
- HOLDMAST 16-18, 16-28, 16-45
- HOLDSTAT 16-28
- MVS 16-20, 16-26
- SAVB 16-17, 16-27, 16-44
- SAVE 16-17, 16-27, 16-45
- temporary Master Files 16-18, 16-28, 16-45
- UNIX 16-36, 16-44
- Windows NT/2000 16-2, 16-16

EXTSORT parameter 15-16, 15-37


## F


Field Validation- List dialog box 20-8

Field Validation- Range dialog box 20-7

FIELDNAME parameter 15-11, 15-16, 15-38

fields 11-4
?FF command 11-9
displaying 11-4

file attributes 16-5
Dialogue Manager commands and 16-32
displaying 16-31
under MVS 16-30

file directory table 11-8
? FDT command 11-8

file locations 2-4

file specifications 14-2

file types 2-4
customizing 3-14, 3-19

FILE[NAME] parameter 15-16, 15-38

FILEDEF command 16-5, 16-9
clearing assignments 16-10
displaying assignments 16-10
UNIX 16-38
Windows NT/2000 16-5, 16-8

Filedef tool 16-5 to 16-6, 16-10
fields and options 16-7

files 2-4
defining paths 3-14
deleting 3-27
deploying 12-2
finding 2-12
partitioning 12-2

FILTER parameter 15-16, 15-39

filters 3-19

Financial Modeling Language (FML) 16-19, 16-46
FOCSML files and 16-19, 16-46

FIXRET[RIEVE] parameter 15-15, 15-39

FML (Financial Modeling Language) 16-19, 16-46
FOCSML files and 16-19, 16-46

FOC$HOLD files 16-44

FOC144 parameter 15-16, 15-40

FOCEXURL parameter 15-13, 15-40

FOCFIRSTPAGE parameter 15-18, 15-40

FOCPOST files 16-19, 16-29, 16-46
Windows NT/2000 16-19, 16-29, 16-46

FOCSML files 16-19, 16-46
FML and 16-19, 16-46
MVS 16-29
Windows NT/2000 16-19, 16-46

FOCSORT files 16-19, 16-29, 16-46
Windows NT/2000 16-19, 16-29, 16-46

FOCSTACK files 16-19, 16-29, 16-46
Dialogue Manager and 16-19, 16-29, 16-46
Windows NT/2000 16-19, 16-29, 16-46

FOCSTACK parameter 15-15, 15-41

FOCUS data sources 14-1, 16-13, 16-24 to 16-25, 16-42
accessing 14-1 to 14-3, 14-5 to 14-7
alternative file specifications 14-8 to 14-10
concatenating 14-14, 14-16 to 14-19
file specifications 14-2
identifying 14-5
MVS 16-24
new 14-11 to 14-12
protecting 14-13 to 14-14
querying 14-20
UNIX 16-42
USE command and 14-11 to 14-12
Use tool and 14-11
Windows NT/2000 16-13

footings 18-11
adding with text editor 18-11

formatting reports 9-33
Cascading Style Sheets and 9-33 to 9-34

forms 5-7
adding controls to 5-7
creating 8-11

FORMULTIPLE parameter 15-16, 15-41

Frame Properties window 6-14 to 6-16

frames
  adding to HTML display pages 6-14 to 6-15
  Frame Properties window 6-14 to 6-16

framesets 7-30

function libraries 16-15
  Windows NT/2000 16-15

## G

generic procedures 4-7
  creating 4-7

global profiles 16-15, 16-26, 16-43

global variables 10-30 to 10-31, 10-34
  naming 10-32

-GOTO command 10-45, 10-73

graph placeholders 6-12
  selecting icon 6-12

Graph Properties window 6-11

GRAPHEDIT parameter 15-14, 15-42

graphical tools 18-14

graphs 7-35
  adding to HTML display pages 6-9 to 6-10
  displaying 7-35, 15-14
  graph placeholder 6-12
  Graph Properties window 6-11
  Manage Layout dialog box 6-12 to 6-13
  modifying components 6-12 to 6-13
  processing 15-14

GRAPHSERVURL parameter 15-42

GRID parameter 15-14, 15-43

GRMERGE parameter 15-18, 15-43

Group Box Properties window 6-24 to 6-25

group boxes
  adding to HTML display pages 6-24
  adding to launch pages 6-24
  Group Box Properties window 6-24 to 6-25

group profiles 16-26, 16-43

GRTREND parameter 15-44

GRWIDTH parameter 15-18, 15-44

## H

handicapped accessibility 2-14

HAUTO parameter 15-14, 15-45

HAXIS parameter 15-14, 15-45

HCLASS parameter 15-14, 15-45

HDAY parameter 15-46

headings 18-11
  adding with text editor 18-11

HISTOGRAM parameter 15-14, 15-46

HMAX parameter 15-14, 15-46

HMIN parameter 15-14, 15-47

HOLD fields 11-14
  ? HOLD command 11-14

HOLD files 16-16, 16-26, 16-28
  MVS 16-26
  UNIX 16-44
  Windows NT/2000 16-16

HOLDATTR[S] parameter 15-16, 15-47

HOLDFORMAT parameter 15-11, 15-48

HOLDLIST parameter 15-11, 15-48

HOLDMAST files 16-18, 16-28
  MVS 16-28
  UNIX 16-45 to 16-46
  Windows NT/2000 16-18

HOLDMISS parameter 15-11, 15-49

HOLDSTAT files 16-28 to 16-29
MVS 16-28 to 16-29

HOLDSTAT parameter 15-11, 15-49

home links 7-40
adding to main pages 7-40

HSTACK parameter 15-14, 15-50

HTICK parameter 15-14, 15-50

HTML display pages 5-9
creating 6-2, 9-57
creating with Report Layout Painter 6-2
customizing 6-56
properties 6-45
Report Layout Painter 6-2
styling 6-51
templates 6-51
themes 6-51

HTML files 3-2, 5-1, 8-2, 8-42, 8-44 to 8-45, 8-53 to 8-54
display pages 5-2, 5-9, 6-2
displaying multiple reports 8-48, 8-51, 8-57
inline pages 8-46, 8-51

HTML Files folder 3-3 to 3-4

HTML forms 5-1
creating 9-59

HTML Forms folder 8-2

HTML pages 9-33
adding HTML templates 6-51
Cascading Style Sheets and 9-33 to 9-34
HTML Page Properties window 6-52
laying out objects 6-53
properties 6-52
Report Layout Painter 6-8

HTML templates
adding to HTML display pages 6-51
adding to launch pages 6-51

HTMLCSS parameter 15-50

-HTMLFORM command 8-42 to 8-43, 8-46, 10-58 to 10-60, 10-62, 10-74
embedding files 10-62

Hyperlink Properties dialog box 6-27

hyperlinks 7-2, 7-8, 7-36, 8-1, 8-7 to 8-8, 8-10, 9-12
adding 7-36, 7-39
adding to HTML display pages 6-23
adding to launch pages 6-23
adding to text 6-23
CGI and 8-7 to 8-8, 8-10
WebFOCUS CGI and 8-7 to 8-8, 8-10

## I

IBIWF_mframename variable 9-9

IBIWF_mprefix variable 9-9

IBIWF_mreports variable 9-9

identifying files 11-30
WHENCE command 11-30 to 11-32

identifying FOCUS data sources 14-2 to 14-3, 14-5 to 14-7, 14-11 to 14-12, 14-20

-IF ... GOTO command 10-46 to 10-49

-IF command 10-75

-IF tests 10-23 to 10-27

Image Properties window 6-19

images
adding to HTML display pages 6-19
adding to launch pages 6-19
Image Properties window 6-19

images pages
images 6-19

-INCLUDE command 10-53 to 10-56, 10-76
nesting procedures 10-56

Include component 4-22, 10-53 to 10-54

INDEX parameter 15-51

indexed variables 10-29 to 10-30
creating 10-30

Information Builders File System 3-6

ISAPI 8-14
calling WebFOCUS with 8-14 to 8-15

## J

JavaScript 7-31, 8-1, 8-28 to 8-29, 9-23
adding to launch pages 7-31
functions 9-23
running reports 7-28

JavaScript functions 9-23

join structures 11-15
displaying 11-15

JOINOPT parameter 15-16, 15-51

JSURL parameter 15-16

## K

KEEPDEFINES parameter 15-11, 15-16, 15-52

## L

-label command 10-76

LANG parameter 17-19

LANG[UAGE] parameter 15-13, 15-52

launch pages 5-2, 5-6, 6-2, 7-1, 7-9, 9-1
adding WebFOCUS forms to 5-7, 7-3
creating 5-8, 7-2, 7-12, 7-30, 7-35
creating with Microsoft FrontPage 2000 7-1
creating with Report Layout Painter 6-2
customizing 5-8, 6-56, 7-2 to 7-3, 7-27, 9-1
frameset 5-8
properties 6-45
single-page 5-8
styling 6-51
templates 6-51
themes 6-51

Layout toolbar 6-55

LEADZERO parameter 15-12, 15-16

LEFTMARGIN parameter 15-18, 15-53

Line Properties window 6-18

lines 6-17
adding to HTML display pages 6-17
adding to launch pages 6-17
Line Properties window 6-18

LINES parameter 9-30, 15-18, 15-54

LINKMODE parameter 15-13, 15-54

List Box Properties window 6-36 to 6-37

list boxes
List Box Properties window 6-36 to 6-37
supplying parameter values 6-36

Local Deploy default value 3-16

local development 3-2

local variables 10-30 to 10-31, 10-33
naming 10-32

localhost 3-2

LOCATION attribute 14-8

logical names 16-5
assigning under UNIX 16-38
clearing 16-39
FILEDEF command and 16-5, 16-8, 16-38
Filedef tool 16-6
Windows NT/2000 16-5

logon pages 5-2
modifying 5-5

LOOKGRAPH parameter 15-14, 15-55

looped procedures 10-50
-REPEAT command 10-50 to 10-51
-SET command 10-50, 10-52

## M

main pages 5-2, 7-38
adding home links 7-40
adding hyperlinks 7-39

Maintain Files folder 3-3

Manage Layout dialog box 6-12 to 6-13

Managed Reporting/ReportCaster properties 13-9

manipulating dates 15-12

Master File Editor 3-12

Master Files 2-3 to 2-4, 3-2, 3-12, 14-8
adding 3-13
deleting 3-14
identifying 11-30 to 11-31
locating 16-11
MVS 16-22 to 16-23
naming 16-11
UNIX 16-40
WHENCE command 11-30 to 11-31
Windows NT/2000 16-11

Master Files folder 3-3 to 3-4

MAXLRECL parameter 15-11, 15-56

memory 15-15
controlling 15-15

menu options 9-42
customizing 9-42, 9-44, 9-46, 9-48, 9-53 to 9-54

MESSAGE parameter 15-13, 15-56

messages 10-68
sending 10-68 to 10-69

metadata 15-11
processing 15-11
storing 15-11

Microsoft FrontPage 2000 2-12, 5-1, 5-8, 7-1, 7-27, 8-2, 8-7
creating launch pages with 7-1
creating main pages 7-38

Microsoft FrontPage 2000 (*continued*)
HTML editor 7-28

Microsoft Internet Explorer (IE) 9-2

MIME (Multipurpose Internet Mail Extensions) 9-2

moving text in Command Console 19-4

MULTIPATH parameter 15-16, 15-57

multiple reports 9-8 to 9-9

multi-select list boxes 8-19, 8-21

MVS 16-20, 16-26, 16-28 to 16-29
Access Files 16-23
application files 16-20, 16-22
data sets 16-34
data sources 16-24 to 16-25
dynamically defining files 16-22
estimating data set size 16-35
external indexes 16-25
extract files 16-20, 16-26
Master Files 16-22
procedures 16-23
profiles 16-26
referencing files 16-20
temporary Master Files 16-28
WebFOCUS files 16-20
WebFOCUS StyleSheets 16-25
work files 16-20

-MVS command 10-66, 10-77

## N

navigating procedures 10-44
branching 10-45 to 10-46

New Filter dialog box 3-26

NEW parameter 14-11 to 14-12

New Procedure dialog box 4-9

NLS (National Language Support) 11-17
? NLS 11-17

NODATA parameter 15-16, 15-57

## O

object relationships 6-53 to 6-54
breaking 6-55
Layout toolbar 6-55
setting distance 6-53 to 6-54
setting size 6-53 to 6-54

objects 3-14
specifying starting object 3-14
specifying starting objects 3-17

OFFLINE files 16-19
UNIX 16-46 to 16-47
Windows NT/2000 16-19

OFFLINE-FMT parameter 15-13, 15-58

OFFLINEFMT parameter 15-18

on-demand paging 9-29 to 9-30, 9-32 to 9-33

one-line text boxes 7-4

ONLINE-FMT parameter 15-58

ONLINEFMT parameter 15-18

opening graphical tools 18-14

operating system commands 10-66 to 10-67
-DOS command 10-72
-MVS command 10-77
-TSO command 10-83
-UNIX command 10-83
-VMS command 10-84
-WINNT command 10-84

ORIENTATION parameter 15-18

OS/390 operating system 14-11

Other component 4-16, 18-1, 18-3
text editor 18-3

Output window 19-2
clearing 19-5

## P

PACKPOS parameter 15-11

page properties 6-56
Settings dialog box 6-56 to 6-57

PAGE[-NUM] parameter 15-18

PAGESIZE parameter 15-13, 15-18

parameter settings 15-1
displaying 11-23

parameter values 6-46, 6-50
check boxes 6-39
drop-down lists and 6-34
dynamic list 6-49 to 6-50
list boxes 6-36
radio buttons 6-42
static list 6-47, 6-50
text boxes and 6-32

parameters 14-11
NEW 14-11 to 14-12
Parameters window 6-45 to 6-47, 6-49
properties 6-46
READ 14-13 to 14-14

Parameters window 6-45 to 6-47, 6-49

partitioning files 12-2

-PASS command 10-67, 10-77

PASS parameter 15-19

passwords 3-27, 10-67

pasting commands in Command Console 19-4

PCOMMA parameter 15-11, 15-59

PDF display format 9-1

positional variables 10-14, 10-16

presentation logic 2-2

PRINT parameter 15-13, 15-18, 15-60

procedure components 4-14
checking 4-20
copying 4-18, 4-22
creating 4-13 to 4-14, 4-18
debugging 4-18
dragging and dropping 4-22
editing 4-18 to 4-19
running 4-18 to 4-19, 18-13
types 4-13
viewing as text 18-7

procedure creation tools 4-2, 4-4

procedures 3-2, 4-1, 7-10, 9-15, 9-17, 10-7, 11-1, 15-19, 16-5 to 16-6, 16-12
activating OLAP 4-25
adding to applications 4-10
adding to project path 4-11
adding to projects 4-12
branching 10-45 to 10-46
calling 4-22, 10-53 to 10-58
canceling 4-29, 10-43 to 10-44, 10-49
comments in 10-8
components 4-13
controlling 10-1
copying 4-13
creating 4-1, 7-10, 7-29, 9-15, 9-17, 9-20, 9-22, 9-55, 9-65, 10-7, 18-5
creating with Text Editor 4-8
debugging 4-25, 4-27, 10-69, 11-1 to 11-2
decrypting 4-25 to 4-26
deleting 4-12
Dialogue Manager 10-1
editing 4-25 to 4-26, 18-2
embedding 4-23
encrypting 4-25 to 4-26
executing 4-23
exiting 10-42
HTML Web pages and 10-58
identifying 11-30 to 11-31
in Dialogue Manager 10-7
including 4-23
looping 10-50

procedures (*continued*)
navigating 10-44 to 10-46
nesting 10-56 to 10-57
processing 10-6
publishing 4-25
running 10-4, 10-6, 10-41, 10-43, 18-13
running locally 4-27
running remotely 4-27 to 4-28, 7-29
scheduling 4-25 to 4-26
screening values in 10-23
selecting tool for creating 4-3
testing 11-1
text components 18-5
tools for creating 4-2, 4-4
variable values 10-9, 10-23 to 10-27
WHENCE command 11-30 to 11-31

Procedures folder 3-3 to 3-4

procedures under MVS 16-23
allocating 16-23

procedures under UNIX 16-41
naming 16-41

procedures under Windows NT/2000 16-12
locating 16-12
naming 16-12

processing data 15-11

profiles under MVS 16-26

profiles under UNIX 16-43

profiles under Windows NT/2000 16-15

program logic 2-2

project development properties 13-11

project directories 3-3

Project Properties dialog box 3-14 to 3-15

Project Wizard 3-3

projects 3-3
applying read-only security 3-17
creating directories 3-5
decrypting 3-27
deleting 3-27
deploying files 12-2
directory paths 3-14
encrypting 3-27
file types 3-4
organizing 3-9
partitioning files 12-2
running 3-14
running on remote server 3-17
securing 3-27
tracking code 3-27
viewing properties 3-14 to 3-15

Properties window 6-45

protecting FOCUS data sources 14-13 to 14-14

PSPAGESETUP parameter 15-60

Publish Utility 5-1, 5-8, 7-1

Publish utility 8-7

punctuating large numbers 17-16 to 17-17

Push Button Properties window 6-27 to 6-29

push buttons 7-7
adding to HTML display pages 6-27 to 6-28
adding to launch pages 6-27 to 6-28
Push Button Properties window 6-27 to 6-29

## Q

QUALCHAR parameter 15-11, 15-16, 15-61

QUALTITLES parameter 15-18, 15-61

queries 19-1
executing 19-3

query commands 11-2, 14-20
? && 11-30
? COMBINE 11-4
? DEFINE 11-5 to 11-6

query commands (*continued*)
? FDT 11-8 to 11-9
? FILE 11-11
? FILEDEF 11-13, 16-10, 16-39
? FUNCTION 11-14
? HOLD 11-14 to 11-15
? JOIN 11-15 to 11-16
? n 11-18
? NLS 11-17
? PATH 11-19
? RELEASE 11-20
? REMOTE 11-20 to 11-21
? SET 11-21 to 11-23
? SET EUROFILE 17-15
? SET GRAPH 11-24 to 11-25
? SET NOT 11-23 to 11-24
? STAT 11-25 to 11-26
? STYLE 11-26 to 11-27
? SU 11-29
? USE 11-29, 14-20
?F 11-7
?FF 11-9 to 11-10

-QUIT command 10-41, 10-43 to 10-44, 10-77

-QUIT FOCUS command 10-41, 10-43, 10-78

## R

radio buttons 7-4
adding to launch pages 7-17
Radio Buttons Properties window 6-42 to 6-43
supplying parameter values 6-42

Radio Buttons Properties window 6-42 to 6-43

-READ command 10-16 to 10-19, 10-79

READ parameter 14-13 to 14-14

read-only security 3-14, 3-17

RECAP-COUNT parameter 15-18, 15-62

RECORDLIMIT parameter 15-15, 15-62

registered file types 3-20

release number 11-20

remote execution 19-3, 19-5

removing environments 13-13

-REPEAT command 10-50 to 10-51, 10-80

report formatting 9-33
    Cascading Style Sheets and 9-33 to 9-34

Report Layout Painter 5-1, 6-1 to 6-2
    accessing 6-3
    adding frames 6-14 to 6-15
    adding graphs 6-9 to 6-10
    adding group boxes 6-24
    adding HTML templates 6-51
    adding hyperlinks 6-23
    adding images 6-19
    adding lines 6-17
    adding push buttons 6-27 to 6-28
    adding reports 6-9 to 6-10
    adding text 6-21
    adding themes 6-51
    creating launch pages 6-2
    HTML pages 6-8
    launch pages 6-2
    page properties 6-56
    Properties window 6-45
    toolbars 6-4
    windows 6-4

Report Painter 10-21

report placeholders
    selecting icon 6-12

report procedures
    creating 4-5

Report Properties window 6-11

report types 5-9
    ad hoc 5-9, 8-24
    drill-down 5-9, 9-23 to 9-24
    dynamic 5-9
    static 5-9

ReportCaster 4-26
    scheduling procedures 4-26

reporting application logic 2-2

reporting applications 3-1 to 3-2
    creating directories 3-5
    decrypting 3-27
    deleting 3-27
    encrypting 3-27
    file types 3-4
    HTML files 3-2
    Master Files 3-2
    procedures 3-2
    securing 3-27
    viewing properties 3-15

reports 7-35, 8-36, 9-1 to 9-2, 15-16
    adding to HTML display pages 6-9 to 6-10
    calling Update Assist 20-12, 20-14 to 20-15, 20-18
    customizing 8-42
    displaying 7-35, 8-36 to 8-37, 8-39 to 8-40, 8-42 to 8-43, 8-45 to 8-46, 9-1 to 9-4, 15-18
    displaying in multiple formats 9-1, 9-8 to 9-9, 9-12 to 9-13, 9-15
    editing as text 18-2
    formatting 9-5, 9-33
    Manage Layout dialog box 6-12 to 6-13
    modifying components 6-12 to 6-13
    on-demand paging 9-29 to 9-30
    report placeholder 6-12
    Report Properties window 6-11
    running 7-28, 19-5
    viewing 9-41

repositioning objects 6-54

resizing objects 6-54

RIGHTMARGIN parameter 15-18, 15-63

RPAGESET parameter 15-18, 15-64

-RUN command 10-41, 10-82

running procedures 4-27
locally 4-27
remotely 4-28, 7-29

running report requests 19-5

## S

SAVB files 16-17
creating 16-44
MVS 16-27
UNIX 16-44
Windows NT/2000 16-17

SAVE files 16-17
creating 16-45
MVS 16-27
UNIX 16-45
Windows NT/2000 16-17

SAVEDMASTERS parameter 15-11, 15-64

SAVEMATRIX parameter 15-16, 15-65

search logic 2-12

search path logic 8-3 to 8-4
choosing 8-3

search paths 2-12, 8-3 to 8-4
displaying 11-19

security 3-27, 5-2, 15-19

self-service reporting applications 2-1, 3-1

sequential data sources 16-14, 16-42
UNIX 16-42
Windows NT/2000 16-14

servlets 8-16
support 8-16

SESSION project 3-3

-SET command 10-13, 10-34 to 10-35, 10-50, 10-52, 10-82

SET command 9-30, 15-1 to 15-3, 15-5 to 15-9, 17-19
parameters 15-10
precedence 15-3

SET parameter types 15-10

SET parameters 9-30, 15-1, 15-3, 15-5 to 15-9, 15-20
3D 15-20
ACCESSIBLE 2-14, 15-20
ALL 15-20
ALLOWCVTERR 15-22
ASNAMES 15-23
AUTODRILL 15-24
AUTOINDEX 15-25
AUTOPATH 15-25
AUTOSTRATEGY 15-26
AUTOTICK 15-26
BARNUMB 15-26
BASEURL 15-27
BINS 15-27
BOTTOMMARGIN 15-27
BUSDAYS 15-28
BYPANEL 15-28
CACHE 15-29
CARTESIAN 15-30
CDN 15-30
CENT-ZERO 15-31
COMPUTE 15-31
CSSURL 15-31
CURRENCY[SYM] 15-32
DATEDISPLAY 15-32
DATEFNS 15-32
DATETIME 15-33
DEFCENT 15-33
DEFINES 15-34
displaying settings 11-21
DTSTRICT 15-34
EMPTYREPORT 15-35
ERROROUT 15-35
ESTRECORDS 15-35
EUROFILE 15-36, 17-9
EXTAGGR 15-36

SET parameters (*continued*)
EXTHOLD 15-37
EXTSORT 15-37
FIELDNAME 15-38
FILE[NAME] 15-38
FILTER 15-39
FIXRET[RIEVE] 15-39
FOC144 15-40
FOCEXURL 15-40
FOCFIRSTPAGE 15-40
FOCSTACK 15-41
FORMULTIPLE 15-41
GRAPHEDIT 15-42
GRAPHSERVURL 15-42
GRID 15-43
GRMERGE 15-43
GRTREND 15-44
GRWIDTH 15-44
HAUTO 15-45
HAXIS 15-45
HCLASS 15-45
HDAY 15-46
HISTOGRAM 15-46
HMAX 15-46
HMIN 15-47
HOLDATTR[S] 15-47
HOLDFORMAT 15-48
HOLDLIST 15-48
HOLDMISS 15-49
HOLDSTAT 15-49
HSTACK 15-50
HTICK 15-50
HTMLCSS 15-50
INDEX 15-51
JOINOPT 15-51
KEEPDEFINES 15-52
LANG parameter 17-19
LANG[UAGE] 15-52
LEFTMARGIN 15-53
LINES 9-30, 15-54
LINKMODE 15-54

SET parameters (*continued*)
LOOKGRAPH 15-55
MAXLRECL 15-56
MESSAGE 15-56
MULTIPATH 15-57
NODATA 15-57
OFFLINE-FMT 15-58
ONLINE-FMT 15-58
PCOMMA 15-59
PRINT 15-60
PSPAGESETUP 15-60
QUALCHAR 15-61
QUALTITLES 15-61
RECAP-COUNT 15-62
RECORDLIMIT 15-62
RIGHTMARGIN 15-63
RPAGESET 15-64
SAVEDMASTERS 15-64
SAVEMATRIX 15-65
SHIFT 15-65
SPACES 15-66
SQLTOPTIF 15-66
SQUEEZE 15-67
STRICTMATH 15-67
STYLE[SHEET] 15-69
STYLEMODE 15-67
SUMPREFIX 15-70
TARGETFRAME 15-70
TEMP 15-70
TEMPDIR 15-71
TEMPERASE 15-71
TESTDATE 15-71
TITLE 15-72
TOPMARGIN 15-72
UNITS 15-72
USER 15-73
VAUTO 15-73
VAXIS 15-73
VCLASS 15-74
VGRID 15-74
VMAX 15-74

SET parameters 9 (*continued*)
VMIN 15-75
VTICK 15-75
VZERO 15-75
WEBVIEWER 15-76
WEBVIEWHOME 9-31
WEEFIRST 15-76
XRETRIEVAL 15-77
YRTHRESH 15-77

Set tool 15-3, 15-5
adding parameters 15-3
changing settings 15-3

setting parameters 15-3, 15-5 to 15-9
setting multiple parameters 15-7

setting the development environment 13-12

Settings dialog box 6-12, 6-56 to 6-57

SHIFT parameter 15-13, 15-65

sink machines 11-29
? SU command 11-29

source file management 3-27

SPACES parameter 15-18, 15-66

SQLTOPTTF parameter 15-15, 15-66

SQUEEZE parameter 15-18, 15-67

stacked commands 10-4, 10-48

Standard toolbar 6-5

starting objects 3-14
selecting 3-14, 3-17

static reports 5-9

statistical variables 10-30, 10-39 to 10-40
testing 10-47

steps for deployment 12-2

storing data 15-11

STRICTMATH parameter 15-67

style sheets 9-33
Cascading Style Sheets 9-33
WebFOCUS StyleSheets 2-3

STYLE[SHEET] parameter 15-18, 15-69

STYLEMODE parameter 15-18, 15-67

StyleSheets 11-26
? STYLE 11-26
identifying 11-30, 11-32
WHENCE command 11-30, 11-32

subfolders 3-11

sub-servers 2-6

SUMPREFIX parameter 15-16, 15-70

Synonym Wizard 3-12, 7-29

syntax coloring 18-8 to 18-10

system variables 10-30, 10-36, 10-38 to 10-39, 10-69
testing 10-47

## T

TARGETFRAME parameter 15-18, 15-70

TEMP parameter 15-15, 15-70

TEMPDIR parameter 15-15, 15-71

TEMPERASE parameter 15-11, 15-71

templates 8-45

temporary files 2-4

temporary Master Files 16-18, 16-28, 16-45
MVS 16-28
Windows NT/2000 16-18

TESTDATE parameter 15-12, 15-71

text
adding to HTML display pages 6-21
adding to launch pages 6-21
finding 18-8
replacing 18-8

text (*continued*)
supplying parameter values 6-32
Text Box Properties window 6-32 to 6-33
Text Properties window 6-21

text areas 8-24

Text Box Properties window 6-33

text colors 18-8
changing 18-8 to 18-10

text components 18-5
creating 18-5

Text Editor 4-8

text editor 18-1 to 18-2
adding footings 18-11
adding headings 18-11
case sensitivity 18-8, 18-10
changing default settings 18-8 to 18-10
components 18-2
creating procedures 18-5
creating text files 18-5
opening files 18-7
opening graphical tools 18-14
Other component 18-3

text editor window 18-8

text files 18-5
creating 18-5

Text Properties window 6-21

themes 6-51
adding to HTML display pages 6-51
adding to launch pages 6-51

three-tier application logic 2-2

TITLE parameter 15-16, 15-72

toolbar in Command Console 19-3

toolbars 6-55
Components toolbar 6-7
Layout toolbar 6-55
Standard toolbar 6-5

TOPMARGIN parameter 15-18, 15-72

tracking project code 3-27

-TSO command 10-66, 10-83

TSO system variables 16-32

-TYPE command 10-68 to 10-69, 10-83

## U

UNC (Universal Naming Convention) 16-9

unconditional branching 10-45
-GOTO command 10-45

UNITS parameter 15-18 to 15-19, 15-72

Universal Naming Convention (UNC) 16-9

UNIX 16-36
Access Files 16-41
application files 16-36, 16-40
data sources 16-42
defining files 16-37 to 16-38
external indexes 16-42
extract files 16-36, 16-44
locating files in 16-36
logical names 16-38
Master Files 16-40
naming files in 16-36
profiles 16-43
WebFOCUS files 16-36
WebFOCUS StyleSheets 16-43
work files 16-36

-UNIX command 10-66, 10-83

UNIX procedures 16-41

update applications 20-1
Update Assist 20-2, 20-4, 20-6, 20-9 to 20-11

Update Assist 20-1 to 20-2, 20-4, 20-6, 20-9 to 20-12, 20-14 to 20-15, 20-18
calling from reports 20-12, 20-14 to 20-15, 20-18

Update Assist windows 20-4, 20-6, 20-9 to 20-11
Field Validation- List dialog box 20-8
Field Validation- Range dialog box 20-7

URL servlet calls 8-19
WebFOCUS Servlet and 8-19

USE command 14-1 to 14-2, 14-6
accessing FOCUS data sources 14-2
alternate file specifications 14-8 to 14-9
concatenating data sources 14-14, 14-16
new FOCUS data sources and 14-11
NEW parameter and 14-11 to 14-12
protecting FOCUS data sources 14-13 to 14-14
READ parameter and 14-13 to 14-14

USE dialog box 14-3 to 14-4

USE directory 14-2

USE options 14-20
clearing 14-20
displaying 14-20

USE query 14-20

Use tool 14-2 to 14-4, 14-20
accessing FOCUS data sources 14-2
alternate file specifications 14-8
concatenating data sources 14-14 to 14-15
identifying new FOCUS data sources 14-11
protecting FOCUS data sources 14-13

user interfaces 5-1 to 5-2, 8-1
coding 8-1 to 8-2
designing 5-1
templates 8-1

USER parameter 15-19, 15-73

user profiles 16-15, 16-26, 16-43

## V

variable values 10-35
? && 11-30
dates 10-22
-DEFAULT command and 10-9, 10-12

variable values (*continued*)
default values 10-12
displaying 10-35
querying 11-30
-READ command 10-9
Report Painter and 10-21
-SET command 10-9
supplying 10-9 to 10-10, 10-12 to 10-15, 10-21, 15-7 to 15-8

variables 10-9
concatenating 10-34
supplying values 15-7 to 15-8

VAUTO parameter 15-14, 15-73

VAXIS parameter 15-14, 15-73

VCLASS parameter 15-14, 15-74

VGRID parameter 15-14, 15-74

viewing output 19-1

virtual fields 11-5
? DEFINE command 11-5
displaying 11-5

virtual folders 3-3, 3-9
adding 3-9, 3-11

VMAX parameter 15-14, 15-74

VMIN parameter 15-14, 15-75

-VMS command 10-66, 10-84

VTICK parameter 15-14, 15-75

VZERO parameter 15-14, 15-75

## W

Web browsers 2-6, 5-5
cookies 5-5

Web pages 2-3 to 2-4, 8-28, 8-36, 8-42, 10-58, 10-62, 16-15
customizing 8-42
default 8-36 to 8-37, 8-39 to 8-40

Web pages (*continued*)
embedding files in 10-62
-HTMLFORM command 10-58 to 10-60
JavaScript in 8-28 to 8-29
referring from a procedure 10-59
variables and 10-62 to 10-63, 10-65 to 10-66

Web server properties 13-7

Web servers 2-3, 2-6, 8-16

WebFOCUS CGI 8-7, 8-14
calling from forms 8-11 to 8-12, 8-14
calling from hyperlinks 8-7 to 8-8, 8-10

WebFOCUS Client 2-6, 5-8, 8-7

WebFOCUS Client properties 13-8

WebFOCUS Client variables 9-8 to 9-9, 9-15
IBIWF_mframename 9-9
IBIWF_mprefix 9-9
IBIWF_mreports 9-9

WebFOCUS code 18-2
editing 18-2

WebFOCUS environment 12-2

WebFOCUS environment properties 13-4

WebFOCUS Environment Properties dialog box 13-4

WebFOCUS environments 13-1

WebFOCUS files 16-1 to 16-2, 16-20, 16-36
application files 16-11, 16-22, 16-40
dynamically defining 16-4, 16-22, 16-37
extract files 16-16, 16-26, 16-44
MVS 16-20, 16-30
referencing 16-2, 16-20
types 16-2, 16-20, 16-36
UNIX 16-36, 16-38
Windows NT/2000 16-2, 16-5
work files 16-19, 16-29, 16-46

WebFOCUS forms 5-7, 7-2 to 7-3, 7-30
adding to HTML display pages 6-30
adding to launch pages 6-30, 7-3, 7-14
check boxes 6-39
drop-down lists 6-34
editing properties 7-3
list boxes 6-36
properties 6-45
Properties window 6-45
radio buttons 6-42
suppying parameter values 6-46
text boxes 6-32
types 6-31

WebFOCUS Reporting Server 2-3, 2-6, 5-2

WebFOCUS Server properties 13-10

WebFOCUS Servlet 8-14, 8-16
calling 8-19
configuring 8-16 to 8-17
requirements 8-16
URL servlet calls 8-19

WebFOCUS StyleSheets 2-3, 9-1, 16-14, 16-25
MVS 16-25
UNIX 16-43
Windows NT/2000 16-14

WebFOCUS Viewer 9-29 to 9-30
closing 9-32
home pages 9-31 to 9-32
navigating 9-33
opening 9-31

WEBVIEWER parameter 9-30, 15-13, 15-76

WEBVIEWHOME parameter 9-31 to 9-32

WEBVIEWTARG parameter 9-31

WEEKFIRST parameter 15-11, 15-76

WFServlet.jar file 8-16 to 8-17

WHENCE command 11-30 to 11-32

Windows NT/2000 16-2
Access Files 16-12
application files 16-2, 16-11
data sources 16-13 to 16-14
defining files 16-5
dynamically defining files 16-4
external indexes 16-13
extract files 16-2, 16-16
FILEDEF command 16-5, 16-8
Filedef tool 16-5 to 16-6
function libraries 16-15
HOLD files 16-16
HOLDMAST files 16-18
locating files in 16-2
logical names 16-5
Master Files 16-11
naming files in 16-2
OFFLINE files 16-19
procedures 16-12
profiles 16-15
referencing files 16-2
SAVB files 16-17
SAVE files 16-17
temporary Master Files 16-18
Web pages 16-15
WebFOCUS files 16-2
WebFOCUS StyleSheets 16-14
work files 16-2, 16-19, 16-29, 16-46

-WINNT command 10-66, 10-84

work files 16-19, 16-29, 16-46
FOCPOST 16-19, 16-29, 16-46
FOCSML 16-19, 16-29, 16-46
FOCSORT 16-19, 16-29, 16-46
FOCSTACK 16-19, 16-29, 16-46
MVS 16-20
OFFLINE 16-19, 16-46
UNIX 16-36
Windows NT/2000 16-2, 16-19, 16-29, 16-46

-WRITE command 10-16, 10-18 to 10-19, 10-85

## X

XML control file 3-5

XRETRIEVAL parameter 15-11, 15-77

## Y

YRTHRESH parameter 15-12, 15-77

# 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:_____

E-mail:_____

Comments:

# Reader Comments

**Information Builders, Two Penn Plaza, New York, NY 10121-2898** **(212) 736-4433**