1. Creating a Reporting Application .................................................. 23
   What Is a Reporting Application? .................................................. 24
   Project-based and Remote Development Environments ...................... 24
   Creating a Project ........................................................................... 25
      Reference: File Types ................................................................. 25
      Procedure: How to Create a Project Directory .............................. 26
   Organizing a Project ....................................................................... 31
      Procedure: How to Add a Virtual Folder to a Project and Customize File Types ...... 32
      Procedure: How to Customize Directories for Virtual Folders ............... 34
      Procedure: How to Rename a Component ...................................... 35
      Procedure: How to Remove a Component ...................................... 36
   Using Workspace Files .................................................................... 36
      Procedure: How to Create a Workspace File .................................. 37
      Procedure: How to Save Workspace Files ...................................... 41
      Procedure: How to Close a Workspace File .................................... 41
      Procedure: How to Load a Workspace File .................................... 41
      Procedure: How to Rename the Workspace File ............................... 42
      Procedure: How to Change the WebFOCUS Environment for the Workspace File . . 42
      Procedure: How to Change the Projects Assigned to the Workspace File .......... 43
      Reference: Properties Dialog Box of a Workspace File ....................... 43
      Procedure: How to Remove a Project From the Workspace File .......... 43
      Procedure: How to Insert a Project Into the Workspace File .............. 44
   Adding a Master File to a Project ..................................................... 44
      Procedure: How to Add a Master File to a Project ............................ 45
      Procedure: How to Remove a Master File From a Project .................. 46
      Procedure: How to Permanently Delete a Master File ....................... 46
2. Partitioning and Deploying Project Files ........................................... 103

Deployment Basics .............................................................................. 103
Summary of Steps .................................................................................. 104
Step 1: Identify the Target Servers ................................................................. 104
Step 2: Create a Deployment Scenario ......................................................... 105
  Procedure: How to Create a Deployment Scenario .................................... 105
  Procedure: How to View the Deploy Namespace Properties ....................... 112
Step 3: Partition the Project Files ................................................................. 112
  Procedure: How to Partition the Project Files .......................................... 112
  Reference: Guidelines for Deploying Files ............................................... 115
  Working in a Default Partition Environment .............................................. 115
    Procedure: How to Modify Properties of Partitioned Files ....................... 116
  Setting Running Paths .............................................................................. 117
    Procedure: How to Set Running Paths .................................................... 118
Step 4: Deploy the Project Files ................................................................. 119
  Procedure: How to Deploy Project Files in the Explorer Using the Shortcut Menu ......................................................... 119
  Reference: Scenario Editor ...................................................................... 121
  Reference: Deploying Application Dialog Box .......................................... 122
Configuring the Target Servers (Optional) ................................................ 123
  Reference: APPROOT Variable for Setting the Deployment Path ............... 123
  Procedure: How to Set the Web Server Deployment Path Using the WebFOCUS Administration Console (Windows) ................................................................. 124
  Procedure: How to Set the WebFOCUS Reporting Server Deployment Path Using the Reporting Server Console (Windows) ......................................................... 125
  Procedure: How to Set the Web Server and WebFOCUS Reporting Server Deployment Path Using a Text Editor (Windows and UNIX) ......................................................... 125
  Procedure: How to Set the Deployment Path (AS/400) ............................ 126
  Reference: APPROOT Alias .................................................................... 126
  Procedure: How to Create an Alias on the Web Server (Windows) ............ 127

3. Creating Reports Overview ........................................................................ 129
  Requirements for Creating a Report .......................................................... 129
  Report Types ............................................................................................ 130
  Reporting Tools ........................................................................................ 132
  Developing Your Report Request .............................................................. 132
    Lowercase Directory Names and File Names in WebFOCUS From UNIX .... 134
  Starting a Report Request ........................................................................ 134
    Procedure: How to Start a Report Request ............................................. 134
Creating a Report Example. ................................................................. 136
Saving a Report. ................................................................. 138
  Procedure: How to Save a Report. ........................................ 138
Customizing a Report ................................................................. 139

4. Joining and Merging Data Sources ........................................... 141
   What Is a Join? ................................................................. 141
   Types of Joins ................................................................. 142
     Dynamic Joins. ............................................................... 142
     Conditional Joins .......................................................... 143
     Single Instance and Multiple Instance Joins. ......................... 143
       Reference: Descriptions of Single Instance and Multiple Instance Joins. 143
     Joins Based on Virtual Fields ............................................ 144
     Left Outer Joins ............................................................. 145
     Host Records With No Matching Cross-Referenced Records ......... 145
   Joining Different Kinds of Data Sources ................................. 146
     Data Source Requirements for Cross-Referenced Fields .............. 146
       Reference: Joining Cross-Referenced Fields in Specific Types of Data Sources. 146
     Joining a Data Source to Itself: The Recursive Join .............. 147
   Creating a Join With Graphical Tools .................................... 148
     Reference: Join Toolbar .................................................. 149
     Procedure: How to Create a Join ..................................... 150
     Procedure: How to Create a Join for a Reporting Object in Developer Studio ........ 151
     Procedure: How to Create a Left Outer Join With the Join Tool in Developer Studio .... 152
     Procedure: How to Add Information About Fields to Join Field Lists .............. 152
   Creating Multi-Field Joins .................................................... 153
     Creating a Concatenated Join ........................................... 155
     Using Conditional Joins ................................................ 156
       Procedure: How to Create Conditional Joins ....................... 157
   Creating Cluster Joins ....................................................... 157
     Procedure: How to Create a Cluster Join by Enhancing Existing Synonyms ......... 158
     Procedure: How to Create a Cluster Join Using a New Synonym .............. 164
   Customizing a Join ........................................................... 165
     Procedure: How to Remove a File From the Join ................... 165
     Procedure: How to Delete a Join .................................... 166
Procedure: How to Specify Join Fields .......................................................... 166
Reference: Usage Notes for Specifying Join Fields ....................................... 166
Procedure: How to Create Additional Joins .................................................. 166
Procedure: How to Make a Join a Single Instance ......................................... 167
Changing the Join Name ................................................................................. 167
Procedure: How to Change or Remove the Default Join Name ....................... 167
Procedure: How to Display Current Join Structures ....................................... 167
Using a Virtual Field as the Join Field .......................................................... 167
Procedure: How to Use a Virtual Field as the Join Field ............................... 168
Reference: Join Define in File Dialog Box ....................................................... 169
Adding a Tag Name ......................................................................................... 170
Procedure: How to Add a Tag Name ............................................................... 170
Working With Joins ......................................................................................... 171
Procedure: How to Retrieve a Join ............................................................... 171
Procedure: How to Run a Join ....................................................................... 172
Procedure: How to Clear a Join ................................................................... 172
Reference: Join Clear Dialog Box .................................................................. 173
Procedure: How to Copy a Join From One Procedure to Another .................. 173
Procedure: How to View Join Syntax ............................................................ 173
Procedure: How to View Qualified Field Names ........................................... 174
Reference: Join Properties Dialog Box ......................................................... 175
Reporting With a Join ..................................................................................... 177
Procedure: How to Report With a Join .......................................................... 177
Saving a Join ................................................................................................. 177
Merging Data Sources Using the Match Wizard ............................................ 179
Procedure: How to Use the Match Wizard .................................................... 180
Procedure: How to Edit a Match Request ..................................................... 186
Reference: Usage Notes for Match Requests ................................................ 186

5. Hierarchical Reporting .............................................................................. 187
Reporting Against Hierarchies ...................................................................... 187
Creating a Report With a Multi-Dimensional Data Source ............................ 189
Viewing the Field Lists .................................................................................. 190
Reference: Organization of Variables, Measures, Characteristics, Properties, and Hierarchies ................................................................. 191
6. Creating a Reporting Procedure ........................................ 215

Selecting a Creation Tool .............................................. 216
  Procedure: How to Select a Creation Tool........................... 217
Reference: WebFOCUS Logon Dialog Box. .............................. 218
Procedure: How to Create a Report Procedure With Report Painter... 218
Procedure: How to Create a Reporting Procedure Using the Procedure Window... 219
Procedure: How to Create a Procedure With the Text Editor...................... 220
Reference: Add Procedure Dialog Box..................................... 221
Incorporating a Procedure Into an Application ....................... 222
7. Creating Temporary Fields .................................................. 273
  What Is a Temporary Field? ............................................... 273
  Reference: Types of Temporary Fields. .......................... 273
  Reference: Evaluation of Temporary Fields. ...................... 275
  Reference: Selecting a Temporary Field. ......................... 276
  Defining a Virtual Field ................................................ 277
  Reference: Usage Notes for Creating Virtual Fields. ............ 277
  Procedure: How to Create a Virtual Field. ....................... 278
  Procedure: How to Assign an Alphanumeric Format With the Define Tool. 278
  Procedure: How to Assign a Numeric Format With the Define Tool. 279
  Procedure: How to Assign a Date/Time Format With the Define Tool. 279
  Reference: Define Tool ................................................. 281
  Procedure: How to Specify Missing Value Attributes Using the Define Tool. 281
  Procedure: How to Establish a Missing Segment Location Using the Define Tool. 285
  Defining Multiple Virtual Fields .................................... 286
    Procedure: How to Add a Virtual Field to a Set of Virtual Fields. 286
  Displaying Virtual Fields ............................................ 287
    Procedure: How to Display Virtual Fields. ...................... 287
  Clearing a Virtual Field .............................................. 287
    Procedure: How to Delete a Virtual Field. ...................... 287
  Increasing the Speed of Calculations in Virtual Fields .......... 287
  Applying Dynamically Formatted Virtual Fields to Report Columns. 288
    Procedure: How to Create a Virtual Field Containing Dynamic Formatting. 288
    Procedure: How to Format a Report Column Using Dynamic Formats. 288
  Creating a Calculated Value ......................................... 290
    Reference: Usage Notes for Calculated Field Values. ......... 290
    Procedure: How to Create a Calculated Value. ................ 290
    Using Positional Column Referencing With Calculated Values. 291
    Using ACROSS With Calculated Values. ......................... 291
    Sorting Calculated Values. ...................................... 291
    Screening on Calculated Values. ............................... 291
Creating Temporary Fields Independent of a Master File ................................. 292
Syntax: How to Define a Function. ................................................................. 292
Reference: DEFINE Function Limits and Restrictions. ................................. 293
Procedure: How to Display DEFINE Functions. ......................................... 294
Syntax: How to Clear DEFINE Functions. .................................................. 294

8. Viewing and Printing Reports and Graphs .................................................. 295
Viewing Reports in the Developer Studio Viewer .......................................... 295
Drilling Down for Details ............................................................................. 297
Navigating Through Multi-Page Web Reports .............................................. 297
    Setting Up for On-Demand Paging. ......................................................... 298
    Procedure: How to Format a Report as HTML ........................................ 299
    Procedure: How to Enable On-Demand Paging for HTML Reports .......... 300
Navigating With On-Demand Paging ............................................................ 300
    Procedure: How to Navigate Through a Report (Viewer Control Panel) .... 301
Visualizing Trends in Reports ...................................................................... 301
Applying Bar Graphs .................................................................................... 301

9. Saving and Reusing Report Output ............................................................ 305
Types of Output Files ................................................................................... 305
    SAVE Files ................................................................................................. 306
    HOLD Files ................................................................................................. 306
    PCHOLD Files ........................................................................................... 307
Creating Output Files .................................................................................... 307
    Output Files: When to Create Them ......................................................... 307
    Output Files: Where to Put Them ............................................................. 308
    Output Files: How to Create Them ........................................................... 308
    Procedure: How to Create a SAVE File .................................................. 308
    Reference: Save File Dialog Box ............................................................... 309
    Procedure: How to Create a HOLD File .................................................. 310
    Procedure: How to Create a PCHOLD File .............................................. 310
Output File Formats ....................................................................................... 310
    SAVE File Formats ................................................................................... 311
    HOLD File Formats ................................................................................... 314
    PCHOLD File Formats ............................................................................. 321
Save Report Output as a Native Temporary Table In Report Painter ............. 326
10. Creating an Update Application With Update Assist ......................... 341

Starting Update Assist ............................................. 341
Update Assist (Step 1 of 4): Selecting a Segment and Fields to Update .......... 342
    Procedure: How to Rename a Segment or a Field. ............................... 344
    Procedure: How to Resort Fields in the Segment & Field Options Window. .... 344
    Procedure: How to Use a Range to Validate a Field. ............................ 345
    Procedure: How to Use a Static List to Validate a Field. ......................... 346
    Procedure: How to Use a Dynamic List to Validate a Field. ...................... 347
Update Assist (Step 2 of 4): Selecting Navigation and Style Options .......... 349
Update Assist (Step 3 of 4): Selecting Output File Options .................... 351
Update Assist (Step 4 of 4): Confirming Selections .............................. 352
About Your Update Assist Application .............................................. 353
    Procedure: How to Run Your Update Assist Application. ......................... 353
    Reference: Files Generated By Update Assist .................................. 353
    Reference: Working With Empty or New Data Sources. ........................... 354
<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure: How to Change the Text to Uppercase or Lowercase</td>
<td>380</td>
</tr>
<tr>
<td>Adding Headings and Footings</td>
<td>380</td>
</tr>
<tr>
<td>Procedure: How to Add Headings and Footings</td>
<td>380</td>
</tr>
<tr>
<td>Using Bookmarks to Move Within a File</td>
<td>381</td>
</tr>
<tr>
<td>Procedure: How to Apply or Remove a Bookmark</td>
<td>381</td>
</tr>
<tr>
<td>Procedure: How to Jump to a Bookmark</td>
<td>381</td>
</tr>
<tr>
<td>Procedure: How to Apply Multiple Bookmarks</td>
<td>381</td>
</tr>
<tr>
<td>Procedure: How to Remove Multiple Bookmarks</td>
<td>381</td>
</tr>
<tr>
<td>Running a Procedure From the Editor</td>
<td>382</td>
</tr>
<tr>
<td>Procedure: How to Run a Full Procedure as Code</td>
<td>382</td>
</tr>
<tr>
<td>Procedure: How to Run a Procedure Component as Code</td>
<td>382</td>
</tr>
<tr>
<td>Procedure: How to Run Several Components From the Text Editor</td>
<td>382</td>
</tr>
<tr>
<td>Opening a Graphical Tool From the Text Editor</td>
<td>382</td>
</tr>
<tr>
<td>Procedure: How to Add Code With a Graphical Tool From the Text Editor</td>
<td>383</td>
</tr>
<tr>
<td>Procedure: How to Edit Code With a Graphical Tool From the Text Editor</td>
<td>383</td>
</tr>
<tr>
<td>12. Alert Procedures</td>
<td>385</td>
</tr>
<tr>
<td>Alert Reporting Overview</td>
<td>385</td>
</tr>
<tr>
<td>Components of an Alert</td>
<td>386</td>
</tr>
<tr>
<td>Reference: Alert Test Sample Code Created in Developer Studio With the Editor</td>
<td>387</td>
</tr>
<tr>
<td>The Life of an Alert</td>
<td>389</td>
</tr>
<tr>
<td>Creating an Alert</td>
<td>391</td>
</tr>
<tr>
<td>Procedure: How to Create an Alert in Developer Studio</td>
<td>391</td>
</tr>
<tr>
<td>Checking Scheduled Alerts</td>
<td>400</td>
</tr>
<tr>
<td>13. Reporting Objects in Developer Studio</td>
<td>403</td>
</tr>
<tr>
<td>Reporting Objects</td>
<td>403</td>
</tr>
<tr>
<td>Procedure: How to Create a Reporting Object</td>
<td>403</td>
</tr>
<tr>
<td>Procedure: How to Set a Different Data Server for a Reporting Object</td>
<td>407</td>
</tr>
<tr>
<td>Properties of a Reporting Object</td>
<td>409</td>
</tr>
<tr>
<td>Components of a Reporting Object</td>
<td>412</td>
</tr>
<tr>
<td>Preprocessing Other Component</td>
<td>413</td>
</tr>
<tr>
<td>Joins</td>
<td>414</td>
</tr>
<tr>
<td>Virtual Fields (DEFINE Statements)</td>
<td>415</td>
</tr>
<tr>
<td>Filters</td>
<td>415</td>
</tr>
<tr>
<td>Procedure: How to Create a Filter Group</td>
<td>416</td>
</tr>
</tbody>
</table>
This documentation describes how to use Developer Studio to create a reporting application, which includes project creation and deployment, and the required elements for creating a report. It is intended for administrators and application developers.

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

### How This Manual Is Organized

This manual includes the following chapters:

<table>
<thead>
<tr>
<th>Chapter/Appendix</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Creating a Reporting Application</td>
<td>Describes how to create a reporting application in the local development environment.</td>
</tr>
<tr>
<td>2 Partitioning and Deploying Project Files</td>
<td>Describes the steps you will perform for successful deployment of your project files to the web.</td>
</tr>
<tr>
<td>3 Creating Reports Overview</td>
<td>Describes the tools available for creating and formatting reports.</td>
</tr>
<tr>
<td>4 Joining and Merging Data Sources</td>
<td>Describes how to join two or more related data sources to create a larger integrated data structure from which you can report.</td>
</tr>
<tr>
<td>5 Hierarchical Reporting</td>
<td>Describes how to create reports from multi-dimensional data sources, such as SAP BW. Level and parent/child hierarchy models are supported and reporting rules are automatically enforced.</td>
</tr>
<tr>
<td>6 Creating a Reporting Procedure</td>
<td>Describes how to create a procedure, which is the core element of a reporting application.</td>
</tr>
<tr>
<td>7 Creating Temporary Fields</td>
<td>Describes how to generate information from existing data.</td>
</tr>
</tbody>
</table>


## Chapter/Appendix

<table>
<thead>
<tr>
<th>Chapter/Appendix</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Viewing and Printing Reports and Graphs. Describes how to view and print reports and graphs.</td>
</tr>
<tr>
<td>9</td>
<td>Saving and Reusing Report Output. Describes how to save report data to files for reuse in different formats.</td>
</tr>
<tr>
<td>10</td>
<td>Creating an Update Application With Update Assist. Describes how to use the Update Assist to create an update application in Developer Studio.</td>
</tr>
<tr>
<td>11</td>
<td>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.</td>
</tr>
<tr>
<td>12</td>
<td>Alert Procedures. Describes how alerts enable Managed Reporting Developers to distribute entire reports or selected sections of a report only when certain test conditions are met.</td>
</tr>
<tr>
<td>13</td>
<td>Reporting Objects in Developer Studio. Describes procedures for working with Reporting Objects in Developer Studio.</td>
</tr>
<tr>
<td>A</td>
<td>Glossary. Describes key terms found in this manual.</td>
</tr>
</tbody>
</table>

## Documentation Conventions

The following table describes the documentation conventions that are used in this manual.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>THIS TYPEFACE</strong> or <strong>this typeface</strong></td>
<td>Denotes syntax that you must enter exactly as shown.</td>
</tr>
<tr>
<td><strong>this typeface</strong></td>
<td>Represents a placeholder (or variable) in syntax for a value that you or the system must supply.</td>
</tr>
<tr>
<td><strong>underscore</strong></td>
<td>Indicates a default setting.</td>
</tr>
<tr>
<td>Convention</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td><em>this typeface</em></td>
<td>Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.</td>
</tr>
<tr>
<td>Key + Key</td>
<td>Indicates keys that you must press simultaneously.</td>
</tr>
<tr>
<td>{  }</td>
<td>Indicates two or three choices. Type one of them, not the braces.</td>
</tr>
<tr>
<td>[  ]</td>
<td>Indicates a group of optional parameters. None are required, but you may select one of them. Type only the parameter in the brackets, not the brackets.</td>
</tr>
<tr>
<td></td>
<td>Separates mutually exclusive choices in syntax. Type one of them, not the symbol.</td>
</tr>
<tr>
<td>...</td>
<td>Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).</td>
</tr>
<tr>
<td>.</td>
<td>Indicates that there are (or could be) intervening or additional commands.</td>
</tr>
</tbody>
</table>

### Related Publications

Visit our Technical Content Library at [http://documentation.informationbuilders.com](http://documentation.informationbuilders.com). You can also contact the Publications Order Department at (800) 969-4636.

### Customer Support

Do you have any questions about this product?
Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques. Access Focal Point at http://forums.informationbuilders.com/eve/forums.

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

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

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

**Information You Should Have**

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

- Your six-digit site code (xxxx.xx).
- Your WebFOCUS configuration:
  - The front-end software you are using, including vendor and release.
  - The communications protocol (for example, TCP/IP or HLLAPI), including vendor and release.
  - The software release.
  - Your server version and release. You can find this information using the Version option in the Web Console.
  - The stored procedure (preferably with line numbers) or SQL statements being used in server access.
- The Master File and Access File.
- The exact nature of the problem:
Are the results or the format incorrect? Are the text or calculations missing or misplaced?

Provide the error message and return code, if applicable.

Is this related to any other problem?

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

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

Is this problem reproducible? If so, how?

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

Do you have a trace file?

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

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. You can contact us through our website, http://documentation.informationbuilders.com/connections.asp.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

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

For information on course descriptions, locations, and dates, or to register for classes, visit our website (http://education.informationbuilders.com) or call (800) 969-INFO to speak to an Education Representative.
Creating a Reporting Application

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 the Managed Reporting manual.

In this chapter:

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

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

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

Project-based and Remote Development Environments

Developer Studio provides two main areas in the Object Explorer tree: the Projects area and the WebFOCUS Environments area. From the Projects folder, you can develop projects locally on your machine or against a remote WebFOCUS environment. From the WebFOCUS Environments folder, you develop directly against environments that are configured remotely.

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

Based on the edition of Developer Studio that you have installed, you can:

- Locally develop and deploy self-service applications from the Projects area.
  - This is available for the full Developer Studio edition and the Developer Studio Visual Discovery edition.

- Connect to one or more WebFOCUS remote servers and modify existing self-service applications on those WebFOCUS servers. For example, you can add a reporting procedure to an existing application.

- Configure access to one or more WebFOCUS environments so that you can manage resources on the WebFOCUS Client and Reporting Server, and in the Managed Reporting Repository (if installed). From the environment tree, you can create and edit procedures, metadata, HTML files, and more.
The last two capabilities are available when using the Managed Reporting Developer edition or the Report Developer edition.

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

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

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

Tip: You can organize a project in many ways. For details, see Organizing a Project on page 31.

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

A sample project, called SESSION, is available for you to use.

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

Reference: File Types

The major project components are organized in folders:

HTML Files folder. Stores HTML files (.htm). This file type represents a web form (launch page) and manages the display of reports in a browser. 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 and the Describing Data With WebFOCUS Language manuals.

Procedures folder. Stores procedure files (.fex). This file type is the core element of a project. It specifies the data to be retrieved from a data source and the format in which the data is presented. For more information, see Creating a Reporting Procedure on page 215.

Other folder. Stores any visible file type that is not assigned to an existing virtual folder. For example, if SQL files are available in the application and are not assigned to a different virtual folder through the Filters option, then SQL files are listed in the Other folder. File types visible in the Other folder can be reassigned so they appear in a different virtual folder. For more information about reassigning file types, see Organizing a Project on page 31.

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

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

Procedure: How to Create a Project Directory

1. Do one of the following:
   - Right-click the Projects on environment name node and choose New Project.
   - or
Highlight the Projects on environment name node and click the New icon on the Main toolbar.

The Create a Project - Step 1 of 2 dialog box opens, as shown in the following image.

2. In the first input field, type a name for the new project. The name can be up to 64 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 directory named Sales under APPROOT. The newly created directory 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 visible in the project, files used in the project, and search path configuration.

Important: Do not modify the .gfa file manually. Modification may corrupt the project.

Directory 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 directory associated with another project. In that case, the new project and the existing project will share project files.

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

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

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

The following image shows a completed dialog box.

5. Click Next to continue.

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

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

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

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 49.
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. For details, see How to Add a Virtual Folder to a Project and Customize File Types on page 32.

- You can customize the file types that display in any folder using the Project Properties dialog box. For details, see How to Customize Directories for Virtual Folders on page 34. To change the display, see How to Customize Display of File Types on page 51.

- You can rename a component to identify its function in the project or remove a component that is no longer used. You can rename and remove components, such as folders, subfolders, HTML files, Master Files, and procedures. For more details, see How to Rename a Component on page 35. To remove a component, see How to Remove a Component on page 36.

Note: If you create a project with the same name as another project, the location of the gfa file directory is shown next to it in the Explorer window. The following image shows the gfa file directory.
Procedure:  How to Add a Virtual Folder to a Project and Customize File Types

1. Select and right-click the project. Select New from the context menu, and click Virtual Folder. The New Virtual Folder dialog box opens, as shown in the following image.

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. The following image shows the New Filter dialog box.

![New Filter dialog box](image)

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

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

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

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

Procedure: How to Customize Directories for Virtual Folders

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

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

3. Click OK to apply the changes and close the Edit Filters dialog box.

**Tip:** You can also customize file types and directories from a drop-down list. To access the drop-down list, click the down arrow next to the Filters button on the Explorer toolbar.

**Procedure:**  **How to Rename a Component**

1. Select and right-click the component. Select *Rename* from the context menu.
2. Type the new name at the cursor location.
Procedure: How to Remove a Component

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

Using Workspace Files

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

When using Workspace files, you may:

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

![Workspace File Options](image.png)

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

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

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

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

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

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

   This step is optional. The default location for Workspace files is:

   - **On Windows 7.** `drive:\Users\user_ID\My Documents\Workspaces`.
   - **On Windows XP.** `drive:\Documents and Settings\user_ID\My Documents\Workspaces`.

5. Click *Next*.
Creating Reporting Applications With Developer Studio

The New Workspace Wizard - Page 2 of 2 appears where you may add or remove projects to your workspace, as shown in the following image.

6. Click Add to insert projects into the workspace.
7. Select from the list of available projects.
Note: You may add multiple projects. The available projects are shown in the following image.

8. Click OK to add the project.

The project name is added to the New Workspace Wizard, as shown in the following image.
9. Click Finish.
   The Projects area refreshes, showing the selected Workspace file.

   **Note:** When creating new projects, they are added to the active Workspace file.

**Procedure: How to Save Workspace Files**

1. From the Projects area in Developer Studio, select the Workspace project name.
2. Right-click and select Save Workspace As from the context menu.
3. Type a file name for the Workspace file.

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

4. Click Save.
   The Projects area shows the selected Workspace file by default.

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

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

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

1. From the Projects area in Developer Studio, select Projects on localhost.
2. Right-click and select Load Workspace from the context menu.
3. Navigate to the application folder location where the Workspace file is saved.
   The default location for Workspace files is:
   - **On Windows 7.** `drive:\Users\user_ID\My Documents\Workspaces`.
   - **On Windows XP.** `drive:\Documents and Settings\user_ID\My Documents\Workspaces`.
4. Select the workspace (.gfw) file.
5. Click Open.
   The Projects area refreshes, showing the selected Workspace file.
**Procedure: How to Rename the Workspace File**

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

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

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

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

3. Select the *Workspace* tab to change the target environment. The Workspace tab is shown in the following image.

![Workspace Properties](image)

4. Click *Target Environment* to select the appropriate environment from the list.
5. Click OK.

The target environment appears as the current environment for the Workspace file.

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

**Procedure:** **How to Change the Projects Assigned to the Workspace File**

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

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

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

The Properties dialog box has the following fields and options:

**General tab**

Shows name, location, and other Workspace file information.

**Workspace tab**

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

**Comment tab**

Enables you to add comments for the Workspace file.

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

1. From the Workspace file, select the project name that you are removing.
2. Right-click and select **Remove from Workspace** from the context menu.
   
   The project is removed from the Workspace file.
How to Insert a Project Into the Workspace File

Procedure:

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

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

The Workspace file refreshes, showing the added project name.

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

Adding a Master File to a Project

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

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

- **Access File.** Contains additional information needed by WebFOCUS to access data. This information, which is required by many data sources, includes the real name and location of the data source.

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

There are several ways to provide metadata to a project:

- **Add visible files to a project.** When you designate one or more project paths in the Project Wizard, all of the Master Files in those paths are visible to your current project and can be made accessible for development and deployment.
When you add a visible Master File to a project, Developer Studio creates an active link to the file. It does not copy the Master File to the project directory. For more information, see How to Add a Master File to a Project on page 45.

- **Create synonyms.** Using the Create Synonym tool, you can create synonyms for remote data sources configured with the WebFOCUS Reporting Server or for tables that reside on a subserver. For more information, see the Describing Data With Graphical Tools manual.

- **Enhancing a synonym.** Intended primarily for use by a database administrator, the Synonym Editor can be used to view and edit attributes of synonym components. An application developer can use this tool to view the structure of a data source to make minor, authorized modifications to the Master File. The Synonym Editor also validates changes to a synonym and displays appropriate error messages. For more information, see the Describing Data With Graphical Tools manual.

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

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

1. Open the Master Files folder in a project.

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

3. Right-click a Master File from the available Master Files list in the right window pane and select Add to Project.
   Or
   Select a Master File and click the Inserts the selected items into the project button on the Explorer toolbar.
   The Master File icon changes color to signify that it has been added to the list of Master Files in your project and the file type description changes to Master File.

**Tip:** You can add two or more Master Files at one time. Hold down the Ctrl key to select multiple Master Files.
Procedure: How to Remove a Master File From a Project

1. Open the Master Files folder in a project.

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

   The right window pane displays a list of all Master Files in the paths defined for the selected project. Master Files that are added to the project path have a color icon next to them and the file type description is Master File.

3. In the right window pane, right-click a Master File and choose Remove from Project.

   or

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

   Click Yes to confirm removal of the Master File.

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

Procedure: How to Permanently Delete a Master File

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

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

Uploading Data Files

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

Upload Data is also available from the Business Intelligence Portal (BI Portal) and the Business Intelligence Dashboard (BI Dashboard).

When the following types of files are uploaded, a synonym is created for each as part of the upload process:

- **Delimited flat files.** Among the file types you can upload are: txt, csv, ftm, dat, and tmp.

  The Upload Data facility supports a wide range of delimiters and enclosure options.
Excel Workbooks. You can upload and create synonyms for multiple worksheets. The Upload Data facility supports Named Ranges in Excel® worksheets.

XML files.

JSON files.

The Upload Data facility uses standard Reporting Server synonym creation syntax. All data types are supported. You can upload large data files and are not limited to java memory resources. The Upload Data facility includes an interface for editing the resulting Master File that allows users to easily set up field attributes. In addition, you can open the synonym in the Synonym Editor, which provides complete editing facilities and error handling.

The applications that are available to the user are controlled by user profiles. The Reporting Server provides additional privileges to control user permissions.

Starting the Upload Process

In Developer Studio, right-click the Master Files folder and select Upload Data from the New menu.

In the BI Portal and BI Dashboard, the Upload Data tool is launched by right-clicking a folder and selecting Upload/Data from the context menu.

The following image is an example of working in the Projects area of Developer Studio. Similar steps are performed for working in the Data Servers area. As shown in the following image, in the Repository area, the option to Upload Data is available from the folder level, and the option is controlled by user privileges.
Uploading Data Files

The Upload Data dialog box opens, as shown in the following image.

On the Console toolbar, click the Wizards icon to access the Upload Wizard. For more information, see the Metadata chapter in the Server Administration manual.

**Reference: Upload Data File Considerations**

- The time required to complete the upload depends on the machine configuration (for example, type of processor, amount of memory, and speed). The Upload utility has been tested with Excel files up to 60MB with 140K lines.

- If your input file contains empty lines, such as a carriage return or a line feed character at the end of the file, they will be preserved in the upload file.

- If your input file contains any other empty lines, they will not get included in the uploaded file.

- When uploading data from an Excel spreadsheet:

  - The data must be arranged in an appropriate tabular format and the spreadsheet must have the same type of data in each column and the same number of fields in every row.

  - Cells with formulas and special characters are not supported. For example, if percentages are used, the cells should be formatted using the percentage data type and should not have the percentage special character "%" in the cell.

  - Files can reside on the network, local machine and can be accessed using the drive letter (for example, c:\) or a UNC path.
The folder location and file name can contain spaces.

- The file name can be up to 64 characters.

Viewing and Modifying Project Properties

For more information, see How to View Project Properties on page 49.

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

  When a project is in read-only mode, new files cannot be created, since an entry needs to be made in the .gfa file. A message will be presented to the developer.

- **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.

  Each project can have a unique search path that is independent from the search path set on the WebFOCUS Reporting Server.

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

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

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

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

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

Procedure: How to View Project Properties

1. Right-click a project and then click Properties.

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

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

2. You can modify certain project properties using this dialog box.
Procedure: How to Apply Read-only Security

1. Right-click a project and then click Properties.
   The properties dialog box opens.
2. On the General tab, click the Read-only check box.
3. Complete the dialog box as described in this topic. Click OK to apply changes and close the dialog box, or click Apply to temporarily apply changes and keep the dialog box open. Click Close to apply the temporary changes.

Procedure: How to Modify Directory Paths

1. Right-click a project and select Properties.
   The properties dialog box opens.
2. Click the Paths tab, as shown in the following image.

![Sales Properties dialog box]

Make the following modifications to the directory paths:
To add a new directory path, click the Add new directory button. Select from the available folders and click OK. The new directory is added to the Related Directories list.

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

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

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

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

1. Right-click a project and choose Properties.

   The properties dialog box opens.
2. Click the *Edit Filters* tab, as shown in the following image.
The Filters list box shows the file types associated with the project. You can make the following modifications to the list:

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

Select a registered file type and click **OK**.

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

3. Click **OK** when you are done.

**Procedure:** How to Add a Comment

1. Right-click a project and select **Properties**. The properties dialog box opens.

Creating Reporting Applications With Developer Studio 53
2. Click the Comment tab, as shown in the following image.

![Comment Tab Image](image)

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

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

The comment displays as a ToolTip when you hover over the project name in the Explorer. The same applies to the virtual folders.

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

1. Right-click a project and select Properties.

   The properties dialog box opens.
2. Select the Deployment tab, as shown in the following image.

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

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

4. Click the Set button to open the Set Starting Object dialog box and make your selection. Typically, the starting object for a deployed application is an HTML file. The dialog box contains all the valid starting objects created for this project.

5. Click OK to apply changes and close the dialog box.

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

For details about deployment, see Partitioning and Deploying Project Files on page 103.
Reference: Project Properties Dialog Box

**General Tab**

![Sales Properties dialog box](image)

**Type**

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

**Location**

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

**Size**

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

**Modified**

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

**Attributes**

Read-only prohibits modification to or deletion of a project.
**OK**

Confirms changes and returns to the Explorer.

**Cancel**

Aborts changes and returns to the Explorer.

**Apply**

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

**Paths Tab**

![Sales Properties](image)

**Related Application Paths**

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

Adds a directory to the project search path.

**Remove Directory**

Deletes a directory from the project search path.

**Move Directory Up**

Moves a directory up in the project search path.

**Move Directory Down**

Moves a directory down in the project search path.

**OK**

Confirms changes and returns to the Explorer.

**Cancel**

Aborts changes and returns to the Explorer.

**Apply**

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

Filters

Are the file types associated with a project. Use the Add New File Type Filter(s) and Remove File Type Filter(s) buttons to modify the Filters list.

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

Adds a filter.

**Remove File Type Filter(s)**

Removes a filter.

**OK**

Confirms changes and returns to the Explorer.

**Cancel**

Aborts changes and returns to the Explorer.

---

Creating Reporting Applications With Developer Studio

59
**Apply**

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

**Comment Tab**

![Comment Tab Image]

**Comment**

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

**OK**

Confirms changes and returns to the Explorer.

**Cancel**

Aborts changes and returns to the Explorer.

**Apply**

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

![Sales Properties dialog box](image)

**Active Scenario**

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

**Target Application Name**

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

If you are deploying files to your development server, you must provide a target application name other than your development directory.
If you are deploying files to a WebFOCUS Reporting Server on z/OS, the target application name can be up to eight characters. For deployment servers on all other supported platforms, the name can be up to 64 characters. Spaces are not allowed. (If a space is entered, it is converted to an underscore.)

**Starting Object**

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

**Smart Deploy**

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

**Ignore unresolved items**

Ensures that deployment is completed in a case where the project contains an unresolved file. An unresolved file can be any of the following:

- A file that once belonged to the project but was deleted outside of Developer Studio.
- A file from another project in which it was shared.
- A file located in a directory that changed and can no longer be found.

**Compile Maintain Procedures**

Compiles Maintain procedures during deployment. Compilation improves application performance. For more information, see the *Developing WebFOCUS Maintain Applications* manual.

**Compile for verbose Maintain trace (debug)**

Reserved for debugging purposes. It can affect performance. For more information, see the *Developing WebFOCUS Maintain Applications* manual.

**Include Server Paths**

Allows the deployment process to include server paths in the Call and Exec statements within Maintain procedures.

**Keep the form generation code in deployed .MNT file.**

Deploys .MNT files with any original form code intact. This option is enabled by default. A decrease in the speed of deployment may be noticed if a project contains many forms.
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 properties dialog box.

Cancel

Aborts changes and returns to the properties dialog box.

Managing a Project With Source Control

Developer Studio supports the following third-party source code management products:

- Microsoft Team Foundation Server
- Microsoft Visual SourceSafe
- Subversion
- CVS
- ClearCase
Source code management (called source control in Developer Studio), provides version control for your individual and multi user application resources, allowing multiple users to work on the same files. A source control product manages access to source code and keeps track of all code changes.

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

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

The WebFOCUS Developer Studio Source Control feature uses the Source Code Control API (SCC API) that is utilized by the most popular Version Control Systems (VCS) on the market. The SCC API is an industry standard specification for Version Control Systems (VCS) connectivity defined by Microsoft.

Reference: Requirements for Source Control

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

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

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

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

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

Procedure: How to Determine if Source Control is Enabled

1. In the Developer Studio Explorer, select Options from the Window menu.
2. From the Developer Studio Options dialog box, select the Source Control tab.
   The source control system or systems detected by Developer Studio during startup are listed.
   **Note:** If no systems are detected this tab indicates that no Source Control providers are available.
3. If more than one system is listed, check the one you want to use, and click OK. Otherwise, Developer Studio will use the system that is listed and checked by default.
   **Note:** If you do not want to use the Source Control feature, uncheck the selected system.
4. Click Cancel to close the dialog box and return to Developer Studio Explorer.

Procedure: How to Access Source Control Options

You can implement source control on a Project or in the Managed Reporting area while selecting a folder or file under the top level Content folder. Developer Studio writes information regarding source control in the .gfa file of the project when you select the option Add to Source Control. For Managed Reporting, information is stored in the WebFOCUS database.

The following information pertains to using Source Control from the Projects area. For more information on using source control from the Managed Reporting environment, see How to Use Source Control From Managed Reporting on page 66.

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

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

Access source control options in one of the following ways:

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

**Procedure: How to Use Source Control From Managed Reporting**

From the Managed Reporting area of Developer Studio, shown as Content in the explorer tree, you can add folders to the Version Control System (VCS) database by creating a new project for the selected folder.

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

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

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

2. Click the Source Control icon on the toolbar, then click Add to Source Control. Individual files or folders can be added using the right-click context menu.
You are prompted to connect to the Version Control System (VCS) database, as shown in the following image.

3. Log on to the Version Control System (VCS) database as follows:
   a. In the Username field, type a valid user name for the Version Control System (VCS) database.
   b. In the Password field, type a valid password for the Version Control System (VCS) database.
   c. In the Database field, type the full path of the Version Control System (VCS) database. You can also click Browse to navigate to the location of the Version Control System (VCS) database.
   d. Click OK.

Once authentication is successful, the Add to SourceSafe Project window opens, as shown in the following image.
4. Select the project directory and the project name. For project name, it is recommended that you use the name provided by Developer Studio to avoid mapping issues between the Managed Reporting folder name and the Version Control System (VCS) project. It also helps developers match and identify the projects.

You can:

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

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

5. If this is a new project, a message appears asking whether you want to create the project. Click Yes.

A window opens that shows the files that will be added to the Version Control System (VCS) project you created, as shown in the following image.

6. Click OK.
The files and their Source Control status are visible in Developer Studio, as shown in the following image.

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Type</th>
<th>Source Control</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>cargraph</td>
<td>1.54 KB</td>
<td>Focus Executable</td>
<td>Controlled</td>
<td>03/30/05 03:42 PM</td>
</tr>
<tr>
<td>carinst</td>
<td>2.56 KB</td>
<td>Focus Executable</td>
<td>Controlled</td>
<td>03/30/05 03:42 PM</td>
</tr>
<tr>
<td>carinst-1</td>
<td>3.14 KB</td>
<td>Focus Executable</td>
<td>Controlled</td>
<td>03/30/05 03:42 PM</td>
</tr>
<tr>
<td>carinst2</td>
<td>656 bytes</td>
<td>Focus Executable</td>
<td>Controlled</td>
<td>03/30/05 03:42 PM</td>
</tr>
<tr>
<td>gr1</td>
<td>1.53 KB</td>
<td>Focus Executable</td>
<td>Controlled</td>
<td>03/30/05 03:42 PM</td>
</tr>
<tr>
<td>Join1</td>
<td>1.49 KB</td>
<td>Focus Executable</td>
<td>Checked Out</td>
<td>03/30/05 03:42 PM</td>
</tr>
<tr>
<td>OLAP_Test</td>
<td>168 bytes</td>
<td>Focus Executable</td>
<td>Controlled</td>
<td>03/30/05 03:42 PM</td>
</tr>
<tr>
<td>pdf_test</td>
<td>43 bytes</td>
<td>Focus Executable</td>
<td>Checked Out</td>
<td>03/30/05 03:42 PM</td>
</tr>
</tbody>
</table>

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

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

To check in file(s), select the required file(s), right-click the selection, and select Check In.

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

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

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

### Reference: Current Limitations for Source Control

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

### Reference: Source Control Icons

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

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

**Procedure:** How to Restore Write Access to a Single File

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

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

**Procedure:** How to Restore Write Access to a Project

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

In the Developer Studio Explorer, notice that the attribute R (read-only) has been removed from the project.
Using AccuRev as the Source Control Provider

This section describes the requirements needed in order to use AccuRev as a source control provider.

In order to use AccuRev as a source control provider with Developer Studio, you must have installed both the AccuRev plug-in, AccuBridge for Microsoft SCC, and Developer Studio Release 8.0 Version 02 or higher.

You can download an AccuRev plug-in from the AccuRev website.

Procedure: How to Download and Install AccuRev

1. Download the AccuRev plug-in file from the AccuRev website.
2. Close all instances of IDEs which support MSCCI APIs, such as Microsoft Visual Studio, Microsoft Access, and Developer Studio.
3. Extract the .zip file to a folder.
4. Register the AccuRev.dll and ScceAcc.dll.
   a. Open the Command Prompt window.
      
      **Note:** If your machine is using Windows 7 or higher, use the *Run as administrator* option to open the Command Prompt window.

   b. Navigate to the folder where you extracted the .zip file.
      a. Enter the following commands in the Command Prompt window:
         
         ```
         regsvr32 AccuRev.dll
         regsvr32 ScceAcc.dll
         ```
         
         The two .dll files have been registered.

5. Run the SwitchSCC.exe program.
   
   The Switch Source Control Provider dialog box opens.

6. Select AccuRev and click OK to close the Switch Source Control Provider dialog box.
   
   The AccuRev source control provider is successfully installed.
To set AccuRev as the source control provider, navigate to the Source Control tab in the Developer Studio Options dialog box and select AccuRev, as shown in the following image.

Using Microsoft Team Foundation Server as the Source Control Product

This section describes the requirements needed in order to use and set Microsoft Team Foundation Server as a source control.

Developer Studio 8 and higher supports Microsoft Team Foundation Server as a source control provider. Information Builders has tested and certified Microsoft Team Foundation Server 2012.

Microsoft Team Foundation Server Requirements

In order to use Microsoft Team Foundation Server as a source control provider with Developer Studio, you must have the Microsoft Visual Studio® Team Foundation Server 2012 MSSCCI Provider 32-bit plug-in installed. The plug-in is available through the Microsoft website.

You can run the MSSCCI software using the following environments:


Microsoft Visual Studio Team Explorer can also be installed on the same machine as Developer Studio so that you can compare files and perform other tasks such as manage team projects.

**Procedure:** How to Set Microsoft Team Foundation Server as the Source Control Provider

1. While in Developer Studio, click *Window*, and select *Options*.

   The Developer Studio Options dialog box opens.

2. Select the *Source Control* tab.

3. Select the *Team Foundation Server MSSCCI Provider* check box from the list of available Source Controls.

   Once the Team Foundation Server MSSCCI Provider option is selected, Microsoft Team Foundation Server is used as your source control provider.

   **Note:** The Team Foundation Server MSSCCI Provider check box is selected by default if the Microsoft Team Foundation Server is the only source control you have installed.

**Using Subversion (SVN) as the Source Control Product**

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

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

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

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

You will have to setup and configure the PushOk SVN plug-in and specify the PushOk SVN Proxy properties.
**Procedure: How to Start the Subversion (SVN) Configuration Tool**

1. Install the PushOk SVN plug-in.
2. From the start menu, select **SVNSCC** from the PushOk Software menu.
3. Select **Configure plugin** from the SVNSCC submenu, as shown in the following image.

![Configure plugin image]

The PushOk Subversion (SVN) Proxy properties dialog box appears.

**Procedure: How to Set the Subversion (SVN) User Options**

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

1. The SVN executables tab enables you to set tools used by the SVN SCC API to expose standard functionality such as running differences, repository viewers, O/S integration tools, and so on.
   - Type or select an SVN GUI viewer application in the GUI executable field. Since the PushOk SVN SCC client does not provide its own stand-alone viewer, you can specify a viewer by passing it to the current connection string. This is necessary in the case that you would want to use a GUI application instead of the command line to query the SVN repository.

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

   - Select a **Diff/merge** tool.

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

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

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

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

   a. To Add a new SVN connection string:

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

      The string is added to the List section box.

   The following image is an example of the Known lists tab on the PushOk SVN Proxy properties dialog box.

![Known lists tab example](image)

**Using CVS as the Source Control Product**

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

The CVS requires that a third-party product (plug-in) is used to provide the SCC Compliance that is used by Developer Studio to connect to the Version Control System (VCS) repository.
**Note:** Although you may use any CVS plug-in, this section is specific to a product provided by PushOk.

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

**Procedure: How to Start the CVS Configuration Tool**

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

The PushOk CVS Proxy properties dialog box appears.

**Procedure: How to Set the CVS User Options**

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

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

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

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

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

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

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

Select a file viewer tool from the Viewing field.

**Note:** PushOk CVS provides a built-in file viewer but you may also specify a custom tool.
The following image is an example of the CVS executables tab on the PushOk CVS Proxy properties dialog box.
3. The CVS options tab enables you to specify how files are treated by CVS and how the status of these files are reported.

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

   a. To Add a New CVSROOT String:
Select Known Roots from the Select list type drop-down list.

Type the CVSROOT string in the input field under the List section.

Click Add.

The string is added to the List section box.

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

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

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

b. **To Add New Projects:**

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

Select Known Modules from the Select list type drop-down list.

Type the name of the module in the input field under the List section. This is the relative path to the repository directory.

Click Add.

The module is added to the List section box.

**Note:** There is no default module option.
5. The Server options tab enables you to specify how the connection to the server is being handled.
   - Select a string from the Select CVSROOT drop-down list.
     
     **Note:** It is recommended to select the default string to avoid confusion. If the selected string is not the default, type the desired string and click Default.

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

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

   - The Encryption option is only enabled for protocols which support encryption. For example, sspi.
Select the **Check filenames on server** option if a UNIX machine hosts the CVS server. This will resolve possible conflicts of file name case-sensitivity.

The following image is an example of the Server options tab on the PushOk CVS Proxy properties dialog box.

---

**Using ClearCase as the Source Control Product**

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

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

- **Snapshot Views** denote a physical location on a hard drive.

  **Note:** Some of the user interface requests are available through the Windows Explorer ClearCase context menu.

- **Dynamic Views** are client logical views of a VOB. VOB is a repository for storing versioned projects and metadata.

  **Note:** Through Windows Explorer, limited operations are allowed to be performed on the set of exposed files. For instance, a checked in file under ClearCase control cannot be removed using Windows Explorer commands or other Windows API call. The same thing is true for changing the Read-Only flag.

Snapshot Views work while disconnected from the network. Views are updated when you perform manual requests. However, this can cause views to become out of date. The file loaded during checkout may not necessarily be the latest version in the VOB.

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

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

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

From Developer Studio, applying a Source Code Control operation on a file found on a remote WebFOCUS Environment involves an HTTP copying from the WebFOCUS application to the view directory, or the other way around. Depending on the view type, these operations could also involve modifying file attributes or timestamps.
**Tip:** Since the view is just a file transfer place between the ClearCase system and the WebFOCUS application, some problems arise with the Snapshot Views as files added to the view may be out of date, or may already be checked in by another developer. Given all these limitations, Dynamic Views are recommended.

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

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

1. From the ClearCase Explorer Tools menu, select **ClearCase Options** from the Options submenu.
2. In the Check In/Add to Source Control section of the Operations tab, select the **Use original checkout information** and **Preserve file modification time** options.
3. In the Check Out section of the Operations tab, select the **Preserve file modification time** option.

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

Reference: ClearCase Notes and Recommendations

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

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

Note: ClearCase can handle most merging aspects. There are, however, cases in which conflicts may arise that will require intervention by the developer. To avoid issues that may occur as a result of merging, single stream development is recommended.
Developers working with WebFOCUS files and applications must be aware that this behaves like a shared file system. Any modification of a WebFOCUS file could be seen by all other developers using that specific file. Working with the file is taking place at the WebFOCUS location where that file resides. The Source Code Control system just imposes restrictions on accessing that specific file.

Setting Up Developer Studio With ClearCase, CVS, Subversion (SVN), or ChangeMan DS

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

Procedure: How to Set the Developer Studio Source Control Options

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

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

- For CVS, Subversion (SVN), and ChangeMan DS, install and configure the client software or the required plug-in, as described in the CVS and Subversion sections. For details, see Using CVS as the Source Control Product on page 78 and Using Subversion (SVN) as the Source Control Product on page 74.

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

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

![Developer Studio Options dialog box](image)

**Note:** If more than one version control system is detected, you must specify which version control system to use.

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

3. Click the *Settings* button.

   The SCC Development Folders dialog box appears.

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

4. From the SCC Development Folder dialog box, type or select a directory in the *Default Development Directory* field.
- **Note when using ClearCase:** This directory should be an existing directory inside a Dynamic view. To avoid errors, it is recommended that you browse to select the development directory. The directory specified in the Development Directory will be used as the root directory for all ClearCase local operations.

- **Note when using CVS:** This is the directory where all CVS file transfers will take place before or after the CVS actions files can be copied to or from their real location in this directory.

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

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

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

![SCC Development Folders](image)

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

![WebFOCUS Developer Studio](image)

**Reference:** Validating the Source Control Variable in WebFOCUS

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

When specifying values for the variable SCC_NAME:

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

- The value needs to be unique for every WebFOCUS environment accessing the same Source Control repository, and the name should not contain spaces or any of the following special characters:
  ```
  / \ * ' ? : * < > | .
  ```
Note: Avoid changing the value once files have been added to the Source Control repository. If you need to change the value, either all files will have to be added to the repository again, or the repository will need to be changed to be consistent with the new value. This is because the SCC_NAME value is used to establish the file identity in the repository. Changing the value will require that all files are added to the repository again or that the repository be changed to be consistent with the new value.

Procedure: How to Set the SCC_Name Variable in the WebFOCUS Client

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

1. From the WebFOCUS Administration Console, click General under the Configuration section.
   a. Identify the development environment for the Source Control in the SCC_NAME field.
      
      Note: By default, there is no value set for this field.

   b. Provide a name for the SCC_Name variable.

   c. Click Save to save the Client Settings.
The following image is an example of the WebFOCUS Administration Console showing the SCC_NAME Field.

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

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

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

Get Latest Version

Select this option to view a file without changing it. This option retrieves the latest version of the selected file and provides you with a read-only copy.

If you attempt to make changes, they are not saved.

Check Out

To make changes to a file in source control, you must first check it out of the database.

The Check Out option retrieves the latest version of the selected file and allows you to edit it. Source control removes the attribute R (read-only) from a checked out file.

A file that has been checked out displays the description Checked Out in the Developer Studio Explorer.

Check In

Select this option to return a checked out file to the source control database. Source control stores any changes made to the file in the database.

Undo Check Out

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

Add to Source Control

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

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

A file added to source control displays the description Controlled in the Developer Studio Explorer.
In this example, the HTML files have been added to source control.

<table>
<thead>
<tr>
<th>All Folders</th>
<th>Contents of HTML Files</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>launch_demographic_...</td>
</tr>
<tr>
<td></td>
<td>launch_qry.htm</td>
</tr>
</tbody>
</table>

**Remove From Source Control**

Select this option to remove a file from the source control database.

A file that has been removed from the database, or one that has not been added to the database, displays the description Not Controlled in the Developer Studio Explorer.

**Show History**

Select this option to view information about a file, such as version history, date of creation or modification, and other details.

**Show Differences**

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

**Source Control Properties**

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

**Share From Source Control**

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

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

A shared file that is checked out by one project displays the description Exclusive Checked Out & Shared in the other project.
**Refresh Status**
Select this option to refresh the display of file status in the Developer Studio Explorer.

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

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

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

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

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

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

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

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

- When you add files to Visual SourceSafe, SourceSafe assigns the attribute read-only (R) to them. When you check out a file, SourceSafe removes the read-only attribute to allow edits to the file.
When you remove files from SourceSafe, SourceSafe retains the read-only attribute originally assigned. You must manually change the attribute to restore write access. For more information, see *How to Restore Write Access to a Single File* on page 71 and *How to Restore Write Access to a Project* on page 71 for instructions.

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

**Securing a Project**

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 context menu. You are asked if you want to delete the .gfa control file and, optionally, the associated files. Respond to the prompts as desired.
   - or
   - Press the *Delete* key. You are asked if you want to delete the .gfa file and, optionally, the associated files. Respond to the prompts as desired.
Removing a Project
Partitioning and Deploying Project Files

After you create, test, and debug a project in the development environment, you are ready to make it available as a live application on the web environment. This process involves moving certain project files to selected target servers in a WebFOCUS environment. Developer Studio automates the process for you.

This section describes the steps you will perform for successful deployment of your project files to the web environment.

**In this chapter:**
- Deployment Basics
- Summary of Steps
- Step 1: Identify the Target Servers
- Step 2: Create a Deployment Scenario
- Step 3: Partition the Project Files
- Step 4: Deploy the Project Files
- Configuring the Target Servers (Optional)

### Deployment Basics

Once you create project files as described in *Creating a Reporting Application* on page 23, you can partition and deploy them to a WebFOCUS environment.

A WebFOCUS environment consists of:

- A web server
- A WebFOCUS Reporting Server
- Alternate server nodes defined in the Data Servers area

**Note:** The web environment may be a web server or application server or a combination of both, depending on your environment. This topic refers to deployment while using a web server to hold the static content. For example, .html and .gif files are required configuration for such an environment.
If you need to place static content on the application server, you must define an Application Root directory (APPROOT) context root for deploying files and perform required configuration steps. The APPROOT context root is used during the deploy process and when you run the deployed web-based application.

**Partitioning** the files means that you identify the target web server on which the web-based files will reside, and the WebFOCUS Reporting Server on which files, such as procedures will reside. You define the partitioning of files in a deployment scenario.

**Deployment** means that Developer Studio moves the files to the target servers, where they make up an application that users run on the web.

The Developer Studio deployment feature allows you to:

- Create multiple deployment scenarios. You can define multiple deployment scenarios and save them for future use. For example, you might have two deployment scenarios for a project, one that maps the project files to a test environment, and another that maps the files to a production environment.

- Deploy files to multiple servers. You can deploy project files to multiple WebFOCUS Reporting Servers or Maintain servers. This capability enables you to access data on multiple servers, run report components in the most suitable environment, and accelerate application processing.

### Summary of Steps

To take your project live, do the following:

- **Step 1: Identify the Target Servers** on page 104
- **Step 2: Create a Deployment Scenario** on page 105
- **Step 3: Partition the Project Files** on page 112
- **Step 4: Deploy the Project Files** on page 119

You can optionally configure the target servers for your site. For details, see *Configuring the Target Servers (Optional)* on page 123.

### 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 the *WebFOCUS 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.
2. From the shortcut menu, select Project Deployment, then select New Scenario.
   The New Scenario Wizard dialog box opens, as shown in the following image.

3. Respond as follows:
   - **Scenario name.** Type a name for the new deployment scenario. You can change the name at a later time with the Rename option.
Use smart deploy. Choose this option to deploy only the files that have changed since the last time you deployed the application. This option can significantly increase the speed of deployment.

When Smart Deploy is active for a deployment scenario, after its first successful deployment, date and time information is stored in your Project GFA file for each file being deployed. Each subsequent Deploy action then compares the date and time stamp of the source files in your project against the date and time stamps stored in the Deployment Scenario. With Smart Deploy, only newer files will be deployed to the destination, where the source file has a newer time stamp than the one stored in the GFA.

Note:
- The date and time information is stored uniquely in each deployment scenario, so if your project has multiple deployment scenarios, there is no ambiguity about when the files were last deployed using a specific scenario. Take note that each time you first use a new deployment scenario, Smart Deploy performs a full deployment of all files referenced for deployment in your project. Subsequently, only changed files will be deployed.
- Smart Deploy stores file date and time stamps in the GFA after a successful deployment. If an error occurs during a first deployment of a new scenario, the date and time stamps are rolled back.
- If you add a new file to be deployed to a deployment scenario, the GFA has no date and time stamp for a file, so it will always be deployed the first time.

Ignore unresolved items. This option ensures that deployment is completed in a case where the project deployment scenario contains an unresolved file. An unresolved file can be either:

A project file that was used in a deployment scenario and deleted outside of Developer Studio (or from another project in which it was shared).

or

A file located in a directory that changed and can no longer be found.

If this option is selected, the deployment scenario will not fail if it contains unresolved files. However, when the deployment process completes, it will display information about files skipped during the process because they were unresolved.
- **Compile Maintain procedures.** Choose this option to compile Maintain procedures during deployment. Compilation improves application performance. For more information, see the *Developing WebFOCUS Maintain Applications* manual.

- **Compile for verbose Maintain trace (debug).** This option is reserved for debugging purposes. It can affect performance. For more information, see the *Developing WebFOCUS Maintain Applications* manual.

- **Include Server Paths.** Choose this option to allow the deployment process to include server paths in the Call and Exec statements within Maintain procedures.

- **Keep the form generation code in deployed .MNT file.** Choose this option to deploy .MNT files with any original form code intact. This option is enabled by default. A decrease in the speed of deployment may be noticed if a project contains many forms.

  Although this code (XML) is not needed at run time once the .WFM files have been created during the deploy process, keeping it in the remote copy of the file can be an added safeguard should anything happen to the original version. MNT files deployed without the form code cannot be reopened in an MDE session.

  **Note:** The XML code for the form should never be edited manually. Use the MDE to open and edit .MNT files containing forms.
4. Respond as follows:

- **Default partition environment.** From the drop-down list, select a WebFOCUS environment that will be used by default when you partition files. Developer Studio automatically partitions project files for you based on the following:

  HTML, GIF, JPG, VBScript, and JavaScript files are partitioned to the default web server. Executable WebFOCUS procedures (.fex and .mnt files) are partitioned to the default WebFOCUS Reporting Server.

  By default, Developer Studio does not partition Master Files and Access Files in order not to overwrite production files with the same name on the deployment server.

  Selecting an environment from this list does not force the deployment of files as defined in the scenario.

  You can change the definition of the scenario.

- **Default reporting server.** From the drop-down list, you can select which WebFOCUS Reporting Server to use as your Default partition environment.
2. Partitioning and Deploying Project Files

This list is not active if you choose the value NONE for the Default partition environment.

- **Target application name.** Optionally, type an application name that will be used on the web server and WebFOCUS Reporting Server deployment paths, or use the default value (the current project name).

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

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

- **Automatically partition the project files.** This sets default locations based on file types.

  Click Next to proceed to the next Scenario Wizard window, as shown in the following image.

![New Scenario Wizard - Choose starting object](image)

5. Respond as follows:

- **No starting object.** Does not associate a starting object for the scenario.
Select from project items. From the drop-down list, you can select a file from the application as the starting object.

This option is enabled only when files are added to the application and when the option to partition files has been selected.

Select from WebFOCUS Environments. Allows you to select a starting object from an existing configured WebFOCUS Environment, for example, a general launch page.

You must ensure that the deployed application has access to the file to avoid an error at run time.

Select a web page. Allows you to select a URL as a starting object. To specify a webpage, type the URL as a complete path to a file.

You must ensure that the deployed application has access to the file to avoid an error at run time.

Click Next to proceed to the next Scenario Wizard window, as shown in the following.
6. Click Finish to complete the deployment process. The following screen appears.

7. Close or minimize the deployment scenario window, known as the Scenario Editor, to return to the Explorer. The project displays a new folder labeled Deploy. Later, when you have defined deployment scenarios, you can display them by clicking the folder. The following image shows the Explorer.
**Procedure: How to View the Deploy Namespace Properties**

1. Right-click the Deploy namespace and choose Properties.
   
   The Deploy Properties dialog box opens, displaying the General, Comment, and Deployment tabs.
   
   The General tab is selected by default. It displays the CLSID, ProgID, and a description.
   
2. Optionally, type a descriptive comment for the deploy namespace in the Comment field.

**Step 3: Partition the Project Files**

After creating a deployment scenario, you define the details of that scenario by identifying project files for distribution to target servers in a WebFOCUS environment.

This topic applies to deployment scenarios that you created in Step 2: Create a Deployment Scenario on page 105. It does not apply to the supplied scenario Local Deploy. Developer Studio uses Local Deploy to prepare files for Maintain and to run on a local server. This option is visible only after you create the first deployment scenario. Do not modify or remove Local Deploy.

**Procedure: How to Partition the Project Files**

1. Open the Deploy folder for the project. Right-click the deployment scenario that you are defining, and select Open.
   
2. The left window pane displays information about the individual files that are visible from the project search path.
   
   The icons next to some files are unavailable. This indicates that the files are available, but not active for your project. You can click the binoculars icon to show the files that you added to your project and hide the inactive files. However, you can deploy even the files that you did not add to your project.

   The value of Assigned is No if the file has not been assigned to a server for deployment. Assigned is Yes if you have assigned the file to a server for deployment.

   **Note:** If you cannot see the value in the Assigned column or the full application path in the Location column, drag your cursor to the right to expand the Application Files window.

3. The right window pane displays the environments you have identified and optionally configured to accept project files.

   For a web server, you can create one or more subfolders on the deployment path. For example, you might separate HTML files from image files by creating an individual folder, or central location, for each type:
a. Select and right-click the current web server deployment path (for example, WEB/APPROOT/Sales).

b. Select New Folder from the shortcut menu. At the cursor location in the entry field, enter the name of the new folder.

**Note:** You must design the application properly in order to take advantage of a multi-level directory structure on the deployment environment.

4. In the right pane, open the environment on which you are partitioning the files. In the following example, the Test Environment includes a web server and a server node that can be used for deployment.

In the example:

- The web server is defined by the node WEB, followed by the descriptive name APPROOT and the name of the directory that will be created when the files are deployed (sales).

- A WebFOCUS Reporting Server is defined by the node EDA and the name of the server (for example, EDASERVE), followed by the descriptive name APPPATH and the name of the directory that will be created when the files are deployed (sales).

The developer selected NONE as the default partition environment when creating the new deployment scenario. For an example of the window that opens when you select a default partition environment, see *Working in a Default Partition Environment* on page 115.
5. Drag one or more files from the left window pane and drop them into the right window pane on the applicable server.

See *Guidelines for Deploying Files* on page 115 for information that will help you deploy files to the applicable server.

To remove a partitioned file from the right window pane, right-click it and select *Remove*.

In this example, the file named `launch_qty.htm` is an HTML page from which a user can run a report named `stockqty.fex`. To partition these files appropriately, you would drag `launch_qty.htm` to the web server, and `stockqty.fex` to the WebFOCUS Reporting Server.

On the toolbar, these buttons enable you to do the following while you remain in the Scenario Editor:

<table>
<thead>
<tr>
<th>Name</th>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy scenario</td>
<td><img src="deploy_icon.png" alt="Deploy" /></td>
<td>Deploys (moves) the files to the servers based on the partitioning defined in the current deployment scenario.</td>
</tr>
<tr>
<td>Deploy and run scenario</td>
<td><img src="deploy_run_icon.png" alt="Deploy And Run" /></td>
<td>Deploys the files to the servers and then runs the starting object defined in the current deployment scenario.</td>
</tr>
<tr>
<td>Run scenario</td>
<td><img src="run_icon.png" alt="Run" /></td>
<td>Runs the starting object defined in the current deployment scenario.</td>
</tr>
<tr>
<td>Displays all files in the project path</td>
<td><img src="display_all_icon.png" alt="Displays All Files" /></td>
<td>Displays available items in the project path. The icons next to some files are unavailable. This indicates that the files are available, but not active for your project. You can click the binoculars icon to show the files that you added to your project and hide the inactive files. However, you can deploy even the files that you did not add to your project.</td>
</tr>
</tbody>
</table>

These buttons are also available on the toolbar in the Explorer.

6. When you have finished partitioning files, choose one of the options in step 5 or close the window. You can perform any of the options in step 5 outside the Scenario Editor. See *Step 4: Deploy the Project Files* on page 119.
Guidelines for Deploying Files

- Typically, you will deploy HTML, GIF, JPG, VBScript, and JavaScript files to a web server. Executable WebFOCUS procedures (.fex, .mnt, and .sty files) will go to a WebFOCUS Reporting Server.

  If the deployed FOCUS procedure has a reference to a style sheet (.sty) file and the generated procedure uses a stylesheet different from the default, you must deploy the style sheet to the Reporting Server, or else the report will not run.

  To view and deploy the .sty file, you must add the filter for this file to the project level or an available virtual folder (for example, Procedures) to view files of this type in the project and in the Scenario Editor.

- Usually, you will not deploy Master Files (.mas) or Access Files (.acx). Master Files and Access Files may already reside on the deployment WebFOCUS Reporting Server or subserver in applications configured to contain them. If you deploy these types of files, you may overwrite production files with the same name on the deployment server.

- Proprietary FOCUS data sources (.foc) files cannot be deployed. If you try to drag an invalid file type to the Reporting Server, an error message appears.

- For files containing the command -HTMLFORM filename:

  - If you are deploying a procedure that contains -HTMLFORM filename (including procedures created with the HTML Composer), also deploy the referenced HTML file to the WebFOCUS Reporting Server.

    In this scenario, the procedure must be launched in order to execute the report.

  - When you use the HTML Composer to create reports with parameters, you must deploy the referenced HTML file containing the launch form to both the web server and the WebFOCUS Reporting Server.

    In this scenario, the HTML file must be launched from the web server.

Working in a Default Partition Environment

In this example, the developer selected the Test Environment as the default partition environment when creating the deployment scenario.
Step 3: Partition the Project Files

Developer Studio automatically partitions the .jpg and .htm files to the web server, and the .fex files to the WebFOCUS Reporting Server, as shown in the following image.

Files in the Name display field that have a check \( \checkmark \) icon next to them have been partitioned.

**Procedure: How to Modify Properties of Partitioned Files**

1. In the Scenario Editor, right-click a file that has been assigned to a server in the Available Servers window and select Properties, as shown in the following image.
2. Click the Advanced tab to see the following two options:

- Always deploy this file.

  or

- Never deploy this file.

3. Choose one of the options. They are unchecked by default.

**Always deploy this file.** Use this option when Smart Deploy is in effect to designate specific files to be deployed even though they have not changed since the last deployment. You can use this option for a file that has not changed, but references another file that has been assigned to a different server within the scenario.

For example, if Smart Deploy is enabled and a WebFOCUS Maintain procedure with forms has not changed since the last deployment, by default, it would not be redeployed. But if a new web server has been assigned for a web-based file that is used by one of the Maintain forms, the Maintain file (.mnt) with the form needs to be updated with the new URLs. In this case, you would select this option for the .mnt file, so that the form would be generated with the new web server information.

The following are examples of other situations when you might want to use this option:

- When script files are embedded in Maintain forms rather than linked, and changes are made to the script code, but not to the Maintain procedure and forms.

- When CALLed or EXECed procedures are assigned to different WebFOCUS servers since the last deployment, but the calling procedure has not changed.

**Never deploy this file.** Use this option when you do not want to deploy a file, but information about the assigned server location is needed by other files in the application.

In WebFOCUS Maintain applications, this option is useful when web-based files already reside on the target web server and you do not need to edit files there or do not have write access. The correct URLs will be used in the forms, without writing the web resource files to the web server.

If you are also using Smart Deploy, you need to assign the *Always deploy files* option to any .mnt files that have not changed but have associated forms that need to reference the new web server location.

**Setting Running Paths**

You can set the execution search path for a deployed application on a Reporting Server. The Reporting Server will use the path specified to locate the resources for that application.
This feature is useful if you set up common resources shared by multiple applications. It allows you to point a server to a common location to find the resources it needs to run a specific application.

**Procedure: How to Set Running Paths**

1. In the Scenario Editor, expand a WebFOCUS environment under Available Servers in the right pane. Then, select and right-click a WebFOCUS Environment.

   Select Set Server Paths from the shortcut menu. The Set Running Paths for Server dialog box opens, as shown in the following image.

   ![Set Running Paths for Server dialog box](image)

2. The Available Paths list box displays the application directories you can add to the execution search path for the current application.

   Select a directory or directories and click Add, or click Add All to include all directories in the execution search path.

   You can also use the Remove and Remove All buttons to make the desired assignments.

3. In the Additional Path field, optionally enter another path to add to the execution search path.

4. Click OK to close the dialog box.
Step 4: Deploy the Project Files

Developer Studio automates the process of moving the files to the target servers.

In addition to deploying files as described in this topic, you can deploy them in the Scenario Editor. See Step 3: Partition the Project Files on page 112. The buttons on the toolbar in the Scenario Editor are also available in the Explorer.

Procedure: How to Deploy Project Files in the Explorer Using the Shortcut Menu

1. Select the project. From the drop-down list on the toolbar, select the deployment scenario.
2. Right-click the project. Select Project Deployment from the shortcut menu. Then select:
   - **Deploy** to move the files to the servers based on the partitioning defined in the deployment scenario.
   - **Deploy and Run** to move the files to the servers and then run the starting object defined in the deployment scenario.
3. During deployment, a dialog box tracks the status. You can click Cancel to terminate the process.
Step 4: Deploy the Project Files

The following image shows the Deployment Progress window detailing successful deployment of HTML files to the web server and procedures to the WebFOCUS Reporting Server.

**Tip:** You can click the Details button if you do not want to view messages during the deployment process.

After deployment, the ACTION field indicates that one of the following occurred:

- Deployment Status: Succeeded, 0 warning(s), 0 error message(s).
- Deployment Status: Failed to complete. (The reasons for failure are shown, such as unresolved files.)
- Deployment Status: Cancelled by user.

During the deployment process, the File field shows the names of objects as they are deployed to a server.

The progress bar indicates how much of the process is complete, as it takes place.

The Status field displays the events that took place in the deployment process.
Reference: Scenario Editor

The following image shows the Scenario Editor.

Application Files

Displays the files in the selected project and their properties.

The value of Assigned is No if the file was not assigned to a server for deployment. Assigned is Yes if you assigned the file to a server for deployment.

Available Servers

Displays the WebFOCUS environments to which you can deploy files.
Step 4: Deploy the Project Files

Reference: Deploying Application Dialog Box

The following image shows the 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 Status: Succeeded, 0 warning(s), 0 error message(s).
  - Deployment Status: Failed to complete. (The reasons for failure are shown, such as unresolved files.)
  - Deployment Status: Cancelled by user.
- During the deployment process, the File field shows the names of objects as they are deployed to the server.
- The progress bar indicates how much of the process is complete, as it takes place.
- The Status field displays the events that took place in the deployment process.
Configuring the Target Servers (Optional)

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.

Follow the instructions in this topic to review or change the settings. You can use the WebFOCUS Administration Console and the Reporting Server Console to edit the files that contain the variables, or you can edit them manually.

You need read and write access to the web server and WebFOCUS Reporting Server in order to deploy files.

**Note:** The Application Root (APPROOT) directory of the WebFOCUS Client and Reporting Server do not need to point to the same physical location. You can install them on different servers or platforms.

**Reference:** APPROOT Variable for Setting the Deployment Path

The cgivars.wfs file contains the APPROOT variable for the path in which deployed files will reside on the web server. Typically, these files include HTML pages, graphic images, cascading style sheets, and JavaScript files.

For the client, the APPROOT variable points to the setting for IBI_Approot_Directory in the Administration Console. This setting is in the webconfig.xml file, which is located in \ibi\WebFOCUS81\config.

The cgivars.wfs file is located in:

<table>
<thead>
<tr>
<th>Type of Installation</th>
<th>Location of cgivars.wfs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Studio with localhost</td>
<td>install_drive:\ibi\DevStudio81\srv81\wfs\etc</td>
</tr>
<tr>
<td>WebFOCUS</td>
<td>install_drive:\ibi\WebFOCUS81\client\wfc\etc</td>
</tr>
</tbody>
</table>

Creating Reporting Applications With Developer Studio 123
The edaserve.cfg file contains the APPROOT variable for the path in which deployed files will reside on the WebFOCUS Reporting Server. Typically, these files include procedures, WebFOCUS StyleSheets, and customized HTML pages that require processing on the Reporting Server.

The edaserve.cfg file is located in:

<table>
<thead>
<tr>
<th>Type of Installation</th>
<th>Location of edaserve.cfg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Studio with localhost</td>
<td>\install_drive:\ibi\DevStudio81\srv81\wfs\bin</td>
</tr>
<tr>
<td>WebFOCUS</td>
<td>\install_drive:\ibi\WebFOCUS81\srv81\wfs\bin</td>
</tr>
</tbody>
</table>

**Procedure:** How to Set the Web Server Deployment Path Using the WebFOCUS Administration Console (Windows)

The APPROOT variable points to the setting for IBI_Approot_Directory in the Administration Console.

1. Access the WebFOCUS Administration Console from the Start menu, choose Programs, Information Builders, WebFOCUS 81 Developer Studio, WebFOCUS Developer Studio Utilities, and WebFOCUS Administration Console.

2. On the WebFOCUS Administration Console, click Configuration from the left pane and under Application Settings, click Application Directories.

3. Locate the IBI_Approot_Directory in the webconfig.xml file. Change the setting to a directory that exists on the web server.

   Ensure that this directory is also configured as a virtual directory on the web server, to allow users access to the application. For details, see APPROOT Alias on page 126 and How to Create an Alias on the Web Server (Windows) on page 127.

   When specifying the setting, use forward slashes (/) to separate directories, as shown in the default setting:

   `IBI_Approot_Directory=install_drive:/ibi/apps`

   An example is:

   `IBI_Approot_Directory=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 112.

4. Click Save and exit the WebFOCUS Administration Console.
Procedure: How to Set the WebFOCUS Reporting Server Deployment Path Using the Reporting Server Console (Windows)

1. Access the Reporting Server Console:
   - or
   - In Developer Studio, expand the applicable environment under WebFOCUS Environments. Expand Data Servers. Select the remote WebFOCUS Reporting Server. Click the WebFOCUS Reporting Server Console button on the toolbar.

2. From the menu bar, select Workspace.
3. In the navigation pane, expand the Configuration Files folder.
4. Right click Workspace - edaserve.cfg and select Edit.
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. Click Save and Restart Server and exit the Reporting Server Console.

Procedure: How to Set the Web Server and WebFOCUS Reporting Server Deployment Path Using a Text Editor (Windows and UNIX)

1. Open the webconfig.xml file in a text editor, such as Windows Notepad.
2. Locate the IBI_Approot_Directory variable. Change the setting to a directory that exists on the web server.
Ensure that this directory is also configured as a virtual directory on the web server, to allow users access to the application. For details, see *APPROOT Alias* on page 126 and *How to Create an Alias on the Web Server (Windows)* on page 127.

When specifying the setting, use forward slashes (/) to separate directories, as shown in the default setting:

```xml
<param-name>IBI_Approot_Directory</param-name>
<param-value>install_drive:/ibi/apps</param-value>
```

An example is:

```xml
<param-name>IBI_Approot_Directory</param-name>
<param-value>d:/ibi/apps</param-value>
```

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 112.

3. Save the changes and close webconfig.xml.
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 (AS/400)**

1. In the CGI_BIN/EXPORT(WEBEXPORT) directory, set the APPROOT variable to the APPNAME defined during installation of the WebFOCUS Reporting Server for AS/400. This value is the same one defined for APPROOT in the xxxBIN/CONFIG(EDASERVE) file.

2. In the web server HTTP configuration file, add the following lines for APPROOT:

   ```
   Pass /APPROOT/* /IBI/APPS/*
   Pass /approot/* /ibi/apps/*
   ```

**Reference: APPROOT Alias**

After you set the APPROOT variable, you may need to create an alias for it. The alias allows browsing of the application directory and read permission.
The following will help you determine if you need to create an alias:

- If you are using Internet Information Server (IIS), an alias is automatically created for you. During installation of Developer Studio or WebFOCUS, you can choose automatic configuration of IIS.
- The APPROOT alias must correspond to the APPROOT path in cgivars.wfs.

**Procedure:** How to Create an Alias on the Web Server (Windows)

1. From the Start menu, select **Settings** to open the Control Panel.
2. Select **Administrative Tools**.
3. Select **Internet Services Manager**. This selection opens a tool that allows you to manage 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 website.

**Note:** The Windows Challenge/Response is supported.
Configuring the Target Servers (Optional)
Developer Studio is a complete information control system with comprehensive features for retrieving and analyzing data that enables you to create reports quickly and easily. It provides facilities for creating highly complex reports, but its strength lies in the simplicity of the request language. You can begin with simple queries and progress to complex reports as you learn about additional facilities.

Developer Studio serves the needs of both end users with no formal training in data processing, and data processing professionals who need powerful tools for developing complete applications. A variety of tools are available that enable you to create reports and charts even if you do not know HTML or WebFOCUS reporting language commands and syntax.

In this chapter:

- Requirements for Creating a Report
- Report Types
- Reporting Tools
- Developing Your Report Request
- Customizing a Report

Requirements for Creating a Report

To create a report, only two things are required:

- **Data.** You need data from which to report. If the data is protected by an underlying security system, you may need permission to report from the data source. In addition, the server must be able to locate the data source. For more information on data source locations, see the Developing Reporting Applications manual.

You can report from many different types of data sources (with variations for different operating environments), including the following:

- Relational data sources, such as DB2, Teradata, Oracle, and Sybase.
- Hierarchical data sources, such as IMS and FOCUS.
Indexed data sources, such as ISAM and VSAM.

Network data sources, such as CA-IDMS.

Sequential data sources, both fixed-format and delimited format.

Multi-dimensional data sources, such as SAP BW and Essbase.

XML data sources.

For a complete list, see the Describing Data With Graphical Tools manual.

A data description. You need a Master File, which describes the data source from which you are reporting. The Master File is a map of the segments in the data source and all of the fields in each segment. For some types of data sources, the Master File is supplemented by an Access File. For more information on Master Files and Access Files, see the Describing Data With Graphical Tools manual.

By looking at the Master File, you can determine what fields are in the data source, what they are named, and how they are formatted. You can also determine how the segments in the data source relate to each other. Although you can create a very simple report without this information, knowing the structure of the data source enables you to generate creative and sophisticated reports.

You can supplement the information in the Master File by generating a picture of the data source structure (that is, of how the data source segments relate to each other). Use the following command:

```plaintext
CHECK FILE filename PICTURE RETRIEVE
```

In the picture, segments are shown in the order in which they are retrieved. Four fields of each segment appear.

Report Types

With Developer Studio, you can create the following basic report types using graphical tools:

- Tabular reports. Displays information in rows and columns. This is the basic report type, incorporating the fundamental reporting concepts. Most of the other report formats build on these concepts. You can display these reports in formats such as HTML, Excel®, and PDF.

- **Financial reports.** Specifically designed to handle the task of creating, calculating, and presenting financially oriented data, such as balance sheets, consolidations, and budgets. You can build these reports with the FML (Financial Modeling Language) Painter, a spreadsheet-like tool that enables you to define the content of the report on a row-by-row basis. This organization provides a number of advantages. You can:
  - Identify and display a title for each row of the report.
  - Perform row-based calculations and include the results at any point on the report.
  - Include the same record in multiple categories.
  - Include many types of formatting enhancements on a cell-by-cell basis.
  - Save individual rows and row titles in extract files.
  For details about the FML language, see the *Creating Financial Reports* manual.

- **Free-form reports.** Presents detailed information about a single record in a form-like context that is often used with letters and forms. If your goal is to present a detailed picture of one record per report page, you can use free-form reports to:
  - Position headers, footers, free text, and fields precisely on a page.
  - Customize your headers and footers by including fields as display variables.
  - Incorporate prefix operators in your headers and footers to perform calculations on the aggregated values of a single field.
  - Use vertical (By) sorting to put one or more report records on each page.

- **Charts.** Presents the same kinds of information as tabular reports, but in a wide variety of two-dimensional and three-dimensional chart types. You can create and customize charts using InfoAssist.

- **OLAP reports.** Allow users to drill down or roll up data hierarchies, pivot fields from columns to rows (or vice versa), and separate information by filtering or querying data sources based on specified criteria or thresholds.
  For details about OLAP reports, see the *Online Analytical Processing (OLAP)* manual.

- **Drill Through reports.** Allow users to create a PDF document that contains a summary report plus a detail report, where the detail report contains all the detail data for designated fields in the summary report. Clicking a Drill Through hyperlink navigates internally in the PDF file and no additional reports are run. Drill Through reports are static. You can save the PDF file to disk or distribute it using ReportCaster. When opened with Acrobat® Reader, it retains its full Drill Through functionality.
Excel Compound reports. Provides a way to generate multiple worksheet reports using the EXL2K output format. By default, each of the component reports from the compound report is placed in a new Excel worksheet. If the NOBREAK keyword is used, the next report follows the current report on the same worksheet.

For more information, see Linking a Report to Other Resources in the Creating Reports With WebFOCUS Language manual.

Excel Table of Contents reports. Provides a way to generate a multiple worksheet report where a separate worksheet is generated for each value of the first BY field in the WebFOCUS report.

For more information, see Choosing a Display Format in the Creating Reports With WebFOCUS Language manual.

Reporting Tools

Developer Studio provides powerful reporting tools for analyzing and formatting information, and building report requests.


The graphical reporting tool is integrated with two language-based tools, which can be used by power reporters, analytical reporters, administrators, and developers:

- Text Editor. Enables users to view the code generated by the graphical tools and to create procedures and other types of application files. The Text Editor can be accessed from the graphical tools and the graphical tools can be launched directly from the Text Editor.

  For details, see Editing Application Components as Text in Developer Studio on page 369.

- Command Console. Enables users to issue commands for monitoring messages and for debugging running procedures. For details, see the Developing Reporting Applications manual.

  In WebFOCUS, you can also use a third-party text editor such as Notepad. If you are developing an application directly on a remote server, you can use any editor supported by the operating system.

Developing Your Report Request

The only requirement for reporting is identifying a data source. Beyond that, the structure of a report request is very flexible and you only need to include the report elements you want. For example, you only need to include sorting instructions if you want your report to be sorted, or selection criteria if you want to report on a subset of your data.
The following are the most frequently used options for structuring a report request.

(Except where otherwise noted, see Report Painter Basics in the Creating Reports With Report Painter manual for details about these and many other reporting features.)

- **Specifying fields and columns.** Each column in your report represents a field. You can specify which fields you want to display, which fields you want to use to sort the report, which fields you want to use to select records, and which data source fields you want to use in creating temporary fields. Therefore, specifying the fields you want in a report is fundamentally tied to how you want to use those fields in your report.

- **Displaying data.** You can display data in your report by listing all the records for a field (detailed presentation), or by totaling the records for a field (summary presentation). You can also perform calculations and other operations on fields, such as finding the highest value of a field or calculating the average sum of squares of all the values of a field, and present the results of the operation in your report.

- **Sorting a report column.** Sorting a report enables you to organize column information. Developer Studio displays the sort field, which is the field that controls the sorting order, at the left of the report if you are sorting vertically, or at the top, if you are sorting horizontally. Sort fields appear when their values change. You can also choose not to display sort fields. You can sort information vertically, down a column, or horizontally, across a row. You can also combine vertical sorting and horizontal sorting to create a simple matrix.

- **Selecting records.** When you generate a report, you may not want to include every record. Selecting records enables you to define a subset of the data source based on your criteria and then report on that subset. Your selection criteria can be as simple or complex as you wish.

- **Showing subtotals and totals.** You can display column and row totals, grand totals, and section subtotals in your report.

- **Customizing the presentation.** A successful report depends upon the information presented and how it is presented. A report that identifies related groups of information and draws attention to important facts will be more effective than one that simply shows columns of data. For example, you can:
  - Give column titles more meaningful names.
  - Control the display of columns in your report.
  - Create headings and footings for different levels of the report (including each sort group, each page, and the entire report), and dynamically control the display of headings and footings based on conditions you set.
Add fonts, colors, grids, and images.

- Highlight a group of related information and separate it from other groups by inserting blank lines, underlines, and page breaks.

- **Creating temporary fields.** When you create a report, you are not limited to the fields that already exist in the data source. You can create temporary fields, deriving their values from real data source fields, and include them in your report.

  For details see, *Creating Temporary Fields* on page 273.

- **Joining data sources.** You can join two or more data sources to create a larger integrated data structure from which you can report in a single request.

  For details, see *Joining Data Sources* in the *Creating Reports With WebFOCUS Language* manual.

- **Storing and reusing the results.** You can store your report data as a data source against which you can make additional queries. This is especially helpful for creating a subset of your data source and for generating two-step reports. You can also format the new data source for use by other data processing tools, such as spreadsheets and word processors.

  For details, see *Saving and Reusing Report Output* in the *Creating Reports With WebFOCUS Language* manual.

You can run the request as an ad hoc query or save it as a procedure. Saving a report request as a procedure enables you to run or edit it at any time.

**Lowercase Directory Names and File Names in WebFOCUS From UNIX**

When working with WebFOCUS GUI tools that access directories and files from a UNIX system, the WebFOCUS Reporting Server returns lowercase directory names and files, by default. The WebFOCUS GUI tools also create directories and files in lowercase, regardless of the text case specified (for example, lowercase, uppercase, or mixed-case). If the user creates directories or files at the UNIX command level, they must create them in lowercase.

**Starting a Report Request**

A report request begins with the designation of a data source. You can then specify the details of your report request.

A report request can use a Master File that is stored in the application directory or in the WebFOCUS Reporting Server APP PATH or baseapp directory.

**Procedure:** How to Start a Report Request

1. To 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 following image shows the Add Procedure dialog box that opens.

2. Enter a name for the new procedure in the File name field.
3. Select the tool you want to use from the Create with drop-down list. The options are:

   - **Procedure Viewer** opens the Component Connector toolbox.
   - **Report Painter** enables you to create complex styled reports.
   - **Composer** enables you to design reports, and to coordinate and distribute layouts made up of multiple reports and graphs in a single output file.
   - **SQL Report Wizard** assists you with SQL passthru which allows you to execute SQL code that retrieves data from an RDBMS. You can use the resulting extract file in Report Painter or InfoAssist.
   - **InfoAssist** enables you to create a chart using an easy to use graphical tool.
   - **Text Editor** enables you to create a procedure with code.
   - **Define Function** enables you to create virtual fields.
4. Click Open.
If you select Report Painter, Composer, or Text Editor, the selected tool opens.

If you select Procedure Viewer, the Component Connector toolbox opens. Do the following:

a. Click a component connector (yellow diamond) at the point where you want to include the report in the procedure, then click the Report button on the component connector toolbar.

The Open dialog box opens.

b. Select the Master File you want to report against.

c. Click Open.

The tool you selected opens.

Creating a Report Example

The example in this topic is a simple report request that illustrates some of the basic functions of Developer Studio. However, there are many more functions not shown here that you can find information on throughout this documentation.
Example: Creating a Simple Report

The following annotated example illustrates some of the basic functions of Developer Studio. The numbered explanation in this example corresponds with the code in this request. This request can be generated using Developer Studio graphical tools, or by typing the commands into a text editor.

1. JOIN PIN IN EMPDATA TO ALL PIN IN TRAINING AS J1
2. DEFINE FILE EMPDATA
   YEAR/YY=COURSESTART;
3. END

4. TABLE FILE EMPDATA
5. HEADING CENTER
   "Education Cost vs. Salary"
6. SUM EXPENSES AS 'Education,Cost' SALARY AS 'Current,Salary'
7. AND COMPUTE PERCENT/D8.2=EXPENSES/SALARY * 100; AS 'Percent'
8. BY DIV
    BY DEPT
9. WHERE YEAR EQ 1991
10. ON TABLE SUMMARIZE
11. ON TABLE SET STYLE *
    TYPE=HEADING, STYLE=BOLD, COLOR=BLUE,
    TYPE=REPORT, FONT=TIMES, SIZE=8,
    TYPE=REPORT, GRID=OFF,
    ENDSTYLE
12. END

The output is:

<table>
<thead>
<tr>
<th>DIV</th>
<th>DEPT</th>
<th>Education</th>
<th>Current</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>ADMIN SERVICES</td>
<td>1,250.00</td>
<td>$25,400.00</td>
<td>4.92</td>
</tr>
<tr>
<td></td>
<td>PROGRAMMING &amp; DVLPM</td>
<td>1,580.00</td>
<td>$40,900.00</td>
<td>3.86</td>
</tr>
<tr>
<td>CORP</td>
<td>ACCOUNTING</td>
<td>2,050.00</td>
<td>$62,500.00</td>
<td>3.28</td>
</tr>
<tr>
<td>NE</td>
<td>CUSTOMER SUPPORT</td>
<td>1,800.00</td>
<td>$19,300.00</td>
<td>9.33</td>
</tr>
<tr>
<td></td>
<td>MARKETING</td>
<td>3,100.00</td>
<td>$32,300.00</td>
<td>9.60</td>
</tr>
<tr>
<td></td>
<td>SALES</td>
<td>1,800.00</td>
<td>$43,600.00</td>
<td>4.13</td>
</tr>
<tr>
<td>SE</td>
<td>CONSULTING</td>
<td>3,350.00</td>
<td>$35,900.00</td>
<td>9.33</td>
</tr>
<tr>
<td>WE</td>
<td>MARKETING</td>
<td>9,850.00</td>
<td>$102,200.00</td>
<td>9.64</td>
</tr>
<tr>
<td></td>
<td>PROGRAMMING &amp; DVLPM</td>
<td>5,310.00</td>
<td>$42,900.00</td>
<td>12.38</td>
</tr>
<tr>
<td></td>
<td>SALES</td>
<td>4,500.00</td>
<td>$100,500.00</td>
<td>4.48</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>34,590.00</td>
<td>$505,500.00</td>
<td>6.84</td>
</tr>
</tbody>
</table>
The request processes in the following way:

1. The JOIN command joins the EMPDATA and TRAINING data sources, allowing the request to access information from both data sources as if it were a single structure.

2. The DEFINE command creates a virtual field which extracts the year from the COURSESTART field in the TRAINING data source.

3. The END command ends the DEFINE command.

4. The TABLE command begins the report request.

5. The HEADING command adds the heading, Education Cost vs. Salary to the report output.

6. The SUM command adds the values within both the EXPENSES field and the SALARY field. The AS phrase changes the name of the column headings.

7. The COMPUTE command creates a calculated value using the values that have been aggregated in the SUM command and sorted with the BY command.

8. The BY phrase sorts the data in the report by the DIV field, and then by the DEPT field.

9. The WHERE command includes only the data that falls in the year 1991.

10. The ON TABLE SUMMARIZE command adds all values in both the EXPENSES and SALARY columns, and recalculates the Percent column.

11. The StyleSheet information formats the report heading and content.

12. The END command ends the report request.

**Saving a Report**

When you exit a reporting tool, you are prompted to save the report you have created. For details, see *Report Painter Basics* in the *Creating Reports With Report Painter* manual for additional save options available in that tool.

When you save a procedure created in Report Painter or the Component Connector toolbox, the entire procedure is saved, not only the report component.

**Procedure: How to Save a Report**

Close the reporting tool and choose Yes to save the changes.

or

Select Save from the File menu. Your report is saved as a component in the procedure in which it was created.

If you add other components to a procedure or edit the report component, you will be prompted to update the procedure (FOCEXEC). Click Yes to save your changes.
Customizing a Report

A successful report depends on the information presented and how it is presented. A report that identifies related groups of information and draws attention to important facts will be more effective than one that simply shows columns of data.

When you have selected the data that is going to be included in your report and how you want it to appear, you can then continue developing your report with custom formatting. There are many things you can add to your request in order to make your report more effective. You can:

- Add titles, headings, and footings. You can also change column titles with the AS phrase, and create headings and footings for different levels of the report (including each sort group, each page, and the entire report).
- Change the format of a field and the justification of a column title.
- Determine the width of a report column.
- Dynamically control the display of subtotals, headings, and footings based on conditions you define.
- Highlight a group of related information and separate it from other groups by inserting blank lines or underlines between each group.
- Emphasize data using color to highlight certain values in your report based on conditions you define.
- Format your report using external cascading style sheets.
- Add drill-down capability to your report. This adds extra value by linking your report to other reports or URLs that provide more detail.
If a report requires data from two or more related data sources, you can temporarily join the files and report from them as if they were one. Joined files remain physically separate, but are treated as one data source structure throughout the session or until you clear the Join.

You can also create a new file structure for reporting by joining existing synonyms of relational tables using the Cluster Joins editor.

**In this chapter:**

- What Is a Join?
- Types of Joins
- Joining Different Kinds of Data Sources
- Creating a Join With Graphical Tools
- Creating Multi-Field Joins
- Using Conditional Joins
- Creating Cluster Joins
- Customizing a Join
- Working With Joins
- Reporting With a Join
- Saving a Join
- Merging Data Sources Using the Match Wizard

### What Is a Join?

A Join is a temporary connection between two or more data source files that share at least one common field. For example, since the EMPLOYEE data source includes a job code field and the JOBFILE data source also includes a job code field, you can join the two data sources.

The field format and the type must be the same for the common fields, but the field name need not be the same.
After you join two files, each time a record is retrieved from the first file (host file), it also retrieves the matching records from the second file (cross-referenced file).

The following diagram illustrates how Developer Studio can associate each employee in the EMPLOYEE data source with a job description in the JOBFILE data source.

![Diagram showing how Developer Studio associates employees with job descriptions]

The data source on the left is the host file. The data source on the right is the cross-referenced file.

**Types of Joins**

The following section describes the type of joins you can create using the Join tool.

**Dynamic Joins**

Dynamic joins allow you to join two or more related data sources and create a virtual structure for reporting. The data sources remain physically separate, but Developer Studio treats them as a single structure. Dynamic joins include:

- Conditional joins
- Single instance joins
- Multiple instance Joins
Left outer joins

You can use dynamic joins for the duration of your session or until you clear the Join. Dynamic joins can read data from the joined data sources using the TABLE and GRAPH facilities.

Conditional Joins

Conditional joins use WHERE-based syntax to specify joins based on conditional criteria, not just on equality between fields. Additionally, the host and cross-referenced join fields do not have to contain matching formats.

Suppose you have a data source that lists employees by their ID number (the host file) and another data source that lists training courses and the employees who attended those courses (the cross-referenced file). Using a conditional join, you can join employee ID in the host file to employee ID in the cross-referenced file to determine which employees took training courses in a given date range (the WHERE condition). For information on how to create conditional joins, see Using Conditional Joins on page 156.

Single Instance and Multiple Instance Joins

The Join tool allows you to create single instance and multiple instance joins. A single instance, or one-to-one join structure matches one value in the host data source to one value in the cross-referenced data source. Joining an employee ID in an employee data source to an employee ID in a salary data source is an example of a single instance join.

A multiple instance or one-to-many join structure matches one value in the host data source to multiple values in the cross-referenced field. Joining employee ID in the employee data source of a company to employee ID in a data source that lists all the training classes offered by that company would result in a listing of all courses taken by each employee, or a joining of the one instance of each ID in the host file to the multiple instances of that ID in the cross-referenced file.

When you create a dynamic or conditional join using the Join tool, you can specify whether the join is a single instance or a multiple instance.

Reference: Descriptions of Single Instance and Multiple Instance Joins

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single instance (one-to-one)</td>
<td>Each host record has at most, one matching record in the cross-referenced file.</td>
<td>If you join EMPLOYEE to JOBFILE, each employee can have only one job code.</td>
</tr>
</tbody>
</table>
### Single Instance and Multiple Instance Joins

The following diagrams illustrate single instance and multiple instance Joins.

**Single Instance (Unique) Join**

![Single Instance (Unique) Join Diagram]

**Multiple Instance (Non-Unique) Join**

![Multiple Instance (Non-Unique) Join Diagram]

### Joins Based on Virtual Fields

Joins that are based on virtual fields use DEFINE-based syntax to create a virtual field in the host file that you can then join to a real cross-referenced field. The DEFINE expression that creates the virtual host field may contain only fields in the host file and constants. It may not contain fields in the cross-referenced file.
You can join a virtual field to more than one real field in more than one cross-referenced file using separate Join commands, each referring to the same host file. For information on how to create Joins based on virtual fields, see Using a Virtual Field as the Join Field on page 167.

**Left Outer Joins**

When a report omits host rows that lack corresponding cross-referenced rows, the join is called an *inner join*. When a report displays all matching rows plus all rows from the host file that lack corresponding cross-referenced rows, the join is called a *left outer join*.

See How to Create a Left Outer Join With the Join Tool in Developer Studio on page 152. For details on left outer join syntax, see the Creating Reports With WebFOCUS Language manual.

**Note:** If the SET ALL=ON or SET ALL=PASS setting is applied, it impacts all joins in the procedure and they are treated as left outer joins.

**Host Records With No Matching Cross-Referenced Records**

Sometimes there is no matching record in the cross-referenced file. A host record that has no matching cross-referenced record is called a *short path*.

The report that results when a host record lacks a corresponding cross-referenced record depends on the following factors:

- **The type of Join.**
  
  With a multiple instance Join, the value of the ALL parameter controls short path behavior. The default value, OFF, eliminates all host records that lack a matching cross-referenced record. Use the Set tool to change the ALL value, if necessary. For more information about the Set tool, see Customizing Your Environment in the Developing Reporting Applications manual.

  With a single instance Join, a missing cross-referenced record never causes Developer Studio to eliminate a host record from a report. Instead, default values (zero for numeric fields and blank for alphanumeric fields) are supplied for missing cross-referenced fields.

- **The types of data sources participating in the Join.**

  Since relational data source engines can join relational data sources efficiently, Developer Studio passes this task to them whenever possible.
Joining Different Kinds of Data Sources

In addition to joining FOCUS data sources, you can join other kinds of data sources. For example, you can join a FOCUS data source to a DB2 table. The fact that the data sources are from different systems is transparent to you as long as you have the appropriate data adapter installed for each non-FOCUS data source you want to access.

You must consider the following when joining files:

- You cannot join files protected by the DBA (Database Administration) security feature to unprotected (non-DBA) files. You can, however, join files protected by different DBA passwords. See Creating Synonyms in the Describing Data With Graphical Tools manual.
- You cannot join comma-delimited files.
- In addition to these considerations, the Join field from the cross-referenced file may have to satisfy other conditions, as described in Data Source Requirements for Cross-Referenced Fields on page 146.

Data Source Requirements for Cross-Referenced Fields

Developer Studio supports a wide variety of data sources. In most cases, you can join any data source to any other data source as long they share at least one common field. This is sometimes called a universal Join.

In general, the host field in a Join can be any field. However, some fields may not be valid as cross-referenced fields.

Certain data sources impose additional restrictions on cross-referenced fields. For example:

- Some data sources require the cross-referenced field to have an index. An index in a data source is similar to an index in a book. Just as you look up a topic in an index to find the correct page without reading the entire book, the database system looks up a field value in its index so it can go to the correct location without reading the entire data source.
- Some data sources require the cross-referenced field to be a key. A key is a field (or group of fields) that distinguishes one record from another. In the EMPLOYEE data source, the employee ID is the key. Its value identifies a particular employee record.

For details, see Joining Cross-Referenced Fields in Specific Types of Data Sources on page 146.

Reference: Joining Cross-Referenced Fields in Specific Types of Data Sources

The following table describes conditions that apply when you join cross-referenced fields in specific types of data sources.
### Conditions for JoinType of File

<table>
<thead>
<tr>
<th>Type of File</th>
<th>Conditions for Join</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOCUS</td>
<td>The cross-referenced field must be indexed.</td>
</tr>
<tr>
<td>Relational (DB2, Oracle, Microsoft SQL, Sybase Server, Informix, MS SQL OLAP Services, Microsoft Access, Hyperion ESSBase, ODBC, EDA)</td>
<td>Any field can be the cross-referenced field.</td>
</tr>
<tr>
<td>Fixed sequential</td>
<td>Any field can be the cross-referenced field, but it must be sorted the same way as the host field. If the cross-referenced file contains only one segment, the host file must have a segment declaration.</td>
</tr>
</tbody>
</table>

### Joining a Data Source to Itself: The Recursive Join

There may be situations in which you need to join a data source to itself. As an example, consider a data source that lists parts for an airplane. Some parts, such as cabins, are made up of smaller parts, such as seats. Therefore, this data source has:

- A root segment with an instance for every part, including its description.
- A descendant segment connecting a part to each of its subparts.

Certain subparts may themselves be composed of smaller parts. If you join the subpart segment to the root segment, you can go from a subpart record to its description record in the root segment to its component part records, as shown in the following diagram.
Creating a Join With Graphical Tools

The Join tool provides a graphical method for creating and manipulating Joins. You must specify a host file, then a cross-referenced file to create a join.

You create the Join as an object, separate from any other object or procedure. You are responsible for placing the Join object in the correct position within your procedure (that is, before a Define, Report, or Graph), and for running it. The Join remains active for the entire session, unless you explicitly clear it.

You can open the Join tool from either the toolbar or from a reporting tool. Before the tool opens, you will be asked to select a host file.

The Join tool displays both files and, ordinarily, a default Join, as shown in the following image.

- A Join is represented by a connector line from the host Join field to the cross-referenced Join field. A single arrow indicates a single instance Join. A double arrow indicates a multiple instance Join, the default Join type.

- Developer Studio automatically creates a Join between the files if they both have fields with the same formats (and, for a FOCUS file, if the cross-referenced field is indexed). You can use the Toggle Auto Join feature to perform automatic joins within tables. You can use the default Join as it is or you can customize the Join. If multiple fields have the appropriate format, the default Join is not created unless one of these fields has the same name as the host field. In this case, you must specify a join explicitly.
A host file is indicated by a yellow caption. A cross-referenced file is indicated by a blue caption.

Indexed fields have a key icon in front of the field, and virtual fields have a define icon in front of the field. A virtual field is a temporary field that does not actually exist in the data source. Its value is computed based on an expression you write. After it is created, the virtual field is treated as if it were a field in the data source.

By default, the list of field names in each data source displays in the Join window. You can change the display to show the following additional information about each field, if it is available in the Master File: alias, format, title, and description.

**Reference:** Join Toolbar

When you access the Join tool, a toolbar opens below the title bar. You can access the following commands from the toolbar:

<table>
<thead>
<tr>
<th>Button</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Save" /></td>
<td>Save</td>
<td>Saves the join.</td>
</tr>
<tr>
<td><img src="image" alt="New Join" /></td>
<td>New Join</td>
<td>Creates additional concurrent joins.</td>
</tr>
<tr>
<td><img src="image" alt="Add File" /></td>
<td>Add File</td>
<td>Selects a cross-referenced file to join to the host file you have already selected.</td>
</tr>
<tr>
<td><img src="image" alt="Clear Join" /></td>
<td>Clear Join</td>
<td>Launches the Join Clear window.</td>
</tr>
</tbody>
</table>

**Note:** You may also select New Join from the right-click context menu in the Join window.

**Note:** You may also select Add File from the right-click context menu in the Join window.

**Note:** You may also select Clear Join from the right-click context menu in the Join window.
Creating a Join With Graphical Tools

<table>
<thead>
<tr>
<th>Button</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Run" /></td>
<td>Run Join</td>
<td>Runs a join to check syntax.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> You may also select Run from the right-click context menu in the Join window.</td>
</tr>
<tr>
<td><img src="image" alt="Delete" /></td>
<td>Delete</td>
<td>Deletes a selected object (table or join).</td>
</tr>
<tr>
<td><img src="image" alt="Define Field" /></td>
<td>Define Field</td>
<td>Launches the Define dialog box to create a virtual field in the host file.</td>
</tr>
<tr>
<td><img src="image" alt="Create/Edit Selection" /></td>
<td>Create/Edit Selection</td>
<td>Launches the WHERE Expression Builder to create WHERE criteria and change the join type to a conditional join. This is enabled only when the host table is selected.</td>
</tr>
<tr>
<td><img src="image" alt="Show/Hide Source Code" /></td>
<td>Show/Hide Source Code</td>
<td>Shows or hides the FOCUS code generated by the Join tool. This is enabled only when the Join link is selected.</td>
</tr>
<tr>
<td><img src="image" alt="Toggle Auto Join" /></td>
<td>Toggle Auto Join</td>
<td>Performs automatic joins within tables.</td>
</tr>
<tr>
<td><img src="image" alt="Cascade" /></td>
<td>Cascade</td>
<td>Creates a cascade view of multiple joins (default).</td>
</tr>
<tr>
<td><img src="image" alt="Horizontal Tile" /></td>
<td>Horizontal Tile</td>
<td>Splits the Join tool horizontally when working with multiple joins.</td>
</tr>
<tr>
<td><img src="image" alt="Vertical Tile" /></td>
<td>Vertical Tile</td>
<td>Splits the Join tool vertically when working with multiple joins.</td>
</tr>
</tbody>
</table>

**Procedure: How to Create a Join**

If you want to define a Join in a procedure:

1. Right-click the procedure and choose *Edit in Developer Studio Tool* from the context menu. The Procedure Viewer opens.
2. Click the component connector (yellow diamond) at the point where you want to include the join in the procedure, then click the *Join* button on the component connector toolbar.
3. Choose a Master File for the data source you wish to use as your host file and click Open. The Join tool opens showing a Fields window for the host file that you selected.

4. To add a cross-referenced data source to the Join, click the Add File button on the Join toolbar or choose File from the Developer Studio Insert menu.

**Tip:** You may also right-click anywhere in the Join window and select Add File from the context menu.

5. Select a cross-referenced file.

6. Click OK.

A default Join is created if the host and cross-referenced files share appropriate fields. If a default Join is not created, follow the instructions in *Customizing a Join* on page 165 to create your own.

**Procedure:** How to Create a Join for a Reporting Object in Developer Studio

**Note:** This procedure applies to creating a Join in the Managed Reporting environment.

1. In the Reporting Object window, double-click Join. The WebFOCUS Table List dialog box opens.

2. Choose a Master File for the data source you wish to use as your host file, and click OK.

   The Join tool opens showing a Fields window for the host file that you selected.

3. To add a cross-referenced data source to the Join, click the Add File button.

   The WebFOCUS Table List opens again.

4. Select a cross-referenced file.

5. Click OK.

A default Join is created if the host and cross-referenced files share appropriate fields. If a default Join is not created, follow the instructions in *Customizing a Join* on page 165 to create your own.
Procedure: How to Create a Left Outer Join With the Join Tool in Developer Studio

1. To open the Join Properties dialog box, do one of the following:
   - In the Join tool, right-click a join link and select *Left Outer Join*.
   - In the Explorer, right-click a join link and select *Properties*.
   - Double-click a join link in the Explorer.

   The Join Properties dialog box opens.

2. Select the *Left Outer Join* option button.
3. Click OK.

Procedure: How to Add Information About Fields to Join Field Lists

The Field list for each data source being joined displays the field names by default. To help you build meaningful joins you can add information about the listed fields.
1. With the Join tool open, right-click within a table to view available field options. To change information displayed for tables globally, select Options from the Window menu. Click the Reporting tab and select the appropriate options from the Field List Options area.

2. Select any or all of the following items: Qualified Names, Show Name, Show Alias, Show Title, Show Description, Show Format. A check mark displays next to your selections. Name and Descriptions are checked by default.

   The selected information categories appear as headings in the field list for each joined file. Details will appear if the corresponding information is included in the Master File.

3. Expand the window to see the additional field list columns. Click the column titles if you want to change the sort order.

Creating Multi-Field Joins

In some cases, you may want to join more than one host field to a single cross-referenced field. The procedure, as well as the relationship, differs for joins between two relational data sources, joins between two FOCUS data sources, and joins between relational and FOCUS data sources.

There are two types of multi-field joins:

- Multi-field joins in which you join multiple host fields to multiple cross-referenced fields.
- Concatenated joins in which the field resulting from the concatenation of two host fields joined to a single cross-referenced field.

The following table lists the types of joins allowed between files:

<table>
<thead>
<tr>
<th>Host</th>
<th>Cross-Referenced</th>
<th>Supported Multi-Field Join Constructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL*</td>
<td>SQL</td>
<td>You can join two SQL files with a multi-field construction. You cannot directly join two SQL files with a concatenated join. You can, however, use a Define-based join to achieve the same result that a concatenated join would produce between two SQL files. For more information on using a Define-based join, see Customizing a Join on page 165.</td>
</tr>
<tr>
<td>FOCUS</td>
<td>FOCUS</td>
<td>You can use only the concatenated join construction to join two FOCUS host fields to a single FOCUS cross-referenced field.</td>
</tr>
</tbody>
</table>
You can use only the concatenated join construction to join two SQL host fields to a single FOCUS cross-referenced field.

You can join a FOCUS file and an SQL file with multi-field join construction. You cannot join a FOCUS file and an SQL file with a concatenated join. You can, however, use a Define-based join to achieve the same result that a concatenated join would produce between these files. For more information on using a Define-based join, see Customizing a Join on page 165.

* SQL represents all relational data sources.

**Example: Creating a Multi-Field Join for Relational Data Sources**

This example uses two relational data sources: ECOURSE, the host file, and COURSE, the cross-referenced file. ECOURSE contains the fields COURSE_NAME and EMP_ID, while COURSE also contains the fields COURSE_NAME and EMP_ID. These fields have the same format.
To join the COURSE_NAME and EMP_ID host fields to their corresponding cross-referenced fields, you would join each host field separately to the cross-referenced field. See *Creating a Join With Graphical Tools* on page 148 for the procedure to join a host field to a cross-referenced field. This procedure results in the following join structure.

This join construction represents one join. Developer Studio retrieves matching records across this join only when:

- COURSE_NAME in the file ECOURSE equals COURSE_NAME in the file COURSE.
- and
- EMP_ID in the file ECOURSE equals EMP_ID in the file COURSE.

**Creating a Concatenated Join**

You can use the concatenated join construction between two FOCUS files and between a host relational file and a cross-referenced FOCUS file. This is the only type of multi-field join construction you can use between these files.

In a concatenated join, first the two host fields are combined, then the resulting field is joined to the cross-referenced field.

**Example: Creating a Concatenated Join for FOCUS Data Sources**

This example uses two FOCUS files: EMPLOYEE, the host file, and HOBBIES, the cross-referenced file. EMPLOYEE contains the fields FIRST_NAME and LAST_NAME, and HOBBIES contains the field FULL_NAME.

To join the host fields to the cross-referenced field:

1. Click *FIRST_NAME* and then click *LAST_NAME* in the EMPLOYEE file.
Both field names are highlighted.

2. Click and hold the FIRST_NAME field.

3. Drag the mouse until you position the Join icon over the FULL_NAME field in the cross-referenced file. The mouse cursor changes when you drop or link a field.

4. Release the mouse button.

The concatenated host field is joined to the cross-referenced field FULL_NAME. Developer Studio graphically represents this join, as shown in the following image.

![Image showing the join between FIRST_NAME and FULL_NAME fields]

The fields FIRST_NAME and LAST_NAME are not joined to FULL_NAME, rather FOCUS joins the virtual field created from the concatenation of FIRST_NAME and LAST_NAME to FULL_NAME.

**Using Conditional Joins**

Using conditional joins, you can establish joins based on conditions other than equality between fields. In addition, the host and cross-referenced join fields do not have to contain matching formats and the cross-referenced field does not have to be indexed.

The conditional join is supported for FOCUS and all relational data adapters. Because each data source differs in its ability to handle complex conditional criteria, the optimization of the WHERE syntax differs depending on the specific data sources involved in the join and the complexity of the conditional criteria.

To display a list of joined data sources, select the following query subject in the Query tool or issue the command from the Command Console:

```bash
? JOIN
```

This displays every join currently in effect and indicates any that are based on conditional criteria.
Procedure: How to Create Conditional Joins

1. Click the Add File button and select the tables you want to use.
2. Drag common fields between the tables to establish a join.
3. Select the join (the connector line) and click the Create/Edit Selection icon from the toolbar. The WHERE Expression Builder appears, as shown in the following image.

![Where Expression Builder Image]

4. Click OK to return to the Join tool after you have specified your WHERE criteria.
   
   If you need to change the join type or join name, double-click the conditional join to launch the Join Properties window.

Creating Cluster Joins

Cluster joins enable you to create a new file structure by linking existing synonyms of two or more relational tables using the same or mixed data sources. For example, you may join a DB2 table and an Oracle table, and so on. Use cluster joins to create new views in the metadata by linking together physical tables and easily report against the new view or structure. You can create cluster joins by using the Modeling View of the Synonym Editor.
The Master File that is created combines the fields of the joined tables within a single file. The Access File from the combined file contains information about the actual location of the data sources and the Join information. It also shows how the tables are linked.

The total number of tables that you can add to the tool is 1024 (using 1023 joins), which results in a new Master File that has a maximum of 1024 segments.

The Cluster Join tool enables you to create a Star Schema which consists of a fact table referencing a number of dimension tables. Optionally, you can also create a view that has more than one fact table.

**Procedure: How to Create a Cluster Join by Enhancing Existing Synonyms**

Use the Modeling View to enhance an existing synonym by adding a segment.

1. From the Projects or Data Servers area, open a synonym by double-clicking a Master File from the Master Files folder.

   The Synonym Editor opens.

2. Click the *Modeling View* tab.

   The Synonym Editor Modeling View tab opens in the workspace.

3. Right-click a segment in the workspace and select *Insert*.

   The following context menu appears, as shown in the following image.

   ![Insert Context Menu](image)

4. Insert tables (segments) through one of the methods listed:

   **To insert a segment from an existing synonym:**
   
   a. Select *Insert*, then *Reference to Existing Synonym*. 

The Insert Reference to Existing Synonym dialog box opens, as shown in the following image.

b. Select a synonym to be inserted and click Select.

**Note:** Use this method if you are creating a cluster join with an existing table or synonym.

**Tip:** Click Save As from the Modeling View File menu if you do not want to modify the original synonym.

**To insert a segment from an existing synonym with Snowflake:**

Select Insert, then *Reference to Existing Synonym with Snowflake*.

The procedure is the same as Reference to Existing Synonym, except the selection list will show both candidate synonyms and any synonyms that they reference.
To insert a segment via Metadata Import:

a. Select Insert, then Segment via Metadata Import. This enables you to add segments by using the Create Synonym tool. This tool creates a synonym and includes it as a segment in the synonym from which the tool was launched.

Note: Use this method if you are creating a cluster join and need to use a synonym that does not exist. This option enables you to create the synonym and continue to create the cluster join.

b. When this option is selected, you are first presented with the Adapter dialog box where you can select a configured adapter connection to continue or configure a new adapter, if necessary. The Adapter dialog box is shown in the following image.
The following image is an example of a screen that appears where you provide information for the connection parameters.

The following image is an example of a screen that enables you to select tables to create synonyms.
The selected synonyms will be created and added to the Modeling View.

**To insert a segment manually:**

Select *Insert*, then *Segment Manually*. This enables you to assign values to segment attribute fields in the Synonym Editor.

*Note:* Use this method if you are coding a new Master File, as you would for a FOCUS data source.

The segment is added in the Modeling View.
5. Right-click a segment and select *Join Properties*, as shown in the following image.

**Note:** This option is only available when using relational tables.
The Join Editor dialog box opens, as shown in the following image.

6. Select a radio button for One-to-Many (Join All) or One-to-One (Join Unique).

- **One-to-Many.** Indicates a multiple-instance join. At run time, each host record can have many matching records in the cross-referenced file.

- **One-to-One.** Indicates a single-instance join. At run time, each host record has, at most, one matching record in the cross-referenced file.

7. Select a **Join Type** from the drop down menu. The choices are Inner, Left Outer, or Cross Join.

The Join Condition field automatically creates a Join if identical fields exist in both segments.

**Procedure: How to Create a Cluster Join Using a New Synonym**

Another way to create a cluster join is to start with an empty synonym:
1. Right-click a Master Files folder, select New, then Synonym via Synonym Editor. The New Master File dialog box opens.

2. Enter a unique file name in the File name field.

3. Click Open. The Synonym Editor opens.

4. Click the Modeling View tab.

5. Right-click in the workspace and select from one of the available options to start building the new view.

Customizing a Join

You can customize an existing Join by changing the files or fields involved. If no default Join was created, you can use the same technique to specify a connection explicitly.

You can also change the Join name, make the Join a single instance, use a virtual field as the Join field, and add tag names to the Join files.

For related information, see Working With Joins on page 171.

Procedure: How to Remove a File From the Join

1. Click the table of the file you want to remove.

2. Click the Delete button on the toolbar or press the Delete key. A confirmation dialog box appears.

If you delete the host file, all cross-referenced files are automatically deleted.
**Procedure:** How to Delete a Join

1. Right-click the Join connector line.
2. Select *Delete* from the context menu.

You can also select the Join connector line and click the *Delete* button on the toolbar, or press the *Delete* key.

The files remain in the window with no connector line between them.

**Procedure:** How to Specify Join Fields

To create your own Join (after deleting the existing Join, if necessary):

1. Select a field in the host file panel.
2. Click and hold the selected field.
3. Drag the selected field from the host file panel onto the matching field located in the cross-referenced file panel. The cursor changes and displays the appropriate icon when a join is possible.
4. Release the mouse button.

A Join connector line appears.

**Reference:** Usage Notes for Specifying Join Fields

- You can multi-select fields in the host file to join to a single field in the cross-referenced file when both are FOCUS files. For other file types (for example, EDA, Oracle, DB2, and so on) you must join a host field to a cross-referenced field, one pair at a time.

- For non-FOCUS joins, you can join a maximum of 16 fields from each file.

**Procedure:** How to Create Additional Joins

You can leave the Join window open and create another Join by:

- Clicking the *New Join* button from the Join toolbar.

- Selecting *New Join* from the Developer Studio File menu.

- Selecting *New Join* from the right-click context menu in the Join window.

You can define up to 63 concurrent Joins that will have a maximum of 64 segments in the new join structure.
**Procedure:** How to Make a Join a Single Instance

By default, each Join is a multiple instance.

To make the Join a single instance:
1. Right-click the Join connector line.
2. Select **Single Instance** from the context menu.

**Changing the Join Name**

A default name is automatically assigned to each Join.

You can use the Join name to later clear the Join and to prevent another Join from overwriting it.
If you remove the Join name completely, any subsequent unnamed Join will overwrite this Join.

**Procedure:** How to Change or Remove the Default Join Name

1. Right-click the Join connector line and select **Join Name** from the context menu. The Join Name dialog box displays the default Join name.
2. Edit or delete the name in the Join Name box and click **OK**.

You can also provide a description. This is not used in the Join command. It is used for reference purposes. A comment (-*) is added to the procedure. For example, -*JOINDESC J0.

**Note:** The following words cannot be used as a Join name: ALL, AND, AS, IN, JOIN, TAG, TO, WITH.

**Procedure:** How to Display Current Join Structures

To display a list of joined data sources, issue the command from the Command Console:

```
? JOIN
```

This displays every Join currently in effect.

**Using a Virtual Field as the Join Field**

A virtual field is a temporary field that you describe in a Master File or procedure. The value of the virtual field is calculated and treated as part of the data source.

Consider the following factors before making a virtual field the host field in a Join:

- If you create the virtual field in a procedure rather than in the Master File, you must make sure the virtual field runs after the Join runs because a Join clears all existing virtual fields.
If there is a Join connector line, you must delete the line before creating the Join on the virtual field. You cannot join multiple fields with this option. The virtual field must be the only Join field.

**Procedure: How to Use a Virtual Field as the Join Field**

1. Click on the caption in the Fields window of the host file to make it active.
2. Click the Define Field button on the toolbar.
   
   The Join Define in File dialog box opens.

   ![Join Define in File dialog box]

3. You must associate the virtual field with a segment of the data source:
   
   **a.** Enter the name of the virtual field in the Define Field box.
   
   **b.** In the With Field drop-down box, choose the name of any real field from the segment in which you want to put the virtual field, and click OK.
The virtual field appears in the Fields window of the host file.

![Field Window](image)

4. To complete the Join, drag the mouse pointer from the virtual field to a matching field in the cross-referenced file.

You cannot join multiple fields with this option. The virtual field must be the only Join field.

**Reference:**  **Join Define in File Dialog Box**

The Join Define in File dialog box has the following fields and options:

**Define Field**

Identifies the virtual field that is the host field in the Join.

**With Field**

Associates the virtual field with a segment in the data source. Select any field from the segment in which you want to place the virtual field.
Adding a Tag Name

Tag names provide a way of making identical field names a single instance. They function as aliases for the file names in the Join. Tag names are useful when you join a file to itself because in such a Join all field names, segment names, and file names are the same in the host and cross-referenced files. For more information, see Joining a Data Source to Itself: The Recursive Join on page 147. Tag names are also useful if a file is used as a target more than once in the join structure.

A procedure can use the tag name of a file as a qualifier for field names and aliases in the file.

If you create a report using a recursive Join without tag names, the field names in the cross-referenced file are prefixes with the first four characters of the Join name.

**Procedure: How to Add a Tag Name**

To add a tag name to a file in the Join:

1. Double-click the link between the tables. The Join Properties dialog box appears. The Join Properties dialog box also appears if you right-click a selected link or select Properties from the View menu. A thick line appears when a join is selected.

2. Enter names in the Host Tag and Cross Tag name fields and click OK.

**Note:**

- The following words cannot be used as a tag name: ALL, AND, AS, IN, JOIN, TAG, TO, WITH.
- Tag names can be up to eight characters in length.
**Example:** Impact of Tag Names on Join Syntax

The following example illustrates the effect of tag names on the Join syntax:

```sql
JOIN JOBFILE.JOBSEG.JOBCODE IN jobfile TAG tag name Job TO ALL
    JOBFILE.JOBSEG.JOBCODE IN jobfile TAG tag name Jobc AS J4
END
```

If you add the tag name FILE1 to the host file, and the tag name FILE2 to the cross-referenced file, the following Join is created:

```sql
CREATE a relationship between your databases (JOIN)-
    JOIN PIN IN JOBHIST TAG FILE1 TO ALL PIN IN JOBHIST TAG FILE2 AS JOIN1
```

A report can refer to the field names in the host file with the qualifier FILE1 and in the cross-referenced file with the qualifier FILE2, for example, FILE1.JOBSTART and FILE2.JOBSTART. For additional information, see How to View Join Syntax on page 173.

Tag names are not supported in Report Painter. You can use qualified field names in Report Painter to differentiate fields and identify the table from which you want to retrieve data.

**Working With Joins**

After a Join exists, you can retrieve it, report from it, view its syntax, and clear it. Clearing a join makes it inactive.

**Procedure:** How to Retrieve a Join

1. Right-click a procedure folder in the Explorer and select Edit in Developer Studio Tool from the context menu.
2. Right-click a Join component in the procedure window and select Open.
The Join window opens, displaying your saved Join.

**Procedure: How to Run a Join**

Running a Join executes the generated code and checks the syntax.

If the Join window is already open, displaying a Join, run the Join by doing any of the following:

- Clicking the *Run Join* button from the Join toolbar.
- Clicking *Run* from the Developer Studio File menu.
- Clicking *Run* from the right-click context menu in the Join window.

**Procedure: How to Clear a Join**

1. Use any of the following methods to clear a Join:
   - Click the *Clear Join* button from the Join toolbar.
   - Click *Clear Join* from the Developer Studio File menu.
   - Select *Clear Join* from the right-click context menu in the Join window.

The Join Clear dialog box opens, as shown in the following image.

2. Select the join or joins you want to clear from the Available Joins list. To clear all joins, select *All Joins (*)*.

3. Click the **OK** button.
The selected joins appear in the Joins to Clear list.

4. Click OK.

**Note:** You can also clear a join that is already in effect from a previous procedure and does not appear in the list of available joins. Type the name of the join in the Available Joins field you want to clear and click OK.

**Reference:** Join Clear Dialog Box

The Join Clear dialog box has the following fields and options:

- **Available Joins**
  - Lists all current joins.

- **Joins To Clear**
  - Displays the joins that will be deleted when you click OK.

- **Double Right Arrow**
  - Moves the selected joins to the Joins to Clear list.

- **Double Left Arrow**
  - Moves the selected joins to the Available Joins list.

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

1. Open the procedure that contains a Join component.
2. Open the procedure to which you wish to copy that component.
3. Select the Join component and press the Ctrl key.
4. Drag the Join component to the component connector in the second procedure where you want the Join to appear.

When you do not use the Ctrl key, you move the object instead of dragging it. Your cursor changes depending on whether you move or copy the object.

**Procedure:** How to View Join Syntax

You can quickly view the underlying syntax created by the Join tool.
Click the Show/Hide Source Code button or choose Focexec from the View menu. Syntax similar to the following appears in an editor window, as shown in the following image.

![Focexec Syntax Example]

**Procedure:**  How to View Qualified Field Names

1. Select Options from the Window menu.
2. Select the Reporting tab.
3. Do one of the following:

- Check *Show fully qualified field names* to make this change globally for all tables.
- Right-click a table inside the Join tool and select *Qualified Names*.

**Reference:** Join Properties Dialog Box

To view or modify the properties of a join, right-click the join link and select *Properties* or double-click the join. The Join Properties window appears, as shown in the following image.
The Join Properties window has the following fields and options:

**Host Tag**

Enter a tag name for the data source.

**Cross Tag**

Enter a tag name for the cross-referenced file.

**Selected Fields**

Displays the field that has been joined from the host file to the cross-referenced file. A single arrow indicates a unique join, and a double-arrow indicates a non-unique join.

**Join Name**

Displays the name of the join. To change the default join name, type a new name in this field and click OK.

**Description**

Enter a description. This is not used in the Join command. It is used for reference purposes. A comment (-*) is added to the procedure.

**Join Type**

Indicates the type of join. Choose from the following:

**Multiple Instance (Non-Unique).** A one-to-many join structure that matches one value in the host data source to multiple values in the cross-referenced field. Joining employee ID in an employee data source to employee ID in a data source that lists all the training classes offered by that company would result in a listing of all courses taken by each employee, or a joining of the one instance of each ID in the host file to the multiple instances of that ID in the cross-referenced file. This is the default selection.

**Single Instance (Unique).** A one-to-one join structure that matches one value in the host data source to one value in the cross-referenced data source. Joining an employee ID in an employee data source to an employee ID in a salary data source is an example of a unique join.

**Note:** The join name must be unique and cannot exceed eight characters.
Unspecified. This option is selected in the Join properties when a join that was created in an earlier release is opened in the Join tool. In earlier versions, the defaults for join type and other operators were assumed and not specified in the generated code. Since the Join tool cannot determine the type of join that was created and if there were other commands set to control the join, select the appropriate options (Inner Join, Left Outer Join, Multiple Instances, Single Instances) from the Join tool to upgrade the generated code.

Inner Join. A join that results when a report omits host rows that lack corresponding cross-referenced rows.

Left Outer Join. Extends the results of an Inner Join and retrieves records from both host and cross-referenced tables, including all records from the left table (host) and any records from the right table (cross-referenced) where the condition values match. If there are no matching values in the cross-referenced table, the join still retrieves records from the host table.

Reporting With a Join

With a Join in effect, you can use fields from all joined data sources in a report or graph.

Procedure: How to Report With a Join

From an open procedure in which you have created a Join component:

1. Click the Report or Graph button on the component connector toolbox. The Open dialog box opens.
2. Select the host file and click Open.

The Fields list for the reporting tool lists all fields from the joined data sources. You can choose any of the fields for your report or graph.

Saving a Join

When you exit the Join window, you are prompted to save your work. If you save the Join, it is added to the procedure in which it has been created.

Note: You may also select Save from the Developer Studio File menu, or select the Save icon from the Join toolbar.
**Example: Rearranging Saved Joins**

The Join tool preserves the table sizing and positioning when you reopen the tool. The Join tool displays Joins that use some of the same files in one panel of the Join window. The following two Joins share a host file, as shown in the following image.

**Tip:** If the field names you wish to join are different, you must drag the field names to make the connection.

**Note:** If you close the Join window and then reopen it, the two Joins appear in the same panel.
Merging Data Sources Using the Match Wizard

You can merge two or more data sources, and specify which records to merge and which to sort out, using the Match Wizard. The wizard creates a new data source (a HOLD file), into which it merges fields from the selected records. You can report from the new data source and use it as you would use any other HOLD file. However, you cannot specify a format for the HOLD file. It will be created as a BINARY or ALPHA HOLD file depending on the value of the HOLDFORMAT parameter. The merge process does not change the original data sources.

You select the records to be merged into the new data source by specifying sort fields in the Match Wizard. You specify one set of sort fields (using the BY phrase), for the first data source, and a second set of sort fields for the second data source. The Match Wizard compares all sort fields that have been specified in common for both data sources, and then merges all records from the first data source whose sort values match those in the second data source into the new HOLD file. You can specify up to 32 sort sets. This includes the number of common sort fields.

In addition to merging data source records that share values, you can merge records based on other relationships. For example, you can merge all records in each data source whose sort values are not matched in the other data source. Yet another type of merge combines all records from the first data source with any matching records from the second data source.

You can merge up to 16 sets of data in one Match request. For example, you can merge different data sources, or data from the same data source.

Note: The limit of 16 applies to the most complex request. Simpler requests may be able to merge more data sources.

Access the Match Wizard from the Procedure Viewer in Developer Studio. It uses typical wizard behavior so you can easily navigate from one step in the process to the next.
**Procedure: How to Use the Match Wizard**

1. In the Procedure Viewer, click a component connector (yellow diamond).
   The Component Connector toolbar opens.

2. On the Component Connector toolbar, click Match.
   The Open dialog box appears.

3. Select the first Master File to use in the MATCH logical expression, as shown in the following image.

![Open dialog box](image)
4. Click Open.

The Match Wizard opens, as shown in the following image. It uses typical wizard behavior in which you click a series of Next buttons to navigate from one window to the next. You can also use the Match Preview area to navigate the Wizard.

5. From the Available fields list, select the fields you want to use to build the Match request.

You can add a Match Alias (how the field appears in the Hold file field list) and a Prefix Operator for each field.

**Note:** You can remove fields from this list by clicking the Delete button. If you add more than one field, you can also use the Move Up and Move Down buttons to rearrange the order of the fields.

6. Select either PRINT, SUM, or LIST from the drop-down list.

7. To add a field as a BY field, click a field in the Available Fields list and click the Add field button in the BY fields list box.

**Note:** You must have at least one BY field.
8. Click the Where/If tab if you want to create a WHERE expression or IF condition. This limits the records that are merged together.

9. Click the Add WHERE expression or Add IF condition button to open the Expression Builder. Use the Expression Builder to create the Where/If condition.

10. Click Next. Step 3 of the wizard opens, as shown in the following image.

![Match Wizard - Level 1: Step 3 of 5](image)

11. Click the Browse button to select the data source you want to merge with the original data source.

   The Open dialog box opens.

12. Select a Master File and click Open.

13. Click Next.

   Step 4 of the wizard opens.

14. Add the fields for the second data source, and Where/If criteria, as in Step 5.
Note: In the following image, the second data source is CENTCOMP and the field added is STATE.

15. Click Next.
Step 5 of the Match wizard opens, as shown in the following image.

16. Type a name for the Hold file and select the type of match phrase.
17. Click Finish.
In the following image, the Match Wizard - Summary window opens where you can review the final Match request.

18. If you:

- Have completed your match request, select the Create a Report, Create a Graph, or the Neither option and click Close. Optionally, click the Save the HOLD file to a permanent location check box. When you select this option, the HOLD file is saved to the current Project directory, as shown in this image.

  If you select the Create a Report, Create a Graph, or Neither option, Report Painter, InfoAssist, or Procedure Viewer opens, respectively.

- Want to make more changes to the Match request, click Back or use the Match Preview area. See *How to Edit a Match Request* on page 186.
Procedure: How to Edit a Match Request

1. Open the Match request.

2. In the Match Preview area, click on the file or option you want to edit. The appropriate step in the Match Wizard opens in the left pane.

3. Make your changes and exit the Wizard.

Reference: Usage Notes for Match Requests

- You cannot specify a HOLD format on a MATCH command. The HOLD file must be created using the default HOLD format. HOLD FORMAT FOCUS, or any other format option, is not supported.

- Up to 128 BY phrases and the maximum number of display fields can be used in each MATCH request. The count of sort sets includes the number of common sort fields. The maximum number of display fields is determined by a combination of factors. For details, see the Creating Reports With WebFOCUS Language manual.

- You must specify at least one BY field for each file used in the MATCH request.

- Since records are merged based on matching fields in the separate data sources, sort field names are often adjusted to be the same in each of the merged data sources. Therefore, the ASNAMES parameter is automatically ON for MATCH requests. Customization of names or formats for display fields on the resulting report output should be specified in the TABLE request against the HOLD file produced by the MATCH.
You can report from multi-dimensional data sources, such as SAP BW, Essbase, Microsoft SQL Server Analysis Services, and so on, in Developer Studio. Level and parent/child hierarchy models are supported and reporting rules are automatically enforced.

**In this chapter:**
- Reporting Against Hierarchies
- Creating a Report With a Multi-Dimensional Data Source
- Using Mandatory Variables (SAP BW)
- Sorting Hierarchical Data
- Selecting Records in a Hierarchy
- Manipulating Aggregated Data in an Essbase or SAP BW Hierarchy
- Summary of Reporting Rules For Multi-Dimensional Data Sources
- Showing Hierarchies in HTML Composer
- Using the Synonym Editor for Cube Data Sources

**Reporting Against Hierarchies**

Developer Studio enables you to create hierarchical reports from a multi-dimensional (cube) data source. Two hierarchy models are supported: Level and Parent/Child. The synonym used to access a cube data source determines which type of hierarchy is in effect. The reporting options and field list displays in the Object Inspector and related Report Painter windows adjust automatically depending on the requirements of the hierarchy model in the metadata.

Said another way, when you begin to construct a request, Report Painter provides the appropriate options for you and generates output consistent with the current hierarchy model. If your synonym reflects a:

- **Level hierarchy.** Each hierarchy level is described using a separate field name.
A report request against a level hierarchy must specify the fieldMaterial class name for each level of the hierarchy required in the report. For example, the following request displays the sales volume measure for levels one through three of a hierarchy. For illustration purposes, the field names in this example are actually Level 1, Level 2, and Level 3. Typically they would have names that are more descriptive of the data.

<table>
<thead>
<tr>
<th>LEVEL1</th>
<th>LEVEL2</th>
<th>LEVEL3</th>
<th>Sales Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Products</td>
<td>Computer systems</td>
<td>Computer accessories</td>
<td>67,967,525.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Computer hardware</td>
<td>17,874,768.15</td>
</tr>
<tr>
<td>Paints / aux. and operating supplies</td>
<td>Paints</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Pumps</td>
<td>Pumps (complete)</td>
<td>59,367,205.93</td>
<td></td>
</tr>
</tbody>
</table>

In this model, hierarchical information is provided, but levels are not represented visually in the output. An option that is offered in a parent/child hierarchy.

- **Parent/Child hierarchy.** The hierarchy is described with a set of fields that contain values for properties which describe the position of each member in the hierarchy. For example, there are fields that contain the unique ID of a member, level number, its parent, and the level number of the parent member. To issue a report request, you only need to specify the field name of one of the hierarchy fields. A Hierarchy sort option recreates and formats the hierarchy for display, with appropriate indentations to show the hierarchy levels and relationships.

For example, the following report again shows the sales volume measure for three levels of the Material class hierarchy, but the output is very different from the level hierarchy. The parent/child hierarchy output clearly shows the relationships between the hierarchy members.

<table>
<thead>
<tr>
<th>Material class Member Caption</th>
<th>Sales Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material class</td>
<td>145,209,499.79</td>
</tr>
<tr>
<td>Products</td>
<td>145,209,499.79</td>
</tr>
<tr>
<td>Computer systems</td>
<td>85,842,293.86</td>
</tr>
<tr>
<td>Computer hardware</td>
<td>17,874,768.15</td>
</tr>
<tr>
<td>Computer accessories</td>
<td>67,967,525.71</td>
</tr>
<tr>
<td>Paints / aux. and operating supplies</td>
<td>.00</td>
</tr>
<tr>
<td>Paints</td>
<td>.00</td>
</tr>
<tr>
<td>Pumps</td>
<td>59,367,205.93</td>
</tr>
<tr>
<td>Pumps (complete)</td>
<td>59,367,205.93</td>
</tr>
</tbody>
</table>
Using the parent/child model, you can also specify a When condition to select the members of the hierarchy you want to display and a Show condition to specify the data in the hierarchy you want to display. See Selecting Records in a Hierarchy on page 199.

Also, depending on the type of cube involved, you can specify whether you want the Measure values for each parent to represent aggregates for all of its children (full aggregation) or only those selected for display (partial aggregation).

**How much do you need to know about synonyms?**

Although the hierarchy type designation in the synonym determines how the report looks, and how much control you have over data retrieval within the hierarchy, you really do not need to know the details of the synonym to successfully create a report. In Developer Studio, reporting options and field list displays automatically adjust to the requirements of the hierarchy model that is in effect in the metadata.

If you are interested in examining the Master File component of the synonym, you can do so in the Synonym Editor or in a text editor. See the Using the Synonym Editor for Cube Data Sources on page 212 for more information.

**Creating a Report With a Multi-Dimensional Data Source**

You create a report using a multi-dimensional data source in Report Painter the same way you would if you were reporting from any other type of data source. For both level and parent/child hierarchy models, all reporting functionality is available. One of the differences between reporting against a multi-dimensional data source and reporting against another type of data source is that you have additional folders that group related fields in the Object Inspector and other tools that display field lists. These cube-related folders are Variables (SAP BW), Measures, Characteristics, Hierarchies, and Properties. See Viewing the Field Lists on page 190.

In addition, the following features are available for hierarchies that use the parent/child model. (If you are working with the level hierarchy model, you can still sort and screen data using the traditional methods.) These include:

- The Hierarchy sort option, which automatically sorts and formats a hierarchy with appropriate indentations that clearly show the parent/child relationships in a hierarchy. See Sorting Hierarchical Data on page 198.

- When and Show conditions, which enable you to select the hierarchy members and data values you want in the report output. See Selecting Records in a Hierarchy on page 199.
Creating a Report With a Multi-Dimensional Data Source

- Full and partial aggregation for some cube data sources, which enables you to show aggregation values as they appear in the cube (full aggregation) or aggregation for only those values that appear in your report output (partial aggregation). See *Manipulating Aggregated Data in an Essbase or SAP BW Hierarchy* on page 201.

There are specific reporting rules that apply to multi-dimensional data (parent/child and level). When you use Report Painter, these rules are automatically enforced. For more information, see *Summary of Reporting Rules For Multi-Dimensional Data Sources* on page 203.

**Viewing the Field Lists**

When you are creating a report with a multi-dimensional data source, you see the following additional folders in the Object Inspector and all other field lists.

- **Variables** (also known as filters or parameters). Used to slice data. They can be used in data selection (Where statements) and in some cases, they can be used to populate a drop-down list box. The drop-down list box can then be assigned to an amper variable, which can be used in a Where statement.

  Variables can appear in your fields list as optional or mandatory. Variables cannot be used as a field in a report with other elements, like Characteristics and Measures. Variables cannot be used in a When phrase. For information on mandatory variables, see *Using Mandatory Variables (SAP BW)* on page 197.

  **Note:** Variables are only applicable for SAP BW data sources.

- **Measures.** Typically define how much or how many. For example, Sales, Revenue, and Gross Margin are possible Measures in an Account characteristic and specify how many units were sold, how much revenue was generated, and at what profit margin, respectively.

  A Measure can also be a formula where a simple aggregation function, such as sum, can combine several Measure values into one. A formula can also be based on references to a point within a characteristic.

- **Characteristics** or **Dimensions.** Are classifications (such as division, region, product, organizational unit, and so on) on which measures (such as revenue and sales) are based. Typical characteristics or dimensions are Time, Chart of Accounts, Geography, Product, Organization, or Routes.

  Characteristics or dimensions can have multiple organized hierarchies that describe similar sets of members upon which to base an analysis. For example, a characteristic named Customer can have a hierarchy that groups customers according to their geographic locations, as well as another hierarchy that groups them according to their industries.
Properties (also known as attributes). Represent additional information about each characteristic. For the Product characteristic, color could be a property.

Hierarchies. Provide a method of displaying a characteristic structured and grouped according to individual evaluation criteria. Characteristics can have multiple hierarchies associated with them.

Reference: Organization of Variables, Measures, Characteristics, Properties, and Hierarchies

Variables, Measures, Characteristics or Dimensions, Properties, and Hierarchies are organized in folders that contain the fields you can use in your report. The following image shows these categories and what they contain for an SAP BW and Essbase data source.

- Cube Name
  - Variables
    - Variable fields
  - Measures
    - Measure fields
  - Characteristics
    - Properties
      - Property fields
    - Hierarchies
      - Hierarchy 1
        - Hierarchy levels or parent child structure
      - Hierarchy 2
        - Hierarchy levels or parent child structure
The following image shows these categories and what they contain for a Microsoft Analysis Server data source.

- Cube Name
  - Measures
    - Measure fields
  - Characteristics
    - Hierarchies
      - Hierarchy 1
        - Hierarchy levels or parent child
          - Properties
          - Attributes
      - Hierarchy 2
        - Hierarchy levels or parent child
          - Properties
          - Attributes

**Reference:** Parent/Child and Level Hierarchy Display

When the synonym for the data is created, it is set up as either a parent/child hierarchy or level hierarchy. Hierarchies appear in the Characteristics category. The following image shows the difference between the parent/child and level hierarchy display. Both field lists show the Material class hierarchy expanded in the tree. The list on the left shows a parent/child hierarchy while the list on the right shows a level hierarchy.
The parent/child and level hierarchies contain the same data but are represented differently. For example, in the parent/child hierarchy shown in the previous image, you have one field (Member Caption) that contains all the members of the hierarchy. In the level hierarchy, the members of the hierarchy are listed as separate fields.

**Note:** The parent/child hierarchy also contains virtual (or define) fields (for SAP BW). See *Fields in a Parent/Child Hierarchy* on page 193 for details.

**Reference:** *Fields in a Parent/Child Hierarchy*

By default, the following fields are shown for each hierarchy in the field lists when you select the Dimension view (this is the default view for SAP BW metadata):

- `fieldname Caption`
These fields show all combinations of Caption and Key that you can use in your report. The last three fields you see in a parent/child hierarchy represent Key, Key-Caption, and Caption-Key. These are virtual fields that are defined during synonym creation for SAP BW.

Name or Key is the technical name and Caption is the user-friendly name (description) for the cube item. Caption cannot be used as a By field unless a field representing either the Name (Member Name) or Member Unique Name is included in the report. The reason for this is since it is possible for more than one item to have the same Caption, if you do not identify the Name or Member Unique Name then there is potential for aggregation of data at different levels. When you include the Name or Member Unique Name, then Caption can be uniquely identified and it can be used as a By field.

Using Advanced Fields in a Parent/Child Hierarchy (SAP BW)

When you are working with a data source that has been set up to use a parent/child hierarchy, the default display for the hierarchy in the Object Inspector and the field lists is to show the Member Caption field and the virtual fields. All characteristics have a hierarchy called flat hierarchy that represents the leaf nodes of the hierarchy. A leaf node is a node that does not have any children. Flat hierarchies are represented with only one field.

You can view additional fields, such as the Member Unique Name, Parent Level, and Member Children Cardinality, and so on. To view these fields, right-click in the Object Inspector or any fields list and select Show Advanced Fields. The fields list refreshes and collapses to show the Advanced Fields for the hierarchy members in the Characteristics folder.
The following image shows the same hierarchy (Product Hierarchy for material MARA), with the default field display on the left and the Advanced Fields display on the right.

The Advanced Fields are:

- **Member Unique Name** represents the unique ID for each member of the hierarchy.
- **Member Name** represents the technical name of each member of the hierarchy.
- **Member Level Number** represents the level number for each member of the hierarchy.
- **Parent Unique Name** represents the unique ID of the parent for each member of the hierarchy.
- **Parent Level** represents the level number of the parent for each member of the hierarchy.
- **Member Children Cardinality** represents the number of children for each member of the hierarchy.

**Reference: Hierarchy Icons**

The following table describes the icons that appear in the Object Inspector and in the field lists when you are creating a report using a multi-dimensional data source.

For more information about Report Painter, see the *Creating Reports With Report Painter* manual.

The first column shows the icon and the second column provides a description of the icon.
<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Characteristics" /></td>
<td>Characteristics contains the Properties and Hierarchies folders.</td>
</tr>
<tr>
<td><img src="image" alt="Optional" /></td>
<td>Optional variable.</td>
</tr>
<tr>
<td><img src="image" alt="Mandatory" /></td>
<td>Mandatory variable that is not satisfied (selection criteria has not been created).</td>
</tr>
<tr>
<td><img src="image" alt="Mandatory" /></td>
<td>Mandatory variable that is satisfied (selection criteria has been created).</td>
</tr>
<tr>
<td><img src="image" alt="Measure" /></td>
<td>Measure.</td>
</tr>
<tr>
<td><img src="image" alt="Hierarchy" /></td>
<td>Hierarchy.</td>
</tr>
<tr>
<td><img src="image" alt="Attribute" /></td>
<td>Attribute.</td>
</tr>
<tr>
<td><img src="image" alt="Leaf" /></td>
<td>Represents a leaf node in a parent/child hierarchy. A leaf node is a member of a hierarchy that does not have any children. Represents Level 1 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image" alt="Level 2" /></td>
<td>Represents Level 2 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image" alt="Level 3" /></td>
<td>Represents Level 3 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image" alt="Level 4" /></td>
<td>Represents Level 4 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image" alt="Level 5" /></td>
<td>Represents Level 5 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image" alt="Level 6" /></td>
<td>Represents Level 6 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image" alt="Level 7" /></td>
<td>Represents Level 7 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image" alt="Level 8" /></td>
<td>Represents Level 8 in a level hierarchy.</td>
</tr>
<tr>
<td>Icon</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image1.png" alt="Icon" /></td>
<td>Represents Level 9 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image2.png" alt="Icon" /></td>
<td>Represents Level 10 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Icon" /></td>
<td>Represents Level 11 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Icon" /></td>
<td>Represents Level 12 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image5.png" alt="Icon" /></td>
<td>Represents Level 13 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image6.png" alt="Icon" /></td>
<td>Represents Level 14 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image7.png" alt="Icon" /></td>
<td>Represents Level 15 in a level hierarchy.</td>
</tr>
<tr>
<td><img src="image8.png" alt="Icon" /></td>
<td>Represents Level 16 in a level hierarchy.</td>
</tr>
</tbody>
</table>

**Using Mandatory Variables (SAP BW)**

Mandatory variables are variables that are required in a procedure. If a mandatory variable is not satisfied, the procedure will not run. To satisfy a mandatory variable, you create selection (Where) criteria for it. Mandatory variables are identified in the field lists with one of the following icons.

- ![Icon](image9.png) - This icon represents a mandatory variable that is not satisfied (selection criteria has not been created).

- ![Icon](image10.png) - This icon represents a mandatory variable that is satisfied (selection criteria has been created).

Use the Where assist tool in Report Painter to create selection (Where) criteria for mandatory variables. Once a mandatory variable is satisfied, the icon changes.

You can use the variable itself or one of its DEFINE fields to create the selection criteria. The DEFINE fields are variable Caption and variable Name. Name is the technical name and Caption is the user-friendly name (description) for the variable. If you use the variable itself, it uses Caption and Name (concatenated with trailing blanks).
If you do not create selection criteria for a mandatory variable, you will not be able to get data values for other Where and/or When statements. Therefore, all mandatory variable selection criteria should be created first.

**Note:** When you create selection criteria for mandatory variables, this criteria is propagated to all other selection (Where) criteria statements and When statements you create. This means that when you retrieve values for other Where or When statements, the values that are returned are only those values that meet the selection criteria for the mandatory variable.

For more information on variables, see *Viewing the Field Lists* on page 190.

**Procedure:** How to Create Selection Criteria for Mandatory Variables

1. Click the *Where/If* button in the Report Painter toolbar. The Report Options dialog box opens, with the Where/If tab selected.
2. Click Assist. The Expression Builder opens.
3. Double-click the mandatory variable, or one of its DEFINE fields, to add it to the expression list. Notice that as soon as you add the mandatory variable to the list, the icon changes.
4. Select the Logical Relation, the Compare Type, and the Compare Value.
5. Repeat these steps for all mandatory variables.
6. Click **OK** to exit the Expression Builder and **OK** to exit the Report Options dialog box.

**Sorting Hierarchical Data**

You can use the standard By and Across sort options for multi-dimensional data sources that are set up with either the parent/child or level hierarchy model.

In addition, for parent/child hierarchies, you can use the Hierarchy sort option. This option sorts and formats a hierarchy with appropriate indentations that clearly show the parent/child relationships. You can have multiple Hierarchy sort fields in a report. To see an example of a report that uses the Hierarchy sort option, see the parent/child example in *Reporting Against Hierarchies* on page 187.

**Procedure:** How to Use the Hierarchy Sort Option

1. In the Object Inspector, expand the Characteristics folder.
2. Expand a Characteristic, then expand the Hierarchies folder.
3. Double-click a field in the hierarchy to add it to the report.
4. Select the field in Report Painter and click *Hierarchy* in the reporting toolbar.
Note: You can have multiple Hierarchy sort fields in your report but they must be from different Characteristics. You will notice that Report Painter automatically enforces the rule that you can only have one hierarchy per Characteristic in a report by graying out all other hierarchies in the Characteristic as soon as you add a hierarchy to the report.

Selecting Records in a Hierarchy

You can select records for both parent/child hierarchies and level hierarchies. For both of these hierarchy models, you can use only Where statements to select data in the hierarchy.

If you are reporting with a parent/child hierarchy, you can display specific members of a hierarchy using a When condition. In addition, you can use a Show command to specify the data to show in the report output relative to the hierarchy members selected in the When condition. If there is no When condition, the Show command is applied to the root node of the hierarchy.

Procedure: How to Select Data in a Parent/Child Hierarchy

1. In Report Painter, right-click a Hierarchy sort field and select When from the context menu.

   The Expression Builder opens. The hierarchy you are working with is automatically selected and expanded. You can only create a When expression with fields in a hierarchy. Characteristics, properties, or measures cannot be used.

   Alternatively, if you have the SHOW Builder dialog box open you can access the Expression Builder by clicking When.

2. Create an expression.

   Note:
   - You must manually enter a value for hierarchy fields since the Expression Builder cannot retrieve values for parent/child hierarchies.
   - Use only the Value or Parameter Compare Types with hierarchies.
   - If you use the Parameter Compare Type, when you enter the compare value using the Variable Editor you cannot use the Variable Type Range since this does not apply to hierarchies.

Procedure: How to Show a Range of Data in a Parent/Child Hierarchy

1. In Report Painter, right-click a Hierarchy sort field and select Show from the context menu.
Alternatively, if you have the Expression Builder open you can access the SHOW Builder by clicking *Show Builder*.

The Show Builder dialog box opens.

2. In the From-To Options field, select your From and/or To options. These specify the range of values in the hierarchy that you want to show in the report output. You can select an actual value or you can select *Variable*, which opens the Variable Editor.

   **Note:** If the Warning icon on the lower-right corner is activated, there is an error in the Show condition you are creating. Click the icon for details.

3. Click *OK*.

**Reference:**  **SHOW Builder Dialog Box**

![SHOW Builder Dialog Box](image)

**From-To Options**

Enables you to select the range of values to display in the hierarchy. You can use From without To and To without From.

**From**

Select either TOP or a level number. TOP specifies that measure values for the ancestors of the selected node to the root node appear in the report.
To
Select either BOTTOM or a level number. BOTTOM specifies that measure values for the
descendants of the selected node up to the leaf nodes of the hierarchy appear in the report.

Edit
This option appears when you have defined a variable for the From or To option. Click this button to edit the variable.

When
Opens the WHEN Expression Builder dialog box.

Warning icon
Shows warnings and miscellaneous information about the Show Builder range that you are entering.

Screening Data
Once hierarchy members are selected, you can screen the retrieved data by applying Where criteria to the selected members. Where criteria are applied to the leaf nodes and are processed after the phase of the request that selects hierarchy members (those members specified in the When phrase). Therefore, Characteristic Properties can be used in Where tests.

These tests can also reference hierarchy fields. However, since the selection criteria are always applied to the values at the leaf nodes, they cannot select data based on values that occur at higher levels. For example, in a characteristic with Continents, Countries, and Cities, your request will not display any rows if you use Where to select at the Country level, but it may if you use it to select at the City level.

Screening Based on Aggregated Values
Since Measures are summarized values they should be referenced in Where Total tests and Computes, which are processed after the hierarchy selection and aggregation phases of the request.

Manipulating Aggregated Data in an Essbase or SAP BW Hierarchy
For data sources that use the parent/child hierarchy model, you can show either full or partial aggregation in the report output.
Full aggregation retrieves aggregated values from the data source cube and displays them in the report output exactly as they appear in the cube. The retrieved value may not reflect the sum of the displayed descendants if some descendants are eliminated from the report output based on When or Show conditions. Full aggregation can only be applied to Measure fields and a Hierarchy sort field must exist in the request.

Partial aggregation retrieves partially aggregated data. Totals that appear in the report output only reflect those values that actually appear in the output. This is the default aggregation mode in Developer Studio.

**Procedure:** How to Set Full Aggregation Mode

1. Select a Measure field in the report.
2. Choose the FROLL prefix operator from the reporting toolbar.

**Note:** Full aggregation can only be applied to Measure fields and a Hierarchy sort field must exist in the request.

**Example:** Using Full and Partial Aggregation

The following image shows a report with partial aggregation (FROLL is off) and full aggregation (FROLL is on). The first Measure field is a SUM Measure field (Net value of incoming orders) that shows partial aggregation. The second Measure field shows full aggregation.

<table>
<thead>
<tr>
<th>Material class</th>
<th>Member Caption</th>
<th>Net value of incoming orders</th>
<th>Net value of incoming orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material class</td>
<td></td>
<td>66,272,326.22</td>
<td>157,092,419.36</td>
</tr>
<tr>
<td>Products</td>
<td></td>
<td>66,272,326.22</td>
<td>157,092,419.36</td>
</tr>
<tr>
<td>Pumps</td>
<td></td>
<td>66,272,326.22</td>
<td>66,272,826.22</td>
</tr>
<tr>
<td>Pumps (complete)</td>
<td></td>
<td>66,272,326.22</td>
<td>66,272,826.22</td>
</tr>
<tr>
<td>Pump PRECISION 100</td>
<td></td>
<td>3,165,739.60</td>
<td>3,165,739.60</td>
</tr>
<tr>
<td>Pump PRECISION 101</td>
<td></td>
<td>7,226,171.82</td>
<td>7,226,171.82</td>
</tr>
<tr>
<td>Pump PRECISION 102</td>
<td></td>
<td>14,138,374.11</td>
<td>14,138,374.11</td>
</tr>
<tr>
<td>Pump PRECISION 103</td>
<td></td>
<td>11,255,431.58</td>
<td>11,255,431.58</td>
</tr>
<tr>
<td>Pump PRECISION 104</td>
<td></td>
<td>15,168,335.16</td>
<td>15,168,335.16</td>
</tr>
<tr>
<td>Pump cast steel IDESNORM 170-230</td>
<td></td>
<td>4,961,384.91</td>
<td>4,961,384.91</td>
</tr>
<tr>
<td>Pump standard IDESNORM 100-402</td>
<td></td>
<td>10,357,390.04</td>
<td>10,357,390.04</td>
</tr>
</tbody>
</table>
Notice that the partial aggregated value for Products in the first column (66,272,826.22) reflects the exact sum of the values for Pump PRECISION 100, Pump PRECISION 101, Pump PRECISION 102, Pump PRECISION 103, Pump PRECISION 104, Pump cast steel IDESNORM 170-230, and Pump standard IDESNORM 100-402.

The full aggregated value for Products in the second column (157,092,419.36) does not reflect the sum of those fields. It reflects the aggregated value that exists in the cube.

**Summary of Reporting Rules For Multi-Dimensional Data Sources**

When you use Report Painter to create a report using a multi-dimensional data source, certain reporting rules are enforced automatically. The tables below summarize these rules.

**Reference: Summary of Reporting Rules for SAP BW**

When you use Report Painter to create a report using an SAP BW data source, the following rules are enforced automatically.

<table>
<thead>
<tr>
<th>Component</th>
<th>Rules</th>
</tr>
</thead>
</table>
| Variables (also known as Parameters or Filters) | Variables cannot be displayed in a report. They can only be used in a Where statement.  
**Note:** There is one exception to this rule, and that is if the variable is the only field in the request.  
Single Value Variables should only be used with the EQ operator and only one literal (value) should appear in the Where statement. It is possible to use an amper variable.  
Mandatory Variables require a value and must exist in the report. You can use these to create a Where statement. A warning message appears if you run the report without creating a Where statement for a mandatory variable. |
## Summary of Reporting Rules For Multi-Dimensional Data Sources

<table>
<thead>
<tr>
<th>Component</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measures</td>
<td>Can be used in:</td>
</tr>
<tr>
<td></td>
<td>- Computes.</td>
</tr>
<tr>
<td></td>
<td>- Sum.</td>
</tr>
<tr>
<td></td>
<td>- Where Total statements. When creating a Where Total statement you should enter values manually instead of retrieving values.</td>
</tr>
<tr>
<td></td>
<td>Cannot be used in:</td>
</tr>
<tr>
<td></td>
<td>- By fields.</td>
</tr>
<tr>
<td></td>
<td>- Defines.</td>
</tr>
<tr>
<td></td>
<td>- Across fields.</td>
</tr>
<tr>
<td></td>
<td>- For.</td>
</tr>
<tr>
<td></td>
<td>- Where statements.</td>
</tr>
<tr>
<td></td>
<td>- Headings or footings.</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Only one hierarchy per characteristic can be used in the report. Hierarchies can be used as:</td>
</tr>
<tr>
<td></td>
<td>- By field.</td>
</tr>
<tr>
<td></td>
<td>- Across field.</td>
</tr>
<tr>
<td></td>
<td>- For.</td>
</tr>
<tr>
<td></td>
<td>- Detail fields.</td>
</tr>
<tr>
<td></td>
<td>- Defines.</td>
</tr>
<tr>
<td></td>
<td>- Computes.</td>
</tr>
<tr>
<td></td>
<td>- Where statements.</td>
</tr>
<tr>
<td></td>
<td>- Headings and footings.</td>
</tr>
<tr>
<td>Properties (also known as Attributes)</td>
<td>Same rules as Hierarchy.</td>
</tr>
</tbody>
</table>
**Summary of Reporting Rules for Essbase**

When you use Report Painter to create a report using an Essbase data source, the following rules are enforced automatically.

<table>
<thead>
<tr>
<th>Component</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables (also known as</td>
<td>Cannot be used for an Essbase data source.</td>
</tr>
<tr>
<td>Parameters or Filters)</td>
<td></td>
</tr>
<tr>
<td>Measures</td>
<td>Can be used in:</td>
</tr>
<tr>
<td></td>
<td>- Computes.</td>
</tr>
<tr>
<td></td>
<td>- Defines.</td>
</tr>
<tr>
<td></td>
<td>- Sum.</td>
</tr>
<tr>
<td></td>
<td>- Headings or footings.</td>
</tr>
<tr>
<td></td>
<td>- Where Total statements. When creating a Where Total statement, you</td>
</tr>
<tr>
<td></td>
<td>should enter values manually instead of retrieving values.</td>
</tr>
<tr>
<td></td>
<td>Cannot be used in:</td>
</tr>
<tr>
<td></td>
<td>- By fields.</td>
</tr>
<tr>
<td></td>
<td>- Across fields.</td>
</tr>
<tr>
<td></td>
<td>- For.</td>
</tr>
<tr>
<td></td>
<td>- Where statements.</td>
</tr>
</tbody>
</table>
### Summary of Reporting Rules For Multi-Dimensional Data Sources

<table>
<thead>
<tr>
<th>Component</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy</td>
<td>Only one hierarchy per characteristic can be used in the report. Hierarchies can be used as:</td>
</tr>
<tr>
<td></td>
<td>- By field.</td>
</tr>
<tr>
<td></td>
<td>- Across field.</td>
</tr>
<tr>
<td></td>
<td>- For.</td>
</tr>
<tr>
<td></td>
<td>- Detail fields.</td>
</tr>
<tr>
<td></td>
<td>- Defines.</td>
</tr>
<tr>
<td></td>
<td>- Computes.</td>
</tr>
<tr>
<td></td>
<td>- Where statements.</td>
</tr>
<tr>
<td></td>
<td>- Headings and footings.</td>
</tr>
<tr>
<td>Properties (also known as Attributes)</td>
<td>Same rules as Hierarchy, with the exception of:</td>
</tr>
<tr>
<td></td>
<td>- Defines, which cannot be used for an Essbase data source.</td>
</tr>
</tbody>
</table>

**Reference:** Summary of Reporting Rules for Microsoft Analysis Server 2005

When you use Report Painter to create a report using a Microsoft Analysis Server 2005 data source, the following rules are enforced automatically.

<table>
<thead>
<tr>
<th>Component</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables (also known as Parameters or Filters)</td>
<td>Cannot be used for a Microsoft Analysis Server 2005 data source.</td>
</tr>
<tr>
<td>Component</td>
<td>Rules</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Measures</td>
<td>Can be used in:</td>
</tr>
<tr>
<td></td>
<td>- Computes.</td>
</tr>
<tr>
<td></td>
<td>- Headings or footings.</td>
</tr>
<tr>
<td></td>
<td>- Sum.</td>
</tr>
<tr>
<td></td>
<td>- Where Total statements. When creating a Where Total statement, you should enter values manually instead of retrieving values.</td>
</tr>
<tr>
<td></td>
<td>Cannot be used in:</td>
</tr>
<tr>
<td></td>
<td>- By fields.</td>
</tr>
<tr>
<td></td>
<td>- Defines.</td>
</tr>
<tr>
<td></td>
<td>- Across fields.</td>
</tr>
<tr>
<td></td>
<td>- For.</td>
</tr>
<tr>
<td></td>
<td>- Where statements.</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Only one hierarchy per characteristic can be used in the report. Hierarchies can be used as:</td>
</tr>
<tr>
<td></td>
<td>- By field.</td>
</tr>
<tr>
<td></td>
<td>- Across field.</td>
</tr>
<tr>
<td></td>
<td>- For.</td>
</tr>
<tr>
<td></td>
<td>- Detail fields.</td>
</tr>
<tr>
<td></td>
<td>- Defines.</td>
</tr>
<tr>
<td></td>
<td>- Where statements.</td>
</tr>
<tr>
<td>Properties (also known as</td>
<td>Same rules as Hierarchy.</td>
</tr>
<tr>
<td>Attributes)</td>
<td></td>
</tr>
</tbody>
</table>
Showing Hierarchies in HTML Composer

By using a tree structure in an HTML report, you can show hierarchical data from a multi-dimensional data source (for example, SAP BW), that uses the parent/child model. You may also use a Tree control for non-hierarchical data sources. Level hierarchies are not supported.

The behavior of the Tree control is integrated with the parameter definition. If a parameter is defined as a single value and that parameter is bound to a tree control, the tree control uses option buttons for each node in the hierarchy. If the parameter is defined as Multiselect OR and Multiselect AND, and that parameter is bound to a tree control, then the tree control uses check boxes for each node in the hierarchy, enabling you to select multiple nodes.

The following image shows the Tree Control component in HTML Composer.

![Tree Control Component](image)

**Note:** Cube data should be used to populate single source trees.

For more information about using HTML Composer, see the *Designing a User Interface for a Web Application With the HTML Composer* manual.

**Procedure: How to Add a Tree Control to an HTML Page Using an Embedded Procedure**

This procedure describes how to add a tree control for a multi-dimensional data source, using an embedded procedure.

1. In HTML Composer, insert a tree control from the Components toolbar.

   **Tip:** You may select Single source Tree control or Multi source Tree control. If no type is selected, Single source Tree control is the default, as shown in the following image.

   ![Tree Control Selection](image)

   The cursor changes to a crosshair.

2. Drag the crosshair to create a tree control, and adjust it to the size you want.
A tree control is created in the layout and assigned the name treecontrol\(n\), where \(n\) is a number. Additionally, the Properties and settings dialog box appears for the tree control.

3. Optionally, you may select the *Expanded* property from the Properties tab of the Properties window to show the tree control expanded at run time.

4. Optionally, you may select the *Hyperlink* property from the Properties tab of the Properties window to show the tree nodes as hyperlinks, instead of radio buttons at run time.

5. From the Properties and settings dialog box, select *Dynamic* as the Data type.

6. Keep *Embedded procedure* selected and click the *browse (⋯)* button adjacent to the first input field.
   
The Get source file dialog box opens.

7. Select a multi-dimensional data source and click *Open*.
   
The multi-dimensional data source is added as the embedded procedure.

8. Click the *Value field* browse button to select a field from the hierarchy.
   
The Value field is the data source field from which the values will be retrieved.

9. Click the *Display field* browse button to select a field from the hierarchy.
   
The Display field is the text that represents the parameter value in the tree control.

10. Save and run the page to view the multi-dimensional data source in the tree control.

**Procedure:**  How to Add a Tree Control to an HTML Page Using an Existing Procedure

You can select an existing procedure to add to the tree control in an HTML page. When you select a procedure, it should use fields from the parent/child hierarchy and be set up as follows:

```
TABLE FILE file
SUM FST.dispfield
BY ParentUniqueField
BY UniqueField
BY datafield
ON TABLE PCHOLD FORMAT XML
END
```

where:

```
file
```

Is the name of the data source.
**dispfield**

Is the field whose values display in the tree control.

**ParentUniqueField**

Is the field that represents the parent for the parent/child hierarchy (PROPERTY = PARENT_OF).

**UniqueField**

Is the field that represents the unique IDs for the hierarchy members (PROPERTY=UID).

**datafield**

Is the field whose values are passed as the parameter value.

After the procedure is set up, follow these steps:

1. In HTML Composer, insert a tree control from the Components toolbar.

   **Tip:** You may select Single source Tree control or Multi source Tree control. If no type is selected, Single source Tree control is the default.

   The cursor changes to a crosshair.

2. Drag the crosshair to create a tree control, and adjust it to the size you want.

   A tree control is created in the layout and assigned the name treecontrol(n), where n is a number. Additionally, the Properties and settings dialog box appears for the tree control.

3. Optionally, you may select the **Expanded** property from the Properties tab of the Properties window to show the tree control expanded at run time.

4. Optionally, you may select the **Hyperlink** property from the Properties tab of the Properties window to show the tree nodes as hyperlinks, instead of radio buttons at run time.

5. From the Properties and settings dialog box, select **Dynamic** as the Data type.

6. Select **External Procedure** and click the browse (...) button adjacent to the first input field.

   The Get source file dialog box opens.

7. Select a procedure and click **Open**.

   The procedure name is added as the external procedure.

8. Click the **Value field** browse button to select a field from the hierarchy.

   The Value field is the data source field from which the values will be retrieved.

9. Click the **Display field** browse button to select a field from the hierarchy.

   The Display field is the text that represents the parameter value in the tree control.
10. Save and run the page to view the multi-dimensional data source in the tree control.

**Reference:** Tree Control Properties in the Properties Window

When a tree control is selected, the Properties tab in the Properties window contains options that control the properties of tree controls.

Click a property to display a description of the selected property at the bottom of the Properties window.

**Select all children.** The Select all children option, when set to Yes, makes it so that when a parent is selected, all children are selected and when a parent is unselected, all children are unselected. No is selected by default.
Using the Synonym Editor for Cube Data Sources

The Synonym Editor enables you to make further modifications to your Cube synonym definitions. The Cube synonym is displayed in the Cube View tab. If you select a component of the synonym, its properties appear in the right pane.

You can use the Synonym Editor to:

- Edit field names and titles for measures, properties, and variables. You may also edit the field prefix for the defined fields.
- Edit titles for multiple languages, delete titles, and add titles for additional languages if you are using code page UTF8.
- Delete measures, dimensions, or parent/child hierarchies.
- View the usage and actual formats and the ability to change the usage display format for the field.
- Change an internal attribute value.
- Change the access property for variables.
- Change the field prefix used to create the field names, mandatory property names, and DEFINE field names generated for parent/child hierarchies by the Synonym Wizard.
- Create virtual measures (this is implemented with COMPUTE fields in the Master File).

Note: The Number of layers property is not available in a single tree source property window.
The following image is an example of an SAP BW Data Source in the Synonym Editor.

For more information about using the Synonym Editor, see the *Describing Data With Graphical Tools* manual.
Using the Synonym Editor for Cube Data Sources
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.

**In this chapter:**

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

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

- **Procedure Viewer.** Provides a graphical environment for the creation of a generic procedure. This option is the default.

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

- **Composer.** Provides a graphical environment for the creation of a compound report.

- **SQL Report Wizard.** Assists you with SQL passthru, which allows you to execute SQL code that retrieves data from an RDBMS. You can use the resulting extract file in Report Painter.

- **Graph.** Generates highly complex reports, charts, and documents with a powerful yet simple to use interface.

- **Text Editor.** Provides administrators with a standard text editor, allowing you to create a procedure by entering commands.

- **Define Function.** Provides a wizard for creating DEFINE functions.
**Procedure:** How to Select a Creation Tool

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

The Add Procedure dialog box opens, as shown in the following image.

![Add Procedure Dialog Box](image.png)

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

3. In the File name field, type a name for the new procedure. Do not include any spaces in the name or add an extension. Developer Studio automatically appends the extension .fex.
4. In the Files of type drop-down list, accept the default, Procedure Files (*.fex).
Selecting a Creation Tool

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

- **Procedure Viewer** provides a graphical environment for the creation of a generic procedure. For information, see *How to Create a Reporting Procedure Using the Procedure Window* on page 219.

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

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

- **Graph** accesses InfoAssist, where you can transform almost any type of data into an effective chart that you can customize. For information, see the *WebFOCUS InfoAssist User’s Manual*.

- **SQL Report Wizard** assists you with SQL passthru, which allows you to execute SQL code that retrieves data from an RDBMS. You can use the resulting extract file in Report Painter or InfoAssist. For information, see *Using the SQL Report Wizard* on page 233.

- **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 220.

- **Define Function** accesses the Define Function Wizard, where you can create a virtual field function.

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

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

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

**Procedure:**  **How to Create a Report Procedure With Report Painter**

1. Expand the folder that will contain the report procedure.

2. Right-click the folder where you want to create the report procedure.
3. Click New, and then click Procedure.  
   The New Procedure dialog box opens.

4. Type a name for the new procedure in the File name field, select Report Painter from the Create with drop-down list, and click Create. 
   The WebFOCUS Table List dialog box opens and prompts you to select a Master File, as shown in the following image.

![WebFOCUS Table List dialog box](image)

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.

5. Select a Master File.

6. Click OK.
   Report Painter opens.
   See the Creating Reports With Report Painter manual for more information.

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

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

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

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

2. Click a diamond object and select a component, such as Report or Graph or use the toolbar buttons. For details on how to proceed, see Creating a Procedure Component on page 224.

Once you have created components, you can do the following with components.

- Move them before or after an existing component (drag and drop).
- Copy by pressing the Ctrl key and dragging the selected object.
- Right-click on a component provides a Context menu with the options to view as text in a separate editor window by selecting Edit Text. You can perform other operations, such as Run, Cut, Copy, Delete, and Open the tool used to create the selected component.

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

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

2. Enter the syntax for your procedure.
See the Creating Reports With WebFOCUS Language manual for details on WebFOCUS syntax.

3. Click the Save button on the text editor toolbar when you are done. See Editing Application Components as Text in Developer Studio on page 369, for details on the text editor.

**Reference: Add Procedure Dialog Box**

![Add Procedure Dialog Box](image)

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

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

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

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

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

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

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

1. Add the directory of the procedure to the project path. When you add the directory of a procedure to the project path, all project files become available. For details on modifying available directories for a project, see How to Modify Directory Paths on page 50.

2. Add a procedure to the project. Once the directory of the procedure is visible to a project, add the procedure to the project so it will be available for reporting and deployment. For details, see How to Add a Procedure to a Project on page 222.

**Note:** You can also remove a procedure from a project. For details, see How to Remove a Procedure From a Project on page 223.

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

1. Open the Procedures folder in the project.

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

   The right window pane displays all procedures in the paths defined for the selected project. Procedures that are *not* added to the project path have a light gray icon next to them, and the file type description is FOCUS Executable Visible within Project Path.
3. Do one of the following:
   - Right-click a procedure from the available procedures on the right, and select *Add to Project* from the shortcut menu.
     
     or
   
   - Select a procedure and click the *Inserts the selected items into the project* button on the Explorer toolbar.
     
     The procedure icon changes color to signify that it has been added to the project, and the file type changes to FOCUS Executable.

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

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

1. Open the *Procedures* folder in the project.

2. If not already selected, click the *Displays all files in the project path* button on the Explorer toolbar.
   
   The right window pane displays all procedures in the paths defined for the selected project. Procedures that have been added to the project path have a color icon next to them, and the file type description is FOCUS Executable.

3. Do one of the following:
   
   - In the right window pane, right-click a procedure and choose *Remove from Project* from the pop-up menu.
     
     or
   
   - Select a procedure and click the *Delete* button on the Explorer toolbar.

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

You can drag any component of a project, including a procedure, between projects or WebFOCUS environments, or within a single project. When a procedure is copied within a project, it is identified as `procedure_name1`. A second copy is identified as `procedure_name2`, and so on.

Procedure: How to Copy a Procedure Within a Project

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

Creating a Procedure Component

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

Several of the components on the toolbar are usually placed in a specific order based on the logical flow of a procedure. These components are Set, Allocation, Use, Join, Define, and Report or Graph, in that order. The remaining components (Execute, Include, and Other) enhance the user interface and efficiency, and can be placed anywhere in the order of execution. However, Dimension must be placed before the Report component.

You can have one or several components in a procedure. The following image includes a Comment, four instances of a Report, a Use, Define, and Other components. Together these components make up the procedure named APP6.
### Component Types

You can include the following components in a procedure.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Creates a tabular representation of the data retrieved from a data source. A Report is placed after a Join and a Define. For details, see the <em>Creating Reports Overview</em> on page 129.</td>
</tr>
<tr>
<td>SQL Report</td>
<td>The SQL Report Wizard assists you with SQL passthru, which allows you to execute SQL code that retrieves data from an RDBMS. You can use the resulting extract file in Report Painter or InfoAssist. For details, see <em>Using the SQL Report Wizard</em> on page 233.</td>
</tr>
<tr>
<td>Define</td>
<td>Creates a temporary (virtual) field that is evaluated before a report is run. A Define is placed before a Report component. For details, see <em>Creating Temporary Fields</em> on page 273.</td>
</tr>
<tr>
<td>Define Function</td>
<td>Creates a virtual field function, which can be called in most of the same situations that are valid for Information Builders supplied functions.</td>
</tr>
<tr>
<td>Graph</td>
<td>Creates a graphical representation of the data retrieved from a data source. A Graph is placed after a Join and a Define.</td>
</tr>
<tr>
<td>Set</td>
<td>Customizes the Developer Studio and WebFOCUS environments, including the way reports and graphs display, report and graph content, data retrieval characteristics that affect performance, system responses to end user requests, metadata setup, and date manipulation. For details on the Set tool and parameters, see <em>Customizing Your Environment</em> in the <em>Developing Reporting Applications</em> manual.</td>
</tr>
<tr>
<td>Join</td>
<td>Defines a relationship between two or more data sources so that a report can use data from all of them at once. A Join is placed before a Define, and if there is no Define, it is placed before a Report component. For details, see <em>Joining and Merging Data Sources</em> on page 141.</td>
</tr>
</tbody>
</table>
### Component Description

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| Use       | Provides flexibility in using a FOCUS data source. You can specify read-only access, treat multiple data sources as one, and request the use of data sources not stored in your current directory (whether elsewhere on your hard disk or on a network file server).  
For details on accessing FOCUS data sources, including techniques for maintaining Use commands, see Accessing a FOCUS Data Source in the Developing Reporting Applications manual. |
| Allocation | Assigns a temporary name to a file created and used by WebFOCUS.  
To preserve an extract file (created by the HOLD command) on the servers disk, you must issue a FILEDEF command to a specific location. Also, the Dialogue Manager -READ and -WRITE commands refer to files by names that must first be established with a FILEDEF command.  
For details on the Allocation Wizard, see Assigning a Logical Name With the Allocation Wizard on page 241. |
| Execute   | Enables one procedure to call another. You can select a procedure to be called by the one you are editing.  
An Execute component can be placed anywhere in a procedure.  
For details on the Execute component, see Calling a Procedure From the Current One on page 259. |
| HtmlForm  | Creates a text file you can use to add HTML code to the procedure currently being edited. HTML forms are used to launch applications in the web environment.  
**Note:** The Deploy Wizard guides you through the process of creating a configuration that manages the deployment of your project files, including HTML forms, to the web.  
For more information on the Deploy Wizard, see the Developer Studio Application Development Getting Started manual. |
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include</td>
<td>Allows you to use a portion of a procedure if you store it separately. A separate, included procedure is treated as part of the procedure currently being edited. It enables the reuse of code, making projects easier to write and maintain. After an included procedure runs, control returns to the calling procedure. An Include component can be placed anywhere in a procedure. For details on the Include component, see <em>Calling a Procedure From the Current One</em> on page 259.</td>
</tr>
<tr>
<td>Other</td>
<td>Enables you to enter commands that connect other components. This feature is useful for writing code that does not have a graphical tool (such as Dialogue Manager). However, you can also use Other for any kind of manual coding, for example, to code a virtual field or a report. When used this way, the Other component automatically converts to the corresponding component type. An Other component can be placed anywhere in a procedure. For information on the Other component, see the <em>Developer Studio Application Development Getting Started</em> manual. For details on the WebFOCUS language, see the <em>Creating Reports With WebFOCUS Language</em> manual.</td>
</tr>
<tr>
<td>Olap Dimensions</td>
<td>Creates an OLAP hierarchy and dimensions based on data for multidimensional analysis without changing the Master File. The new logical view is saved as part of the procedure. A Dimension component applies to FOCUS data sources and relational tables. It must be placed before the Report component. For information on defining dimensions, see the <em>Describing Data With Graphical Tools</em> and the <em>Describing Data With WebFOCUS Language</em> manuals. Unlike the Dimension Builder (addressed in the <em>Describing Data With Graphical Tools</em> 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.</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| SQL Editor  | Enables you to code SQL Passthru and highlights any SQL commands within the code. This editor also allows you to choose your connection engine and connection name from drop-down lists on the toolbar.  
For more information on the SQL Editor, see the Developer Studio Application Development Getting Started manual. |
| Engine      | The Engine tool exposes the FOCUS ENGINE SET commands and enables you to enter ENGINE commands or connection attributes, and override parameters.  
**Note:** The Engine tool is only used to create ENGINE statements. You are responsible for having the knowledge of the ENGINE SET command or the Connection Attribute needed. For details, see the Using the Engine Tool on page 268. |
| Match       | The Match Wizard enables you to merge two or more data sources, and specify which records to merge and which to sort out. The wizard creates a new data source, a HOLD file, into which it merges fields from the selected records. You can report from the new data source and use it as you would any other HOLD file.  
For more information on the Match Wizard, see the Developer Studio Application Development Getting Started manual. |
| Dialogue Mgr| Enables you to control the flow of your application with the use of variables.  
For more information on the Dialogue Manager, see Managing Flow of Control on page 270. |
| 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.  
For details on Modify, see Unique FOCUS Topics on the WebFOCUS Documentation CD. |
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment</td>
<td>Includes a comment for the project. This comment is not necessary and you can delete it if you wish. A procedure shell, when created, contains a default Comment component. Once you create an executable component for a procedure, the file name defaults as a comment in this component. For example, if you named the shell TEST, the default Comment component will contain the comment `- File TEST.FEX. For information on the Comment component, see the Developer Studio Application Development Getting Started manual.</td>
</tr>
</tbody>
</table>

**Procedure:** How to Create a Component in an Existing Procedure

1. Right-click a procedure in the Procedures folder and choose Open. The Procedure window opens.
2. Click a component connector. The Component Connector toolbox opens, as shown in the following image.

![Component Connector Toolbox](image)

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

The corresponding tool opens. For details, see *Component Types* on page 225.

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

**Working With a Component**

You can perform the following tasks on a component:
Running. You can run some types of components, for example, a report or graph. This shows you how the output appears from a visual point of view.

Editing. You can edit a component within a graphical tool or with the Text Editor.

Using the Text Editor requires that you have some knowledge of the WebFOCUS language. You must be familiar with the types of commands, how to use them, and where to place them in a request.

Checking. Checking a component enables you to locate errors. Each component tool includes a check button that checks the syntax of the code against the server. For more information, see How to Check a Component on page 232.

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

Procedure: How to Run a Component

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

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 syntax. For information about the Text Editor, see Editing Application Components as Text in Developer Studio on page 369.
**Procedure: How to Check a Component**

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

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

![Check window for Define tool](image)

3. Click OK to return to your component tool.

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

1. Open both procedures by right-clicking them and choosing Open from the shortcut menu.

2. Choose Tile from the Window menu to view both Procedure windows simultaneously.
3. In the first Procedure window, click and hold the component you wish to copy.
4. While holding the left mouse button, press the Ctrl key.
5. Drag the mouse pointer to the second Procedure window.

**Note:** You must press the Ctrl key to copy. Not pressing Ctrl before dragging moves the component.

6. Release the mouse button when the mouse pointer is over the second Procedure window. The component is copied to the second window.

When you drag a component, its icon changes to show whether its target location is valid. If it is not valid, the cursor resembles a circle with a slash through the word *Drop*. When you have dragged the component to a valid location, the cursor resembles a small square with a plus sign (+) inside it.

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

---

### Using the SQL Report Wizard

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

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

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

---

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

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

In the Procedure Viewer, click the component connector and then click SQL Report on the drop-down list.

You can click the SQL Report icon on the Procedure Viewer toolbar.

The SQL Report Wizard - Welcome window opens, as shown in the following image.

![SQL Report Wizard - Welcome](image)

The SQL Report Wizard - Welcome window opens, as shown in the following image.

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

**Note:** You can only browse the files on the APP PATH if your .sql extension is a valid filter in the Properties dialog box. To do this, see *How to Add an .sql Extension as a Valid Filter* on page 240.
3. Click Next to see the SQL Report Wizard - Data access information window, as shown in the following image.

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

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

![SQL Report Wizard - Include external SQL file](image)

7. Enter the external SQL file name in the field, or click Browse to select it.

**Note:** You can only browse the files on the APP PATH if your .sql extension is a valid filter in the Properties dialog box. To do this, see How to Add an .sql Extension as a Valid Filter on page 240.

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

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

Do one of the following:

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

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

10. Click Finish to run the SQL procedure. When you have completed the procedure, you can run it from the Procedures folder in the Explorer view.
Procedure: How to Pass SQL Commands to the RDBMS Using SQL Passthru

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

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

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

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

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

![SQL Report Wizard - Enter SQL statements](image)

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

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

9. Click **Run SQL** to run your report.

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

   Do one of the following:

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

11. Click **Finish** to run the SQL procedure. When you have completed the procedure, you can run it from the Procedures folder in the Explorer view.

**Procedure: How to Import SQL Commands From an Existing .sql File**

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

In the Procedure Viewer, click the component connector and then click SQL Report on the drop-down list.

You can click the SQL Report icon on the Procedure Viewer toolbar.

The SQL Report Wizard - Welcome window opens.

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

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

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

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

6. Click Next to see the SQL Report Wizard - Import external SQL file window, as shown in the following image.

7. Type the SQL file name that you want to import or click Browse to select it.
8. Optionally, you can run with limited records by clicking Run SQL. By default, the Run with limited records check box is selected so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. 100 is the default limit.

9. Click Run SQL to run your report.

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

11. Click Run SQL to run your report.

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

   Do one of the following:
   
   - To create a report, select the Create Report option button. This option is selected by default.
   
   - To create a graph, select the Create Graph option button.

13. Click Finish to run the SQL procedure. When you have completed the procedure, you can run it from the Procedures folder in the Explorer view.

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

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

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

2. Scroll through the list until you find the .sql extension in the Extensions column.

   If the .sql file is not part of the displayed list in the Extensions column, click the Add new filter file type filter(s) icon.

3. Select the .sql extension.

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

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

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

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

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

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

- **Redirecting input and output.** Three ddnames are used for input and output. By reassigning these ddnames, you can redirect input or output:
  - SYSIN for input.
  - SYSPRINT for output displayed on the screen.
  - OFFLINE for output sent to the printer.

You can also use the Universal Naming Convention (UNC) to assign logical names to files that are located on a server. In order to take advantage of the UNC you must first attach to the server you want to use. For information on attaching to a server or mapping network drives, consult your Network Administrator.

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

1. Right-click the procedure in the Procedures folder and choose *Edit in Developer Studio Tool* from the shortcut menu.

   The Procedure (FOCEXEC) window opens.
2. Click a component connector (yellow diamond), and select Allocation from the Component Connector toolbar.

The Allocation Wizard opens, as shown in the following image.

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

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

5. Click Next to select additional options.

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

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

6. When selecting Disk as the device:
   a. Select the application folder that contains the physical file.
Note: The application folders from the Reporting Server are shown by default and depend on the area from which the Allocation Wizard is invoked. For example, Local Projects, Data Servers, or Managed Reporting. You may specify a file in a directory that is not in the Reporting Server Application Path by using the Advanced option from the completed Allocation. For more information, see Advanced Allocation Dialog Box on page 257. The Application folders are shown in the following image.
b. Click Next and enter the new file name, as shown in the following image.

or

Click the *Browse* button to display the Open dialog box and select from the list of existing files, as shown in the following image.

**Note:** Data files (*.dat) is the default value for the files of type drop-down list. You may also select Temporary Files (*.ftm), and All Files (*.*) as the type of file.
c. Click Next to specify additional options for the Disk device, as shown in the following image.

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

7. When selecting Printer as the device:
   a. Select the application folder that contains the physical file.
**Note:** The application folders from the Reporting Server are shown by default and depend on the area from which the Allocation Wizard is invoked. For example, Local Projects, Data Servers, or Managed Reporting. You may specify a file in a directory that is not in the Reporting Server Application Path by using the Advanced option from the completed Allocation. For more information, see *Advanced Allocation Dialog Box* on page 257. The application folders are shown in the following image.
b. Click Next and enter the new file name, as shown in the following image.

or

Click the Browse button to display the Open dialog box and select from the list of existing files, as shown in the following image.

Note: Data Files (*.dat) is the default value for the Files of type drop-down list. You may also select Temporary Files (*.ftm), and All Files (*.*), as the type of file.
c. Click Next to enter the name of the printer, as shown in the following image.

d. Click Next to view the summary of the allocation settings you selected for the Printer device, as shown in the following image.
Note: The following syntax is applied to the FILEDEF statement when using a Printer device:

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

8. When selecting Http as the device:

   a. Enter the full URL path, as shown in the following image.

   ![Allocation Wizard](image)

   - Enter the full URL:
     - http://server:port/dias/file
b. Click Next to view the summary of the allocation settings you selected for the Http device, as shown in the following image.

9. When selecting Terminal as the device:
   a. Click Next to specify additional options for the Terminal device.
You may choose to send the keyboard input to WebFOCUS as entered (lowercase or mixed-case). The default is mixed-case. Select this option to specify lowercase, as shown in the following image.
b. Click Next to view the summary of the allocation settings you selected for the Terminal device, as shown in the following image.

10. When selecting Clear as the device:

a. Click Next to view the summary of the allocation settings that you cleared, as shown in the following image.
11. Click *Finish* to close the Allocation Wizard and create the FILEDEF command.

The completed file name appears for the Allocation, as shown in the image below.

![Allocation Wizard](image)

12. Optionally, click *Check* to validate the FILEDEF command.

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

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

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

The Advanced Allocation dialog box enables you to enter the FILEDEF command without using the Allocation Wizard. For details, see *Advanced Allocation Dialog Box* on page 257.

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

The Allocation component is added to the procedure. You may double-click the component to view the completed Allocation Wizard options again.
Reference: Advanced Allocation Dialog Box

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

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

![Advanced Allocation Dialog Box](image)

**Name**

Shows the logical (defined) name of a file or device (one to eight characters).

**Device**

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

- **Disk** associates the specified logical name with a file.
- **Printer** associates the specified logical name with a printer.
- **Http** associates the Hypertext Transfer Protocol link, by allocating the Master File to the result of running the URL.
- **Terminal** indicates that the keyboard and monitor are the input source and output destination for the file.
Clear clears the FILEDEF command assigned to a file name. (Select the logical name in the list below the Name box.)

File name

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

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

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

Fixed

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

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

Record Size

Specifies the record length of the file.

Append

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

Lower Case

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

OK

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

Cancel

Closes the Advanced Allocation dialog box without saving the allocation settings.

Help

Opens the online documentation for this topic.

Check

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

New

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

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

Clearing Allocations

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

Procedure: How to Clear an Allocation With the Allocation Wizard

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

Procedure: How to Clear an Allocation With the Advanced Allocation Dialog Box

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

Calling a Procedure From the Current One

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

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

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

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

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

**Procedure: How to Call a Procedure With the Include Component**

1. Open the host procedure.
2. Click a 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 code.
**Procedure:** How to Execute a Procedure With the Execute Wizard

1. Click the component connector at the point at which you wish to include the existing procedure.
   The Component Connector toolbox opens.

2. Click the **Execute** button.
   The Execute Procedure Wizard opens, as shown in the following image.

3. Select the procedure you want to execute from the list of available procedures, and click **Next**. The Execute Procedure Wizard - Enter Procedure Parameters window opens, as shown in the following image.
The Enter Procedure Parameters window contains a list of variables contained in the procedure you are executing, and fields in which to enter values for those variables. This screen will not appear if the procedure you are calling does not have parameters.

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

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

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

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

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

**Example:** **Calling a Procedure From Another Procedure**

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

**Create the SALESREP procedure:**
1. Create a new procedure named SALESREP.

2. From the Create with drop-down list, select Procedure Viewer. The Procedure Viewer opens.


   a. Enter CITY in the Name input field.
   b. From the Accept List area, select the Static list radio button.
   c. From the Variable Type drop-down list, select the Single Select option.
   d. In the Data Context area, select Values for field.
   e. Click the Select a field ellipsis to get the Value Retrieval dialog box.
   f. Select the field CITY and then click Get Data.
   g. Double-click the returned values to add them all to the Accept List.
   h. Click OK to close the Variable Editor.
   i. The new variable is visible in the Object Inspector by expanding the Variables folder and then the Report Variables.

5. Insert a Report Heading and type Monthly Report for and add Report Variable &CITY by dragging it from the Object Inspector. When prompted to insert a variable as a field to be evaluated or as a literal value (default option), select the default option of Insert as literal value.

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

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

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

9. Filter the PROD_CODE field by doing the following:
   a. Select the Where/If tab at the bottom of the Object Inspector.
   b. Right-click the WHERE Conditions folder and select Where. The Expression Builder opens.
10. Add the CITY field to your report, and click By.

11. Filter by the CITY field by doing the following:
   a. Select the Where/If tab at the bottom of the Object Inspector.
   b. Right-click the WHERE Conditions folder and select Where. The Expression Builder opens.
   c. Double-click CITY. CITY appears in the Column to filter field.
   d. Select equals from the Logical Relation drop-down menu.
   e. Select Parameter from the Compare Type drop-down menu.
   f. Double-click the Compare Value field. The Variable Editor opens.
   g. Click OK. &CITY is entered in the Compare Value field.
   h. Click OK.

12. Save and close the procedure.

**Create the SVALUES procedure:**

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

2. Click a component connector, and select Execute. The Execute Wizard opens.

3. Select SALESREP from the list of procedures, and click Next.
The Enter Procedure Parameters window opens, as shown in the following image.

4. Select **STAMFORD** from the Value drop-down menu for **CITY**.

5. Enter **B10** in the Value field for **CODE1**.

6. Enter **B20** in the Value field for **CODE2**.

7. Click **Next**.
The Summary window opens, as shown in the following image.

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

9. Close and save the procedure.

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

Using the Engine Tool

The Engine tool exposes the FOCUS ENGINE SET commands and enables you to enter ENGINE commands or connection attributes, and override parameters.
Procedure: How to Create an Engine Statement

1. From the Procedure Viewer, click a component connector (yellow diamond), then click the Engine button on the Component Connector toolbox.

   or

   Select Engine from the Insert menu.

   The ENGINE dialog box opens.

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

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

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

5. Click OK to save the ENGINE statement in the Procedure Viewer.

   You may click the ENGINE file to update the existing ENGINE statement.

Reference: ENGINE Dialog Box

The following image shows the Engine dialog box.

![ENGINE Dialog Box Image]

The ENGINE dialog box has the following fields:

**Engine**

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

**Connection**

Contains any existing connections found in your environment.
**SET parameters**

Type a parameter statement.

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

**Managing Flow of Control**

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

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

Dialogue Manager commands are stored in procedures. Their execution can be triggered by actions on a form, such as clicking a button. Procedures containing Dialogue Manager commands can be reused by other procedures in a project.

The ability of Dialogue Manager commands to display forms and the ability of form objects to call Dialogue Manager commands are two of the most important aspects of application development. The most powerful applications are a combination of graphical and command-driven components.

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

**Working With a Full Procedure**

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.

- **Scheduling a report for distribution.** You can schedule a report for distribution to a single user or multiple users at specific times.
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 271.

- **Create a web-based user interface.** You can use HTML Composer to create a launch page (web-based interface) that makes a report available to browser users.

- **Securing procedure 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 a Procedure* on page 271.

- **Editing a procedure.** You can edit a procedure as text. See *How to Edit a Procedure as Text* on page 271.

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

**Procedure:** *How to Access ReportCaster to Schedule a Report*

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

   The ReportCaster Explorer opens.

**Procedure:** *How to Encrypt a Procedure*

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

2. Select the *Encrypted* check box, and click *Apply*.

**Procedure:** *How to Decrypt a Procedure*

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

2. Deselect the *Encrypted* check box, and click *Apply*.

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

Right-click the procedure and select *Edit As Text* from the shortcut menu, or select the procedure and choose *Edit As Text* from the File Menu. The Text Editor opens, displaying the syntax of the procedure.
Running a Procedure

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

Procedure: How to Run a Procedure on a Local Server

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

Canceling a Running Procedure

You can cancel any procedure while it is running. This may be necessary if you run a procedure and find that it takes too long to process or that it is the wrong one. To cancel a procedure at any time, you must use the Web Console. For more information, see the Server Administration for UNIX, Windows, OpenVMS, IBM i, and z/OS manual.
Creating Temporary Fields

When you create a report, you are not restricted to the fields that exist in your data source. If you can generate the information you want from the existing data, you can create a temporary field to evaluate and display it. A temporary field takes up no storage space in the data source. It is created only when needed.

**In this chapter:**

- What Is a Temporary Field?
- Defining a Virtual Field
- Creating a Calculated Value
- Creating Temporary Fields Independent of a Master File

**What Is a Temporary Field?**

A temporary field is a field whose value is not stored in the data source, but can be calculated from the data that is there, or assigned an absolute value. A temporary field takes up no storage space in the data source, and is created only when needed.

When you create a temporary field, you determine its value by writing an expression. You can combine fields, constants, and operators in an expression to produce a single value. For example, if your data contains salary and deduction amounts, you can calculate the ratio of deductions to salaries using the following expression:

\[ \text{deduction} \div \text{salary} \]

You can specify the expression yourself, or you can use one of the many supplied functions that perform specific calculations or manipulations. In addition, you can use expressions and functions as building blocks for more complex expressions, as well as use one temporary field to evaluate another.

**Reference:** Types of Temporary Fields

You can use two types of temporary fields (a *virtual field* and a *calculated value*), which differ in how they are evaluated:
A virtual field (Define) is evaluated as each record that meets the selection criteria is retrieved from the data source. The result of the expression is treated as though it were a real field stored in the data source. A virtual field is in effect until it is cleared.

A calculated value (Compute) is evaluated after all the data that meets the selection criteria is retrieved, sorted, and summed. Therefore, the calculation is performed using the aggregated values of the fields.
Reference: Evaluation of Temporary Fields

The following illustration shows how a request processes, and when each type of temporary field is evaluated:

1. Locates the Master Files and data sources.
2. Retrieves records that meet selection criteria.
3. Determines values of virtual fields (DEFINE).
4. Applies filtering criteria to virtual field values.
5. Sorts the data.
6. Applies a display command.
7. Determines calculated values (COMPUTE).
8. Applies filtering criteria to total values or calculated values.
10. Formats the report.
11. Sends the report to a screen, printer or file.
Distinguishing Between Virtual Fields and Calculated Values

Example: In the following example, both the DRATIO field (virtual field) and the CRATIO (calculated value) use the same expression DELIVER_AMT/OPENING_AMT, but do not return the same result. The value for CRATIO is calculated after all records have been selected, sorted, and aggregated. The virtual field DRATIO is calculated for each retrieved record.

```
DEFINE FILE SALES
DRATIO = DELIVER_AMT/OPENING_AMT;
END
TABLE FILE SALES
SUM DELIVER_AMT AND OPENING_AMT AND DRATIO
COMPUTE CRATIO = DELIVER_AMT/OPENING_AMT;
END
```

The output is:

<table>
<thead>
<tr>
<th>DELIVER_AMT</th>
<th>OPENING_AMT</th>
<th>DRATIO</th>
<th>CRATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>760</td>
<td>724</td>
<td>28.41</td>
<td>1.05</td>
</tr>
</tbody>
</table>

Reference: Selecting a Temporary Field

The following is to help you choose the kind of temporary field you need.

**Choose a virtual field** when you want to:

- Use the temporary field to select data for your report. You cannot use a calculated value, since it is evaluated after data selection takes place.
- Use the temporary field to sort on data values. A calculated value is evaluated after the data is sorted. With the BY TOTAL phrase, you can sort on this type of field.

**Choose a calculated value** when you want to:

- Evaluate the temporary field using total values or prefix operators (which operate on total values). You cannot use a virtual field, since it is evaluated before any totaling takes place.
- Evaluate the temporary field using fields from different paths in the data structure. You cannot use a virtual field, since it is evaluated before the relationship between data in the different paths is established.
Defining a Virtual Field

A virtual field can be used in a request as though it is a real data source field. The calculation that determines the value of a virtual field is performed on each retrieved record that passes any screening conditions on real fields. The result of the expression is treated as though it were a real field stored in the data source.

You can define a virtual field in the following ways:

- **In a Master File.** These virtual fields are available whenever the data source is used for reporting. These fields cannot be cleared by JOIN or DEFINE FILE commands.
  
  For more information, see the manual.

- **In a procedure.** A virtual field created in a procedure lasts only for that procedure.

**Tip:** If your environment supports the KEEPDEFINES parameter, you can set KEEPDEFINES to ON to protect virtual fields from being cleared by a subsequent JOIN command.

Reference: Usage Notes for Creating Virtual Fields

- When a JOIN is issued, all pre-existing virtual fields for that data source are cleared except those defined in the Master File.

- To join structures using a virtual field with the source, make sure the Define follows the Join. Otherwise, the Join clears the temporary field. For an explanation of reporting on joined data sources, see *Joining and Merging Data Sources* on page 141.

- If you edit a DEFINE and change the name, you will still see the original DEFINE, in addition to the new named DEFINE.

- If no field in the expression is in the Master File or has been defined, use the WITH command to identify the logical home of the defined calculation.

- WITH can be used to move the logical home for the virtual field to a segment lower than that to which it would otherwise be assigned (for example, to count instances in a lower segment).

- You may define fields simultaneously (in addition to fields defined in the Master File) for as many data sources as desired. The total length of all virtual fields and real fields cannot exceed 32,000 characters.

- When you specify virtual fields in a request, they count toward the display field limit.

- Virtual fields are only available when the data source is used for reporting. Virtual fields cannot be used with MODIFY.
A DEFINE command may not contain qualified field names on the left-hand side of the expression. If the same field name exists in more than one segment, and that field must be redefined or recomputed, use the REDEFINED command.

Using a self-referencing DEFINE such as x=x+1 disables AUTOPATH (see the manual).

Procedure: How to Create a Virtual Field

To define a virtual field (Define object) in a procedure:

1. Right-click the procedure in the Procedures folder and choose Edit in Developer Studio tool from the context menu. The Procedure Viewer opens.

2. Click a component connector (yellow diamond) at the point where you want to include the virtual field in the procedure, then click the Define button on the component connector toolbar.

3. The Open dialog box opens. Select the Master File from which you want to create the virtual field and click Open.

   The Define dialog box opens. For details see Define Tool on page 281.

4. Type the name of the virtual field in the Field input box.

5. Enter an expression in the expressions box. See specific procedures for details on assigning alphanumeric, numeric, date, and CLOB formats.

   For information on expressions, see the Creating Reports With WebFOCUS Language manual.

   You can also use a function in an expression by clicking the Functions button on the Define dialog box. For details on functions, see the Using Functions manual.

Note: If you want to make a virtual field available for use with all procedures during your session, issue the DEFINE command from the Command Console. The virtual field is retained for the duration of the session.

Procedure: How to Assign an Alphanumeric Format With the Define Tool

In the Define tool window:

1. Click the Format button.

   The Format dialog box opens.

2. Select Alphanumeric as the Format Type.

   The default length is 20. It appears in the Length input area.

3. To assign a different length, specify a number between 1 and 4096 in the Length input area.
4. Click OK.

The Format dialog box closes.

For more information on field formats, see Assigning Field Formats in the Creating Reports With Report Painter manual.

**Procedure:** How to Assign a Numeric Format With the Define Tool

1. Click the Format button.

   The Format dialog box opens.

2. Select one of the format option buttons in the Format Types section:

   The default length appears in the Length input area. The Decimal input area shows the number of decimal places for Floating Point, Decimal, and Packed.

3. To assign a different length, specify a number between 1 and 9 for Floating Point, between 1 and 11 for an Integer, between 1 and 20 for Decimal, or between 1 and 33 for Packed in the Length input area.

4. To assign a different number of decimal places for Floating Point, Decimal, or Packed, specify the desired number in the Decimal input area.

5. If you wish to include numeric display options, select these options in the Edit Options section.

6. Click OK.

   The Format dialog box closes.

For more information on field formats, see Assigning Field Formats in the Creating Reports With Report Painter manual.

**Procedure:** How to Assign a Date/Time Format With the Define Tool

In the Define tool window:

1. Click the Format button.

   The Format dialog box opens.

2. Click the Date/Time button from the Format Type section.

   The Date and Time Formats dialog box opens.

3. Select Date as the Field Format.

   The Date section options are enabled.
4. Select a date format from the Format drop-down list.
   When a date format is selected, the Format String and Sample Date are displayed for the format selected.

5. Optionally, you may select the Month Name, Day Name, Prefix, and Separator options.

6. Click OK.
   The Date and Time Formats dialog box closes. The selected format appears in the Date/Time Format section of the Format dialog box.

7. Click OK to close the Format dialog box.

**Note:** Administrators and developers should note that date options set in a virtual field will override date options set in the Master File or with a SET command.
Reference: Define Tool

The following image shows the Define tool.

The Define tool has the following fields or options:

**Field**

Displays the name of the temporary field.

To create a field, type the name that you want to use in the Field text box.

To edit a field, select the field. When you select a field, the corresponding information appears in the Format box, and the expression box. You can edit the information, eliminate the field, or run it.

**Add**

When selected, it indicates that the current temporary field definitions should be added to the list of temporary fields previously defined for the same data source.

If this button is not selected, the current set of temporary fields replaces those previously defined for the same data source.
In Application View, this action marks all definitions created during the current use of the Define tool to be added to other virtual fields defined for the same data source during earlier use of the tool.

**Note:** You can define and execute several virtual fields during a single use of the Define tool. However, unless you select Add, running the current list will erase other virtual fields created outside of the Master File for the same data source. Virtual fields created in the Master File remain in effect.

**Clear**

When selected, it clears any current temporary field definitions you previously added to the list of temporary fields.

**Format button**

Opens the Format dialog box, where you assign a format to the temporary field.

**Format box**

Displays the field type, field length, and display options. The field type can be alphanumeric, numeric, or date/time.

**Options button**

Opens the Options window, which enables you to establish a segment location for a temporary field in the associated Master File, and/or assign attributes to set how missing values in the virtual field are handled.

**Expressions box**

Displays the expression used to evaluate the field.

Type the expression or use the Fields list, calculator, and functions list to help you create the expression.

**Calculator buttons**

Provide numbers and operators that you can use to create numeric, alphanumeric, Boolean, and conditional expressions.

Click the desired number or operator to add it to the expression box.

- To enclose a value in parentheses, click the ( ) key in the calculator. Parentheses affect the order in which the specified operations are performed.

- To enclose a value in single quotation marks, click the ' ' key in the calculator. Use single quotation marks to enclose alphanumeric and date literals.
To convert entries in the expression box to uppercase, click the U key in the calculator. Note that field names are case-sensitive.

**Check**

Displays the Define phrase in code, and specifies any errors.

**New**

Clears the entry fields of the tools, including the Field text box and the corresponding expression. It also returns the format to the default value D12.2, and places the cursor in the Field text box so you can begin to create a new field.

**Delete**

Deletes the temporary field identified in the Field text box. The field is no longer available.

**Functions**

Opens the Function Arguments dialog box, which lists all available built-in functions. (A function is a program that returns a value.)

Double-click the desired function to add it to the expression box. Then, in the expression box, highlight each argument and substitute the value or field name you wish to use. For details, see the *Using Functions* manual.

**Defined Fields**

Lists the names of the temporary fields already associated in the Master File.

**Fields List**

Lists the fields defined in the Master File.

---

**Example:**  **Creating a Virtual Field**

Using information in the sample data source EMPLOYEE, the following example shows how to create a virtual field, INCREASE, to calculate the annual salary increase each employee will receive. This example assumes that you have already created a procedure with which you want to use this virtual field.

1. Open a procedure in which you want to create a virtual field.
2. Select the EMPLOYEE Master File and click Open.
3. Open the Define tool from the component connector toolbar.
4. Type *INCREASE* in the Field input box.
5. Click the Format button.
The Format dialog box opens.

6. Confirm that the *Decimal* option button is selected under Format Types.

7. Click the down arrow in the Length input area to specify the field length. Leave the number 2 in the Decimal field.

8. Select the *Floating dollar--M* option in the Edit Options list box.

9. Click OK to make the changes and return to the previous dialog box.

10. Click the *Fields List* tab, then double-click *CURR_SAL*.

11. Enter the following by typing or using the number and operator buttons in the Expressions window:

   \[ * \ .05 \]

   The expression for the Define field now appears as *CURR_SAL* \* .05.

12. Click OK.

You can then select the new field, INCREASE, in the Fields window of the reporting tools.

**Procedure:**  **How to Specify Missing Value Attributes Using the Define Tool**

1. Open the Define tool by:

   - Select *Define* from the component connector toolbar in the Procedure Viewer.
   
   or

   - From Report Painter, right-click anywhere in the Fields tab of the Object Inspector and select *New Define Virtual Field*.

   The Define tool opens.

2. Create a Define expression.
3. Click the Options button and select the **Override missing values handling** check box to specify how the missing value attributes are handled. The Options dialog box is shown in the following image.

4. Click **OK** to close the Options dialog box.

For more information about Missing Field Values, see *MISSING Attribute in a DEFINE or COMPUTE Command* in the *Handling Records With Missing Field Values* chapter of the *Creating Reports With WebFOCUS Language* manual.

**Procedure: How to Establish a Missing Segment Location Using the Define Tool**

1. Open the Define tool by:
   - Select Define from the component connector toolbar in the Procedure Viewer.
   - or
   - From Report Painter, right-click anywhere in the Fields tab of the Object Inspector and select New Define Virtual Field.

   The Define tool opens.

2. Create a Define expression.

3. Click the Options button.
4. Select the **Associate the temporary field with the real field that is selected** check box and select a field. The Options dialog box is shown in the following image.

![Options dialog box](image)

5. Click **OK** to close the Options dialog box.

For more information about Missing Field Values, see *Establishing a Segment Location for a Virtual Field* in the *Handling Records With Missing Field Values* chapter of the *Creating Reports With WebFOCUS Language* manual.

### Defining Multiple Virtual Fields

You may wish to have more than one set of virtual fields for the same data source, and to use some or all of the virtual fields in the request. The Add option enables you to specify additional virtual fields without clearing existing ones. If you omit the Add option, previously defined virtual fields in that data source are cleared.

If you want to clear a virtual field for a particular data source, use the Clear option.

**Procedure:** **How to Add a Virtual Field to a Set of Virtual Fields**

1. In the Define dialog box, type the field name in the Field input box.
2. Enter the desired expression in the expression box.

   For information on expressions, see the *Creating Reports With WebFOCUS Language* manual.

3. Select the **Add** option button.
4. Click OK.

**Displaying Virtual Fields**

You can display all virtual fields with the ? DEFINE command. You can also access this information in the Define tool.

**Procedure:** How to Display Virtual Fields

Click the Defined Fields tab in the Define tool.

**Clearing a Virtual Field**

The following can clear a virtual field created in a procedure:

- A new list of virtual fields for the same data source without the Add option selected.
- A join. When a join is created for a data source, all pre-existing virtual fields for that data source are cleared except those defined in the Master File. This may affect virtual fields used in an expression.
- A change in the value of the FIELDNAME SET parameter.

Unlike fields created in a procedure, virtual fields in the Master File are not cleared in the above ways.

**Procedure:** How to Delete a Virtual Field

In the Define tool window, with the field in the Field text box, click Delete.

The field is no longer available.

**Increasing the Speed of Calculations in Virtual Fields**

Virtual fields can be compiled into machine code in order to increase the speed of calculations. When you enable compilation of expressions, expressions in IF criteria are also compiled. In some cases, an expression in an IF test may contain too many elements to be compiled (the limit is 8192). In this case, a FOC1975 message is generated to indicate that the expression could not be compiled. However, the report completes successfully, and the expression is evaluated correctly without compilation.
Applying Dynamically Formatted Virtual Fields to Report Columns

Dynamic formatting enables you to apply different formats to specific data in a column by using a temporary field that contains dynamic data settings. You can create a dynamic format by writing an expression using the Define tool inside or outside of Report Painter.

Before you can format a report column using the dynamic format, you must create the report, then apply the temporary field to a column in the report. For example, you can create a temporary field that contains different decimal currency formats for countries like Japan (which uses no decimal places) and England (which uses 2 decimal places). These currency formats are considered dynamic formats. You can then apply the temporary field containing the dynamic formatting to a Sales column. In a report, the Sales column reflects the different currency formats for each country.

Procedure: How to Create a Virtual Field Containing Dynamic Formatting

In the Define tool:

1. Type the name of the virtual field in the Field input box.
2. Click Format and type A8 (Alphanumeric, 8 characters). Click OK.
3. Type the expression in the Expressions window or use the Fields list, calculator, or functions list to help you create the expression. The expression should contain the parameter values necessary for defining the dynamic formatting. All expressions must be written in uppercase.
4. Click OK.

You can apply dynamic formatting to a report column.

Procedure: How to Format a Report Column Using Dynamic Formats

1. Right-click a report column in Report Painter and select Format.
2. Click the Dynamic option button in the Format Types area.
3. Select the field with the desired dynamic format from the Format Field drop-down list.
4. Click OK to format the report column.

For more information about Report Painter, see the Creating Reports With Report Painter manual.

Example: Setting Appropriate Formats for Different Currencies

The following example demonstrates how the dynamic field format can be used to set different currency formats for specific countries in a report.
1. Open the procedure for which you want to create the virtual field, then open the Define tool from the component connector toolbar.

   The Open dialog box opens.

2. Select the CAR Master File and click Open.

   The Define tool opens.

3. Type MYFORMAT in the Field box.

4. Click the Format button. Select the Alphanumeric option button, and specify 8 as the length. Click OK.

5. Click the Functions button to open the Functions Arguments dialog box:
   a. Select DECODE from the Select a function drop-down list. The DECODE function appears with placeholders for its arguments.
   b. Select COUNTRY as the DECODE field name.
   c. Enter the actual and display values by typing the following placeholder fields with specific values that define the decimal precision for each currency: ENGLAND P15.2C JAPAN P15.0 ELSE P15.2M.
   d. Enter the default value P15.2M to be assigned if the code is not found among the list of codes.
   e. Click OK to return to the Define tool. The DECODE function appears as the Define expression.

6. Click OK to close the Define tool.

7. Select Report from the Connector toolbox. At the Open dialog box, select CAR from the list of Master Files and click Open.

8. In Report Painter, double-click the COUNTRY and SALES fields in the Fields list.

9. Select the SALES column and click the Sum button on the Columns toolbar.

10. Right-click the SALES column and select Format.

11. Click the Dynamic option button and select the MYFORMAT field from the Format Field drop-down list.

12. Click OK to apply the formats associated with the MYFORMAT field to the SALES column.

13. Select the COUNTRY column and click the By button on the Columns toolbar.
14. Run the report, as shown in the following image.

<table>
<thead>
<tr>
<th>COUNTRY</th>
<th>SALES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLAND</td>
<td>12,000.00</td>
</tr>
<tr>
<td>FRANCE</td>
<td>$0.00</td>
</tr>
<tr>
<td>ITALY</td>
<td>$30,200.00</td>
</tr>
<tr>
<td>JAPAN</td>
<td>78030.00</td>
</tr>
</tbody>
</table>

Notice that the SALES column displays the format variations you defined for each country in the DECODE function.

Creating a Calculated Value

A calculated value is a temporary field that is evaluated after all the data that meets the selection criteria is retrieved, sorted, and summed. Calculated values are available only for the specified report request.

Use the Computes tab to create a calculated value. You can access the Computes tab only when you are creating reports in Report Painter.

Reference: Usage Notes for Calculated Field Values

The following apply to the use of calculated values:

- If you specify any optional Compute expressions (such as, AS, IN, or NOPRINT), and you compute additional fields following these phrases, you must repeat the Compute expression before specifying the additional fields.

- You can rename and justify column totals and row totals.

- Expressions in a Compute expression can include fields with prefix operators. Fields referred to in a Compute expression are counted toward the display field limit, and appear in the internal matrix.

Procedure: How to Create a Calculated Value

In Report Painter:

1. Click Computes from the Report menu or click the Compute icon located on the Setup toolbar.
The Report Options dialog box opens at the Computes tab.

2. Type the field name in the Field input box.
3. Enter the desired expression in the expression box.
4. Click OK.

**Using Positional Column Referencing With Calculated Values**

In a COMPUTE command, it is sometimes convenient to refer to a field by its report column position, rather than its name. This option is especially useful when the same field is specified for several report columns.

Column referencing becomes essential when you are using the same field name in a variety of ways. The following image shows that columns produced by display commands (whether displayed or not) can be referred to as C1 for the first column, C2 for the second column, and so forth. The BY field columns are not counted.

<table>
<thead>
<tr>
<th>AVE DEALER_COST</th>
<th>COUNTRY</th>
<th>AVE DEALER_COST</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>7,989</td>
<td>ENGLAND</td>
<td>9,463</td>
<td>.84</td>
</tr>
<tr>
<td>FRANCE</td>
<td>4,631</td>
<td>1.73</td>
<td></td>
</tr>
<tr>
<td>ITALY</td>
<td>10,309</td>
<td>.77</td>
<td></td>
</tr>
<tr>
<td>JAPAN</td>
<td>2,756</td>
<td>2.90</td>
<td></td>
</tr>
</tbody>
</table>

**Using ACROSS With Calculated Values**

If the Compute expression is issued immediately following an Across phrase, only a recap type of the calculation is performed once for all columns.

**Sorting Calculated Values**

You can sort a report by a virtual field or a calculated value. To sort by a calculated value, you must use the BY TOTAL phrase in your request.

**Screening on Calculated Values**

You can screen on values produced by Compute expressions by using the WHERE TOTAL test.
Creating Temporary Fields Independent of a Master File

The temporary fields you create with the Define and Compute expressions are tied to a specific Master File, and in the case of values calculated with the COMPUTE command Compute expression, to a specific request. However, you can create temporary fields that are independent of either a Master File or a request using the DEFINE FUNCTION command.

A DEFINE function is a named group of calculations that use any number of input values and produce a return value. When calling a DEFINE function, you must first define the function.

A DEFINE function can be called in most of the same situations that are valid for Information Builders-supplied functions. Data types are defined with each argument. When substituting values for these arguments, the format must match the defined format. Alphanumeric arguments shorter than the specified format are padded with blanks, while longer alphanumeric arguments are truncated.

All calculations within the function are done in double precision. Format conversions occur only across equal signs (=) in the assignments that define temporary fields.

Syntax: How to Define a Function

DEFINE FUNCTION name (argument1/format1,..., argumentn/formatn)
[tempvariablea/formata [TITLE 'line1,[line2 ...']
[DESCRIPTION 'description'] = expressiona;]
.
.
[tempvariablex/formatx = expressionx;]
name/format = [result_expression];
END

where:

name

Is the name of the function, up to 64 characters. This must be the last field calculated in the function, and is used to return the value of the function to the calling procedure.

argument1...argumentn

Are the argument names. They can be any names that comply with FOCUS field naming rules.

format1...formatn

Are the formats of the function arguments.

If the format of an argument is alphanumeric, the argument value must also be alphanumeric. Shorter arguments are padded on the right with blanks, and longer arguments are truncated.
If the format of an argument is numeric, the argument value must also be numeric. To prevent unexpected results, you must be consistent in your use of data types.

`tempvariablea...tempvariablex`  
Are temporary fields. Temporary fields hold intermediate values used in the function. You can define as many temporary fields as you need.

`tempformata...tempformatx`  
Are the formats of the temporary fields.

`line1, line2 ...`  
Are the lines of default column title to be displayed for the virtual field unless overridden by an AS phrase.

`description`  
Is the description to be associated with the virtual field.

`expressiona...expressionx`  
Are the expressions that calculate the temporary field values. The expressions can use parameters, constants, and other temporary fields defined in the same function.

`format`  
Is the format of the value the function returns.

`result_expression`  
Is the expression that calculates the value returned by the function. The expression can use parameters, constants, and temporary fields defined in the same function.

**Reference: DEFINE Function Limits and Restrictions**

- The number of functions you can define and use in a session is virtually unlimited.
- A DEFINE function is cleared with the DEFINE FUNCTION CLEAR command. It is not cleared by issuing a join, or by any WebFOCUS command.
- When an expression tries to use a cleared function, an error appears.
- DEFINE functions can call other DEFINE functions, but cannot call themselves.
- If you overwrite or clear a DEFINE function, a subsequent attempt to use a temporary field that refers to the function generates the following warning:
  
  (FOC03665) Error loading external function '%1'
**Procedure:** How to Display DEFINE Functions

Issue the following command from the Command Console:

```
? FUNCTION
```

**Example:** Displaying DEFINE Functions

Issuing the command

```
? FUNCTION
```

displays information similar to the following:

```
FUNCTIONS CURRENTLY ACTIVE

NAME FORMAT PARAMETER FORMAT
--------- -------- --------- --------
SUBTRACT D8.2 VAL1 D8
         VAL2 D8
```

If you issue the `? FUNCTION` command when no functions are defined, the following appears:

```
NO FUNCTIONS CURRENTLY IN EFFECT
```

**Syntax:** How to Clear DEFINE Functions

```
DEFINE FUNCTION {name|*} CLEAR
```

where:

- `name`
  - Is the name of the function name to clear.

- `*`
  - Clears all active DEFINE functions.
Your reports and graphs appear in your viewer. When you run a report, a visual image of all of the data values that you have requested are collected and automatically appear in your viewer.

**In this chapter:**

- Viewing Reports in the Developer Studio Viewer
- Drilling Down for Details
- Navigating Through Multi-Page Web Reports
- Visualizing Trends in Reports

**Viewing Reports in the Developer Studio Viewer**

You can view HTML, PDF, and Excel reports in the Developer Studio Viewer.

**Example:** HTML Report in the Developer Studio Viewer

The following is a sample HTML report.
**Example:**  **PDF Report With Acrobat Reader Controls**

PDF format opens in Acrobat Reader, which is invoked by the Developer Studio Viewer. Unlike HTML, which does not observe certain styling options, PDF format shows all styles as you define them in Report Painter or in a report request.

The full range of Acrobat Reader navigation options are available through the Developer Studio Viewer. The following image shows a report in PDF format.
**Example:** Excel 2000 Report in the Developer Studio Viewer

The following image shows an example of a report in Excel 2000 format. When the report is run, Excel 2000 launches within your viewer, offering complete Excel 2000 functionality.

![Excel 2000 Report Example](image)

**Drilling Down for Details**

Your viewer supports drill-down functionality. That means you can drill down to other reports (located locally on your workstation or remotely on a server) or to URLs for details. After a drill-down link is defined, the "hot spots" in your report are underlined (similar to a hyperlink in a webpage). For details on linking to reports and URLs, see *Creating a Drill Down Procedure* in the *Creating Reports With Report Painter* manual.

**Navigating Through Multi-Page Web Reports**

Normally, after the web server receives the output from the WebFOCUS application server, it returns the entire report output to the viewer. The on-demand paging facility provides significant performance advantages by downloading one page of HTML report output to the viewer, instead of the entire report. When you request data using on-demand paging, the WebFOCUS Reporting Server receives the entire report, but sends only the first page to the viewer and holds the remaining pages on the servers disk, awaiting your signal to send other sections of the report.

Your viewer window is divided into two frames: the upper frame displays one page of report output. The lower frame provides controls that you use to display report pages and search report output for specific strings. For details, see *Navigating With On-Demand Paging* on page 300.
Note: This feature is only available if you are running HTML reports remotely on the WebFOCUS Reporting Server.

Example: HTML Report With On-Demand Paging

The following image shows HTML report output in your viewer using on-demand paging. Note that the first page appears in the main panel, with navigational controls in the panel.

Setting Up for On-Demand Paging

To set up for on-demand paging and to display report pages in your viewer you must do the following:

- **Format the report output as HTML.** On-demand paging requires that report output be formatted as HTML. If your report default is HTML format, you can simply create a new HTML report. If HTML is not the default, you can set format to HTML from Report Painter. See How to Format a Report as HTML on page 299.

- **Enable on-demand paging.** This setting activates on-demand paging for your report. You can enable on-demand paging through Report Painter. See How to Enable On-Demand Paging for HTML Reports on page 300.
Select a WebFOCUS Reporting Server to run the report on. You must select a WebFOCUS Reporting Server to see on-demand paging results. If you do not select a server, you will be prompted to select a server when you run the report.

Run the report. When you run the report, on-demand paging delivers the first page of the report output to the viewer and holds the remaining pages on the server disk.

A page of report output is determined by the number of lines per page. You can set the viewer to display a specified number of lines per page using the SET LINES command. You issue this command either using the Set tool or by manually including it in your request from the Text Editor or Command Console. If you do not specify a value for the SET LINES command, the viewer defaults to 57 lines per page. For more information on SET commands, see Customizing Your Environment in the Developing Reporting Applications manual.

Procedure: How to Format a Report as HTML

In Report Painter, click the Options button to open the Report Options dialog box. Select an HTML Styled report format from the select output format drop-down list.
**Procedure:** How to Enable On-Demand Paging for HTML Reports


  or

- In the Text Editor or Command Console, administrators and developers can add the SET WEBVIEWER=ON command to a procedure to enable the on-demand paging attribute. The SET command may be placed outside or within the TABLE request. See *Viewing Reports in the Developer Studio Viewer* on page 295 for an example.

**Example:** Report Procedure Syntax for On-Demand Paging

Administrators and developers can initiate on-demand paging using syntax, as shown in the following example:

```
SET WEBVIEWER=ON
TABLE FILE MOVIES
PRINT DIRECTOR WHOLESALEPR
BY TITLE
ON TABLE SET ONLINE-FMT HTML
END
```

The following report request generates the same report output. In this request, the SET commands are placed within the TABLE request.

```
TABLE FILE MOVIES
PRINT DIRECTOR WHOLESALEPR
BY TITLE
ON TABLE SET WEBVIEWER ON

ON TABLE SET ONLINE-FMT HTML
END
```

To see the output generated by these procedures, see *Viewing Reports in the Developer Studio Viewer* on page 295.

**Navigating With On-Demand Paging**

When on-demand paging is enabled, the viewer displays one page of report output and navigational controls. You use the navigational controls to change the report page that appears in your viewer.

The viewer consists of two frames, the Report Frame and the Viewer Control Panel frame. The Report Frame is the larger frame and contains one page of report output. When you first run a report, the Report Frame contains the first page of report output.
The Viewer Control Panel frame contains the controls you use to navigate through the report and to search for a string. The navigational controls allow you to display the next or previous page, the first or last page, or to display a specific page. You use the searching function to have the viewer locate a page of the report that contains a search string you specify.

**Procedure: How to Navigate Through a Report (Viewer Control Panel)**

The Viewer Control Panel offers several ways to view pages in your report:

- To display a specific page:
  - Enter a page number in the Page input box.
  - Click the Go to Page button.
- To display the previous or the next page in sequence, click the Previous or Next arrow button.
- To display the first or last page of the report, click the First Page or the Last Page button.

To download the entire report to your viewer as a single document, click the All Pages button. Your viewer displays the entire report without your viewer Control Panel. You can return to viewing a single page of your report by clicking the Back button on your viewer toolbar.

**Visualizing Trends in Reports**

To make your HTML, PDF, and PostScript reports more powerful, you can insert visual representations of selected data directly into the report output. These visual representations are in the form of vertical or horizontal bar graphs that make relationships and trends among data more obvious.

**Applying Bar Graphs**

Vertical or horizontal bar graphs highlight relationships and trends among data.
- **Vertical Bar Graph.** You can apply a vertical bar graph to report columns associated with an ACROSS sort field. The report output displays a vertical bar graph in a new row above the associated data values, as shown in the following image.

<table>
<thead>
<tr>
<th>Region</th>
<th>Midwest</th>
<th>Northeast</th>
<th>Southeast</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dollar Sales</td>
<td>Difference from Budget Dollar Sales</td>
<td>Difference from Budget Dollar Sales</td>
<td>Difference from Budget Dollar Sales</td>
<td>Difference from Budget Dollar Sales</td>
</tr>
<tr>
<td>$11,400,661.00</td>
<td>$11,392,310.00</td>
<td>$11,710,379.00</td>
<td>$11,652,957.00</td>
<td></td>
</tr>
<tr>
<td>$206,292.00</td>
<td>-$184,622.00</td>
<td>-$97,602.00</td>
<td>$11,444.00</td>
<td></td>
</tr>
</tbody>
</table>

Bar graphs that project above the zero line represent positive values, while bar graphs that project below the zero line represent negative values.

- **Horizontal Bar Graph.** You can apply a horizontal bar graph to report columns. The report output displays a horizontal bar graph in a new column to the right of the associated data values, as shown in the following image.

<table>
<thead>
<tr>
<th>City</th>
<th>Budget Dollars</th>
<th>Dollar Sales</th>
<th>DIFFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>$4,247,597.00</td>
<td>$4,100,107.00</td>
<td>$147,490.00</td>
</tr>
<tr>
<td>Boston</td>
<td>$3,818,397.00</td>
<td>$3,707,936.00</td>
<td>$110,411.00</td>
</tr>
<tr>
<td>Chicago</td>
<td>$3,866,856.00</td>
<td>$3,924,401.00</td>
<td>-$57,545.00</td>
</tr>
<tr>
<td>Houston</td>
<td>$3,690,679.00</td>
<td>$3,714,978.00</td>
<td>-$34,299.00</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>$3,669,484.00</td>
<td>$3,772,014.00</td>
<td>-$102,530.00</td>
</tr>
<tr>
<td>Memphis</td>
<td>$3,689,979.00</td>
<td>$3,687,057.00</td>
<td>$2,922.00</td>
</tr>
<tr>
<td>New Haven</td>
<td>$3,832,202.00</td>
<td>$3,782,049.00</td>
<td>$50,153.00</td>
</tr>
<tr>
<td>New York</td>
<td>$3,926,333.00</td>
<td>$3,902,275.00</td>
<td>$24,058.00</td>
</tr>
<tr>
<td>Orlando</td>
<td>$3,870,405.00</td>
<td>$3,923,215.00</td>
<td>-$52,810.00</td>
</tr>
<tr>
<td>San Francisco</td>
<td>$3,916,863.00</td>
<td>$3,870,258.00</td>
<td>$46,605.00</td>
</tr>
<tr>
<td>Seattle</td>
<td>$4,055,166.00</td>
<td>$4,010,685.00</td>
<td>$44,481.00</td>
</tr>
<tr>
<td>St. Louis</td>
<td>$3,646,838.00</td>
<td>$3,761,286.00</td>
<td>-$114,448.00</td>
</tr>
</tbody>
</table>

Bar graphs that project to the right of the zero line represent positive values, while bar graphs that project to the left of the zero line represent negative values.
The length of each vertical or horizontal bar graph is proportional to the magnitude of its associated data value. The shortest bar graph appears for the value with the minimum magnitude, the longest bar graph for the value with the maximum magnitude, and bar graphs of varying length appear for each value within the minimum-maximum magnitude range. Notice in the figure that a value of 147,490.00 produces a longer horizontal bar graph than a value of 50,153.00. Therefore, a complete row of vertical bar graphs or a complete column of horizontal bar graphs forms a bar chart.

You can only apply data visualization bar graphs to numeric report columns (integer, decimal, floating point single-precision, floating point double-precision, and packed). Bar graphs applied to alphanumeric, date, or text field formats are ignored.

You can display data visualization bar graphs in:

- OLAP-enabled HTML reports, where bar graphs are applied to measures. For more information on OLAP, see the *Online Analytical Processing (OLAP)* manual.

- Other HTML, PDF, or PostScript reports, where bar graphs are applied to columns. See *Associating Bar Graphs With Columns* in the *Creating Reports With Report Painter* manual.

You can also apply data visualization bar graphs to columns by adding a StyleSheet declaration to a procedure that includes the GRAPHTYPE attribute. For details, see the *Creating Reports With WebFOCUS Language* manual.
Saving and Reusing Report Output

When you run a report request, by default the data values that you request are collected and presented in a viewable form complete with column headings and other formatting features. You can take those same data values and, instead of formatting them for viewing, use them to create a new file, data source, or spreadsheet.

When you create a report request, you have many choices for displaying, reusing, and distributing your report output. You can:

- Store your report output for reuse by another FOCUS procedure or a variety of other applications, including most popular spreadsheet programs and Relational Database Management Systems.

- Publish your report in a format that facilitates distribution to a large audience. For example:
  - Storing your report output in a file for viewing with popular applications (Excel, Word, and so on).
  - Publishing your report output on the web or corporate Intranet.

In this chapter:

- Types of Output Files
- Creating Output Files
- Output File Formats
- Save Report Output as a Native Temporary Table In Report Painter
- Saving Reports as HTML Output
- Saving Reports as Print Display Output: PDF, PS
- Saving Reports Using Excel Formats

Types of Output Files

You decide which type of output file you need by considering what you plan to do with the output after you capture it. You can even capture the output multiple times in several different formats. Each output file is classified as either Save or Hold.
Tip: When saving or holding output files, it is recommended to have an Allocation in place for the file. This is not applicable for PCHOLD files.

SAVE Files

A SAVE file is the output of a report request stored in a format that can be used by a variety of applications. The default Save format is ALPHA, an ASCII text file containing your data values as printable characters. Depending on the format type you specify, you can use a Save file as input to other programming languages, applications, and WebFOCUS facilities. For example, you can create a Save file as a Microsoft Excel spreadsheet or as a webpage. For details, see the procedure How to Create a SAVE File on page 308.

A SAVE file does not necessarily retain the spacing between columns and the other formatting features that are supported when report output appears on your screen. The values are formatted as required by the type of Save file you create. For information about the types of SAVE files, see SAVE File Formats on page 311.

A special type of SAVE file is the SAVB file. A SAVB file is the same as a SAVE file except that numeric data is stored in internal format (BINARY), which means that it is not viewable on your screen without conversion. You can use the SAVB file as input to a variety of applications. For example, you can use it as the transaction file for a Microsoft Visual Basic program.

HOLD Files

In its purest form, a HOLD file is the output of a report request stored in a file that you can use as input to another WebFOCUS procedure. When a HOLD file is created for this purpose, a corresponding Master File is created along with the HOLD file. For example, you can create a HOLD file in FOCUS data source format. WebFOCUS automatically creates two files: a new FOCUS data source (.foc) containing the records that your request retrieves and a new Master File (.mas) that describes the fields in those records. You can then create new report requests that extract data from the HOLD file.

There are several reasons you might want to create such a HOLD file. You may want to extract the few fields you need from a large data source for faster and more efficient retrieval, or you may need to store virtual field values or summary values calculated in one request for further processing in another request.
PCHOLD Files

You can also use data in the server environment and automatically transfer files you create to your local computer using special types of files called PCHOLD files. Both the data source and, when created, the internal Master File (.mas) are automatically transferred to the personal computer running Developer Studio. PCHOLD formats are particularly valuable when you wish to run a report remotely and save the data output files on a local workstation or transfer the files from a web server to a browser.

In addition, you can produce a local Master File when executing HOLD commands remotely on a server. Your remote HOLD command automatically delivers a copy of the Master File to the PC so you can create new reports against the extracted data using Report Painter. This process eliminates the step of having to transfer the Master File from the server, perform a PCHOLD, or physically create the Master File locally.

You can create both HOLD and PCHOLD files in formats that can be used (like SAVE files) by a variety of other applications. Since these applications do not require Master Files, the report output is extracted in the requested format, but no Master File is created.

For more information, see HOLD File Formats on page 314 and How to Create a HOLD File on page 310. You may also wish to refer to topics in the Describing Data With Graphical Tools manual for information on how to create Master Files and how to generate Master File synonyms from existing schema with a wizard.

Creating Output Files

Creating an output file is just like creating any report request except that you must identify which type of output file you want to create and, optionally, a format type and a name for the output file.

Output Files: When to Create Them

You can create an output file by doing the following:

- Within a report request using Report Painter.
  
  You would want to create the output file directly in the report request if it is a standard report that is part of an application, or if you know you will need a particular type of output file every time a user runs the request.

- Outside of a request after running it in the viewer.
  
  You may want to create an output file after running the report request if you need to view the report output before determining whether to create the output file, or if you want to create more than one type of output file to use for different purposes.
You can even create one output file in a request and additional output files after running the request.

**Output Files: Where to Put Them**

When you create a HOLD file, the file is automatically saved in a default location that was specified during installation, or in the directory specified by the TEMPDIR parameter. Administrators and developers can use the SET TEMPDIR command to assign a location for WebFOCUS temporary files and HOLD files.

HOLD files should not be moved to a new location because, they may not be found when you try to use them in a WebFOCUS procedure.

SAVE files are not bound by the same constraint since they are not meant to be used as input to a WebFOCUS procedure. You can provide a complete file specification that places your SAVE file in a new location from the Save File dialog box. However, you should make sure that any file you create is accessible to the application that will use it. Refer to the appropriate documentation for any application you plan to use with a Save file. For more information, see *How to Create a SAVE File* on page 308.

PC HOLD files are only supported in Report Painter.

**Output Files: How to Create Them**

You can create output files within a report request or after running the request and viewing the report output. Your graphical tools support both options.

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

1. In Report Painter, choose *Options* from the Report menu and select the Save file option from the Destination Output result drop-down list.

   **Note:** Ensure that the selected report output format is a SAVE file format. For a list of available formats, see *SAVE File Formats* on page 311.

2. Type a name for your file in the Name input field or click the ellipsis button to open the Save File dialog box.

   In the Save File dialog box, type a name for your file in the File name input field and select a location for your file using the Save in list box.

3. Click *OK* to close the Report Options dialog box.
Reference: Save File Dialog Box

You use the Save File dialog box to create SAVE files.

Options in this dialog box include the following:

File name

Is where you enter a file name for your Save file.

Save in

Determines the project name where your file is saved.

Example: Creating an Excel Worksheet

The following Report Painter example saves the report output as an Excel worksheet by first selecting Excel 2007 as the selected output format from the Report Options dialog box, selecting Save file as the Destination Output result, and then saving the Excel worksheet as savexls.xls.

You can open the resulting output file in Microsoft Excel as shown in the following image.

Note: See Saving Reports Using Excel Formats on page 332 for examples of output saved in new formats.
How to Create a HOLD File

Procedure: How to Create a HOLD File

1. In Report Painter, choose Options from the Report menu and select the Temporary file option from the Destination Output result drop-down list.

   Note: All report output formats are available for the Temporary (HOLD) file format. See HOLD File Formats on page 314 for a list of HOLD file formats.

2. Enter the Temporary file name. The default file name is HOLD. If you use the default name, each time you create a HOLD file, the new file overwrites the previous one.

   or

   Save the report output to a file. Enter up to 66 characters as the Temporary (HOLD) file name or click the ellipsis button to open the Save File dialog box.

   Tip: This is recommended to avoid having your file overwritten the next time you create a HOLD file.

   When you run the report, the output is saved with an associated Master File.

3. Click OK to close the Report Options dialog box.

How to Create a PCHOLD File

Procedure: How to Create a PCHOLD File

1. In Report Painter, choose Output from the Report menu and select Web browser from the Destination Output result drop-down list. This is the default option.

   Note: Ensure that the selected report output format is a Web browser (PCHOLD) file format. For a list of available formats, see PCHOLD File Formats on page 321.

   The report output displays in a web browser and adds PCHOLD to the code.

Output File Formats

Each type of output file supports specific format types. For details, see SAVE File Formats on page 311 and HOLD File Formats on page 314.
SAVE File Formats

The following is a list of file formats available for SAVE files from Report Painter.

Administrators and developers can request all formats by issuing the appropriate WebFOCUS command from the Command Console or by creating a procedure in the Text Editor.

<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>Saves report data in character (ASCII) format. Alpha is the default value.</td>
<td>.ftm</td>
<td>Unstyled</td>
</tr>
<tr>
<td>COMMA</td>
<td>Saves all the columns of the report request and creates a CSV (Comma Separated Values) file. Alphanumeric fields are enclosed in quotation marks. Columns are separated by commas.</td>
<td>.prn</td>
<td>Unstyled</td>
</tr>
<tr>
<td>COMT</td>
<td>Comma-delimited text file with field names.</td>
<td>.csv</td>
<td>Unstyled</td>
</tr>
<tr>
<td>DIF</td>
<td>Saves the entire report, including the report data, column headings, footings, and subtotals, and creates a DIF (Data Interchange Format) file that can be easily imported into most spreadsheet packages.</td>
<td>.dif</td>
<td>Unstyled</td>
</tr>
<tr>
<td>DOC</td>
<td>Captures the entire report output, including headings, footings, and subtotals, and creates a text file that can be easily incorporated into most word processing packages. DOC format uses an ASCII form-feed character to indicate page control information.</td>
<td>.doc</td>
<td>Unstyled</td>
</tr>
<tr>
<td>EXCEL</td>
<td>Saves the report output as a Microsoft Excel 95 worksheet.</td>
<td>.xls</td>
<td>Unstyled</td>
</tr>
<tr>
<td>HTML</td>
<td>Creates an output file that contains an HTML document. For more information, see Saving Reports as HTML Output on page 329.</td>
<td>.htm</td>
<td>Styled</td>
</tr>
<tr>
<td>HTMLTABLE</td>
<td>HTML Table.</td>
<td>.html</td>
<td>Styled</td>
</tr>
</tbody>
</table>
### Output File Formats

<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHTML</td>
<td>An HTML active report is a report that is designed for offline analysis. For more information, see <em>Using an Active Technologies Report in the Active Technologies User's Guide.</em></td>
<td>.htm</td>
<td>Styled</td>
</tr>
<tr>
<td>FLEX</td>
<td>The active report for Adobe Flash Player includes most of the capabilities available in the HTML version of active reports delivered as a self-contained Adobe® Flash® Player compatible file (.swf). For more information, see <em>Active Technologies Report Integration With Adobe Flash Player in the Active Technologies User's Guide.</em></td>
<td>.swf</td>
<td>Styled</td>
</tr>
<tr>
<td>APDF</td>
<td>Using Active Technologies for Adobe Flash Player in PDF provides the ability to include interactive Flash® based reports, dashboards, and animations in PDF documents. For more information, see <em>Active Technologies Report Integration With Adobe Flash Player in the Active Technologies User's Guide.</em></td>
<td>.pdf</td>
<td>Styled</td>
</tr>
<tr>
<td>PDF</td>
<td>Saves the report output in Adobe Portable Document Format, which allows precise placement of output (all formatting options) on the printed page so the report looks exactly as it would when printed. For more information on PDF files, see <em>Saving Reports as Print Display Output: PDF, PS</em> on page 330.</td>
<td>.pdf</td>
<td>Styled</td>
</tr>
<tr>
<td>WP</td>
<td>Saves the entire report output, including titles, headings, and footings, and creates a text file (with IBM OEM codes) that can be easily incorporated into most word processing packages. Text with layout, without linebreaks.</td>
<td>.wp</td>
<td>Unstyled</td>
</tr>
<tr>
<td>File Format</td>
<td>Output</td>
<td>Extension</td>
<td>Report Format</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>&amp;WFFMT</td>
<td>User (Variable output).</td>
<td>.html, .xht, .pdf</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL2K</td>
<td>Generates fully styled reports in Excel 2000 HTML format. Requires Excel 2000 on your PC. For details, see Saving Reports Using Excel Formats on page 332.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL07</td>
<td>Generates fully styled reports in Excel 2007. Requires Excel 2007 on your PC. For details, see Saving Reports Using Excel Formats on page 332.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL2K FORMULA</td>
<td>Excel 2000 with formulas.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL2K PIVOT</td>
<td>Generates fully styled reports in Excel 2000 HTML format, with added pivoting capabilities. Requires Excel 2000 on your PC. For details, see Saving Reports Using Excel Formats on page 332.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL97</td>
<td>Excel 97.</td>
<td>.xls</td>
<td>Styled</td>
</tr>
<tr>
<td>PS</td>
<td>PostScript.</td>
<td>.ps</td>
<td>Styled</td>
</tr>
</tbody>
</table>
DHTML

Provides HTML output that has most of the features normally associated with output formatted for printing such as PDF or PostScript output. You can create an HTML file (.htm) or a Web Archive file (.mht). The type of output file produced is controlled by the value of the HTMLARCHIVE parameter. HTMLARCHIVE=ON creates a Web Archive file.

Some of the features supported by format DHTML are:

- Absolute positioning. DHTML precisely places text and images inside an HTML report, allowing you to use the same StyleSheet syntax to lay out HTML as you use for PDF or PS output.
- On demand paging. On demand paging is available with SET HTMLARCHIVE=OFF.
- PDF StyleSheet features. For example, the following features are supported: grids, background colors, OVER, bursting, coordinated compound reports.

LOTUS

Comma-delimited text file for Lotus 123.

VISDIS

Tab delimited file for Visual Discovery.

**HOLD File Formats**

The following is a list of file formats available for Hold files.
These formats are supported in the WebFOCUS language and in Report Painter (RP).

<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>Saves the report output as MS-DOS text data. It also creates a Master File.</td>
<td>.ftm</td>
<td>Unstyled</td>
</tr>
<tr>
<td>BINARY</td>
<td>Saves the report data and stores numeric fields as internal binary numbers and also creates a Master File. BINARY is the default format.</td>
<td>.ftm</td>
<td>Unstyled</td>
</tr>
<tr>
<td>COMMA</td>
<td>Saves all the columns of the WebFOCUS report request and creates a CSV (Comma Separated Values) file. Alphanumeric fields are enclosed in quotation marks. Columns are separated by commas.</td>
<td>.prn</td>
<td>Unstyled</td>
</tr>
<tr>
<td>COMT</td>
<td>The COMT format saves the column headings in the first row of the output file. It produces a variable-length text file, with fields separated by commas and with character values enclosed in double quotation marks. This is useful for further processing in a database application. This format type can be imported into applications, such as Excel or Lotus.</td>
<td>.csv</td>
<td>Unstyled</td>
</tr>
<tr>
<td>DBASE</td>
<td>Creates an extract file in dBase format that includes column headings in the first row.</td>
<td>.dbf</td>
<td>Database</td>
</tr>
<tr>
<td>DB2</td>
<td>Captures the report data and creates an IBM DB2 data source table.</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>DIF</td>
<td>Captures the entire report, including the report data, column headings, footings, and subtotals, and creates a DIF (Data Interchange Format) file that can be easily imported into most spreadsheet packages.</td>
<td>.dif</td>
<td>Unstyled</td>
</tr>
<tr>
<td>File Format</td>
<td>Output</td>
<td>Extension</td>
<td>Report Format</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>DOC</td>
<td>Saves the report output as MS-DOS text with layout and line breaks.</td>
<td>.doc</td>
<td>Unstyled</td>
</tr>
<tr>
<td>&amp;WFFMT</td>
<td>User Variable output.</td>
<td>.html, .xht, .pdf</td>
<td>Styled</td>
</tr>
<tr>
<td>EXCEL</td>
<td>Saves the report output as a Microsoft Excel 95 worksheet.</td>
<td>.xls</td>
<td>Unstyled</td>
</tr>
<tr>
<td>EXL2K</td>
<td>Generates fully styled reports in Excel 2000 HTML format. Requires Excel 2000 on your PC. For details, see Saving Reports Using Excel Formats on page 332.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL07</td>
<td>Generates fully styled reports in Excel 2007. Requires Excel 2007 on your PC. For details, see Saving Reports Using Excel Formats on page 332.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL2K FORMULA</td>
<td>Generates fully styled reports in Excel 2000 HTML format, with added pivoting capabilities. Requires Excel 2000 on your PC. For details, see Saving Reports Using Excel Formats on page 332.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL2K PIVOT</td>
<td>Generates fully styled reports in Excel 2000 HTML format, with added pivoting capabilities. Requires Excel 2000 on your PC. For details, see Saving Reports Using Excel Formats on page 332.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>EXL97</td>
<td>Excel 97.</td>
<td>.xls</td>
<td>Styled</td>
</tr>
<tr>
<td>File Format</td>
<td>Output</td>
<td>Extension</td>
<td>Report Format</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>Default</td>
<td>Captures the report data and creates a FOCUS data source.</td>
<td>.foc</td>
<td>Database</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> There is an additional Index Input field on the Output tab of the Report Options dialog box.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML</td>
<td>Creates an output file that contains an HTML (web) document.</td>
<td>.htm</td>
<td>Styled</td>
</tr>
<tr>
<td></td>
<td>For more information, see <em>Saving Reports as HTML Output</em> on page 329.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTMTABLE</td>
<td>Creates an output file that contains an HTML table, not a complete HTML document.</td>
<td>.htm</td>
<td>Styled</td>
</tr>
<tr>
<td></td>
<td>For more information, see <em>Saving Reports as HTML Output</em> on page 329.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AHTML</td>
<td>An HTML active report is a report that is designed for offline analysis. For more information, see <em>Using an Active Technologies Report</em> in the <em>Active Technologies User's Guide</em>.</td>
<td>.htm</td>
<td>Styled</td>
</tr>
<tr>
<td>FLEX</td>
<td>The active reports using Adobe® Flex® include most of the capabilities available in the HTML version of active reports delivered as a self-contained Adobe® Flash® Player compatible file (.swf). For more information, see <em>Active Technologies Report Integration With Adobe Flash Player</em> in the <em>Active Technologies User's Guide</em>.</td>
<td>.swf</td>
<td>Styled</td>
</tr>
</tbody>
</table>
### Output File Formats

<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>APDF</td>
<td>Using Active Technologies for Adobe Flash Player in PDF provides the ability to include interactive Flash® based reports, dashboards, and animations in PDF documents. For more information, see <em>Active Technologies Report Integration With Adobe Flash Player</em> in the <em>Active Technologies User’s Guide</em>.</td>
<td>.pdf</td>
<td>Styled</td>
</tr>
<tr>
<td>PDF</td>
<td>Saves the report output in Adobe Portable Document Format, which allows precise placement of output (all formatting options) on the printed page so the report looks exactly as it would when printed. For more information on PDF files, see <em>Saving Reports as Print Display Output: PDF, PS</em> on page 330.</td>
<td>.pdf</td>
<td>Styled</td>
</tr>
<tr>
<td>PostScript (PS)</td>
<td>Saves the report as a PostScript document. You must have installed a third party tool capable of displaying PS. See <em>Saving Reports as Print Display Output: PDF, PS</em> on page 330.</td>
<td>.ps</td>
<td>Styled</td>
</tr>
</tbody>
</table>
DHTML | Provides HTML output that has most of the features normally associated with output formatted for printing, such as PDF or PostScript output. You can create an HTML file (.htm) or a Web Archive file (.mht). The type of output file produced is controlled by the value of the HTMLARCHIVE parameter. HTMLARCHIVE=ON creates a Web Archive file. Some of the features supported by format DHTML are:
- Absolute positioning. DHTML precisely places text and images inside an HTML report, allowing you to use the same StyleSheet syntax to lay out HTML as you use for PDF or PS output.
- On demand paging. On demand paging is available with SET HTMLARCHIVE=OFF.
- PDF StyleSheet features. For example, the following features are supported: grids, background colors, OVER, bursting, coordinated compound reports.
| .fmp | Styled |

DB2 | DB2 database table. | n/a | Database |
<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMEDB</td>
<td>Creates a report output file as a native DBMS temporary table. This increases performance by keeping the output file on the DBMS server, instead of downloading it to your computer. When SAMEDB is selected, a Persistence option is available on the Report Options Output tab. Select from VOLATILE, GLOBAL_TEMPORARY, and PERMANENT. For more information, see <em>Save Report Output as a Native Temporary Table In Report Painter</em> on page 326.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>SQLDBC</td>
<td>Creates an extract file stored in the Teradata Server.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>SQLINF</td>
<td>Captures the report data and creates an Informix data source table.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>SQLMAC</td>
<td>Saves report output as a Microsoft Access database table.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>SQLMSS</td>
<td>Captures the report data and creates a Microsoft SQL Server data source table.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>SQLSYB</td>
<td>Captures the report data and creates a Sybase data source table.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>SQLODBC</td>
<td>Captures the report data and creates a file or data source table using the current ODBC data source driver.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>SQLORA</td>
<td>Captures the report data and creates an Oracle data source table.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>SQLNEO</td>
<td>Captures the report data and creates an HP Neoview data source table.</td>
<td>n/a</td>
<td>Database</td>
</tr>
</tbody>
</table>
## PCHOLD File Formats

The following is a list of file formats available for PCHOLD files from Report Painter. Several formats support PCHOLD variations for transferring remotely-created output from a host to a local workstation or from a web server to a browser.

<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>INGRES</td>
<td>Ingres database table.</td>
<td>n/a</td>
<td>Database</td>
</tr>
<tr>
<td>TAB</td>
<td>Tab Delimited file.</td>
<td>.tab</td>
<td>Unstyled</td>
</tr>
<tr>
<td>TABT</td>
<td>Tab Delimited text file with field names.</td>
<td>.tab</td>
<td>Unstyled</td>
</tr>
<tr>
<td>VISDIS</td>
<td>Creates a tab-delimited output file in which the first record has the column titles and the second record contains Visual Discovery formats based on the FOCUS field formats of the data.</td>
<td>.txt</td>
<td>Unstyled</td>
</tr>
<tr>
<td>LOTUS</td>
<td>Comma-delimited text file for Lotus 123.</td>
<td>.csv</td>
<td>Unstyled</td>
</tr>
<tr>
<td>SYLK</td>
<td>Microsoft Multiplan spreadsheet.</td>
<td></td>
<td>Unstyled</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>Binary file without padding.</td>
<td>.ftm</td>
<td>Unstyled</td>
</tr>
<tr>
<td>XML</td>
<td>Standard XML.</td>
<td>.xml</td>
<td>Unstyled</td>
</tr>
<tr>
<td>WP</td>
<td>Saves the entire report output, including titles, headings, and footings, and creates an MS-DOS text file (with IBM OEM codes) with no line breaks. This file can be easily incorporated into most word processing packages.</td>
<td>.wp</td>
<td>Unstyled</td>
</tr>
<tr>
<td>WK1</td>
<td>Captures the report output data and creates a file that can be imported into Lotus 1-2-3, Release 2.</td>
<td>.wk1</td>
<td>Unstyled</td>
</tr>
</tbody>
</table>
The PCHOLD variations are only supported in Report Painter.

<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCHOLD ALPHA</td>
<td>Saves report data in character (ASCII) format. Alpha is the default value.</td>
<td>.ftm</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PCHOLD BINARY</td>
<td>Binary Data file with File Description.</td>
<td>.ftm</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PCHOLD COMMA</td>
<td>Saves all the columns of the report request and creates a CSV (Comma Separated Values) file. Alphanumeric fields are enclosed in quotation marks. Columns are separated by commas.</td>
<td>.prn</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PCHOLD DIF</td>
<td>Saves the entire report, including the report data, column headings, footings, and subtotals, and creates a DIF (Data Interchange Format) file that can be easily imported into most spreadsheet packages.</td>
<td>.dif</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PCHOLD DOC</td>
<td>Captures the entire report output, including headings, footings, and subtotals, and creates a text file that can be easily incorporated into most word processing packages. DOC format uses an ASCII form-feed character to indicate page control information.</td>
<td>.doc</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PCHOLD EXCEL</td>
<td>Saves the report output as a Microsoft Excel 95 worksheet.</td>
<td>.xls</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PCHOLD HTML</td>
<td>Creates an output file that contains an HTML document. For more information, see Saving Reports as HTML Output on page 329.</td>
<td>.htm</td>
<td>Styled</td>
</tr>
<tr>
<td>PCHOLD HTMLTABLE</td>
<td>Transfers the data from a web server to a browser.</td>
<td>.html</td>
<td>Styled</td>
</tr>
</tbody>
</table>
### File Format | Output | Extension | Report Format
--- | --- | --- | ---
PCHOLD AHTML | The HTML active report is a report that is designed for offline analysis. For more information, see *Using an Active Technologies Report* in the *Active Technologies User's Guide*. | .htm | Styled
PCHOLD FLEX | The active reports using Adobe® Flex® include most of the capabilities available in the HTML version of active reports delivered as a self-contained Adobe® Flash® Player compatible file (.swf). For more information, see *Active Technologies Report Integration With Adobe Flash Player* in the *Active Technologies User's Guide*. | .swf | Styled
PC HOLD APDF | Using Active Technologies for Adobe Flash Player in PDF provides the ability to include interactive Flash® based reports, dashboards, and animations in PDF documents. For more information, see *Active Technologies Report Integration With Adobe Flash Player* in the *Active Technologies User's Guide*. | .pdf | Styled
PCHOLD PDF | Saves the report output in Adobe Portable Document Format, which allows precise placement of output (all formatting options) on the printed page so the report looks exactly as it would when printed. For more information on PDF files, see *Saving Reports as Print Display Output: PDF, PS* on page 330. | .pdf | Styled
PCHOLD TAB | Tab Delimited file. | .tab | Unstyled
PCHOLD TABT | Tab Delimited text file with field names. | .tab | Unstyled
PCHOLD WK1 | Saves report output as a Lotus 1-2-3 worksheet, Release 2. | .wk1 | Unstyled
## Output File Formats

<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC HOLD WP</td>
<td>Saves the entire report output, including titles, headings, and footings, and creates a text file (with IBM OEM codes) that can be easily incorporated into most word processing packages.</td>
<td>.wp</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PC HOLD &amp;WFFMT</td>
<td>User Variable output.</td>
<td>.html, .xht, .pdf</td>
<td>Styled</td>
</tr>
<tr>
<td>PC HOLD EXL2K</td>
<td>Excel 2000.</td>
<td>.xls</td>
<td>Styled</td>
</tr>
<tr>
<td>PC HOLD EXL07</td>
<td>Excel 2007.</td>
<td>.xls</td>
<td>Styled</td>
</tr>
<tr>
<td>PCHOLD EXL2K FORMULA</td>
<td>Excel with formulas.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>PCHOLD EXL2K PIVOT</td>
<td>Excel 2000 PivotTable.</td>
<td>.xht</td>
<td>Styled</td>
</tr>
<tr>
<td>PCHOLD EXL97</td>
<td>Excel 97.</td>
<td>.xls</td>
<td>Styled</td>
</tr>
<tr>
<td>PCHOLD PS</td>
<td>PostScript.</td>
<td>.ps</td>
<td>Styled</td>
</tr>
</tbody>
</table>
**Report Format**

<table>
<thead>
<tr>
<th>File Format</th>
<th>Output</th>
<th>Extension</th>
<th>Report Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCHOLD DHTML</td>
<td>Provides HTML output that has most of the features normally associated with output formatted for printing such as PDF or PostScript output. You can create an HTML file (.htm) or a Web Archive file (.mht). The type of output file produced is controlled by the value of the HTMLARCHIVE parameter. HTMLARCHIVE=ON creates a Web Archive file. Some of the features supported by format DHTML are:</td>
<td>.fmp</td>
<td>Styled</td>
</tr>
<tr>
<td></td>
<td>- <strong>Absolute positioning.</strong> DHTML precisely places text and images inside an HTML report, allowing you to use the same StyleSheet syntax to lay out HTML as you use for PDF or PS output.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <strong>On demand paging.</strong> On demand paging is available with SET HTMLARCHIVE=OFF.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <strong>PDF StyleSheet features.</strong> For example, the following features are supported: grids, background colors, OVER, bursting, coordinated compound reports.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC HOLD LOTUS</td>
<td>Comma-delimited text file for Lotus 123.</td>
<td>.csv</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PCHOLD XML</td>
<td>Standard XML.</td>
<td>.xml</td>
<td>Unstyled</td>
</tr>
<tr>
<td>PCHOLD VISDIS</td>
<td>Tab delimited file for Visual Discovery.</td>
<td>.txt</td>
<td>Unstyled</td>
</tr>
</tbody>
</table>
Save Report Output as a Native Temporary Table In Report Painter

You can now create a report output file (that is, a HOLD file) as a native DBMS temporary table. This increases performance by keeping the entire reporting operation on the DBMS server, instead of downloading data to your computer and then back to the DBMS server.

For example, if you temporarily store report output for immediate use by another procedure, storing it as a temporary table instead of creating a standard HOLD file avoids the overhead of transmitting the interim data to your computer.

The temporary table columns are created from the following report elements: Display columns, Sort (BY) columns, and COMPUTE columns except for those for which NOPRINT is specified.

The temporary table that you create from your report will be the same data source type (that is, the same DBMS) as the data source from which you reported. If the data source from which you reported contains multiple tables, all must be of the same data source type and reside on the same instance of the DBMS server.

You can choose between several types of table persistence.

You can create extract files as native DBMS tables with the following adapters: DB2 (on z/OS, UNIX, and Windows), Informix, Microsoft SQL Server, MySQL, Oracle, and Teradata.

Procedure: How to Save Report Output as a Native Temporary Table In Report Painter

1. In Report Painter, select Options from the Report menu.

2. From the Output Format drop-down list, expand the Database Formats folder and select Native Database Table (SAME DB).

   The top level of the Report Options dialog box shows the selected format and determines what tabs and options appear in the lower part of the dialog box.

3. Specify the name of the output file. It defaults to HOLD. To select an existing file, click the Browse button.

   Because each subsequent HOLD command overwrites the previous HOLD file, it is advisable to specify a name in each request to direct the extracted data to a separate file, thereby preventing an earlier file from being overwritten by a later one.

4. Select the type of table persistence:

   - VOLATILE. Specifies that the table is local to the DBMS session. A temporary synonym (a Master File and Access File) is generated automatically. It expires when the server session ends.
For information about support for the volatile setting, and about persistence and other table properties, for a particular DBMS, see *Temporary Table Properties for SAME_DB Persistence Values* on page 327, and consult your DBMS vendor documentation.

- **GLOBAL_TEMPORARY.** Specifies that the table persistence depends on your DBMS. While the table exists, its schema will be visible to other database sessions and users though its data will not be. A permanent synonym (a Master File and Access File) is generated automatically.

For information about support for the global temporary setting, and about persistence and other table properties, for a particular DBMS, see *Temporary Table Properties for SAME_DB Persistence Values* on page 327, and consult your DBMS vendor documentation.

- **PERMANENT.** Specifies that the table will be permanent. A permanent synonym (a Master File and Access File) is generated automatically.

**Reference:** *Temporary Table Properties for SAME_DB Persistence Values*

The following chart provides additional detail about persistence and other properties of temporary tables of different data source types that are supported for use with HOLD format SAME_DB.

<table>
<thead>
<tr>
<th>DBMS</th>
<th>VOLATILE</th>
<th>GLOBAL_TEMPORARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 on UNIX, Windows, and DB2 for z/OS: a volatile table is created using the DECLARE GLOBAL TEMPORARY TABLE command with the ON COMMIT PRESERVE ROWS option. Declared global temporary tables persist and are visible only within the current session (connection). SESSION is the schema name for all declared global temporary tables.</td>
<td>DB2 Release 7.1 and up for z/OS only: a global temporary table is created using the CREATE GLOBAL TEMPORARY TABLE command. The definition of a created global temporary table is visible to other sessions, but the data is not. The data is deleted at the end of each transaction (COMMIT or ROLLBACK command). The table definition persists after the session ends.</td>
<td></td>
</tr>
<tr>
<td>DBMS</td>
<td>VOLATILE</td>
<td>GLOBAL_TEMPORARY</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Informix</td>
<td>A volatile table is created using the CREATE TEMP TABLE command with the</td>
<td>This type of table is not supported by Informix®.</td>
</tr>
<tr>
<td></td>
<td>WITH NO LOG option. The definition and the data persist, and are</td>
<td></td>
</tr>
<tr>
<td></td>
<td>visible, only within the current session.</td>
<td></td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>A volatile table is created as a local temporary table whose name is</td>
<td>The name of a global temporary table is prefixed with two number signs (##).</td>
</tr>
<tr>
<td></td>
<td>prefixed with a single number sign (#). Therefore, the name of a</td>
<td>Therefore, the name of a global temporary table used as a HOLD file is the name</td>
</tr>
<tr>
<td></td>
<td>volatile table used as a HOLD file is the name specified by the HOLD</td>
<td>specified by the HOLD phrase, prefixed with two number signs (##). The table is</td>
</tr>
<tr>
<td></td>
<td>phrase, prefixed with a number sign (#). The table definition and the</td>
<td>dropped automatically when the session that created the table ends and all other</td>
</tr>
<tr>
<td></td>
<td>data persist, and are visible, only within the current session.</td>
<td>tasks have stopped referencing it. The table definition and data are visible to</td>
</tr>
<tr>
<td>MySQL</td>
<td>A volatile table is created using the CREATE TEMPORARY TABLE command. A</td>
<td>This type of table is not supported by MySQL.</td>
</tr>
<tr>
<td></td>
<td>temporary table persists and is visible only within the current session</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(connection). If a temporary table has the same name as a permanent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>table, the permanent table becomes invisible.</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>This type of table is not supported by Oracle.</td>
<td>The table definition is visible to all sessions and its data is visible only to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the session that inserts data into it. The table definition persists for the same</td>
</tr>
<tr>
<td></td>
<td></td>
<td>period as the definition of a regular table.</td>
</tr>
</tbody>
</table>
A global temporary table persists for the same duration as a permanent table. The definition is visible to all sessions, but the data is visible only to the session that inserted the data. The global temporary table is created with the ON COMMIT PRESERVE ROWS option.

A volatile table definition and data are visible only within the session that created the table and inserted the data. The volatile table is created with the ON COMMIT PRESERVE ROWS option.

<table>
<thead>
<tr>
<th>DBMS</th>
<th>VOLATILE</th>
<th>GLOBAL_TEMPORARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teradata</td>
<td>A volatile table definition and data are visible only within the session</td>
<td>A global temporary table persists for the same duration as a permanent table.</td>
</tr>
<tr>
<td></td>
<td>that created the table and inserted the data. The volatile table is created</td>
<td>The definition is visible to all sessions, but the data is visible only to the</td>
</tr>
<tr>
<td></td>
<td>with the ON COMMIT PRESERVE ROWS option.</td>
<td>session that inserted the data. The global temporary table is created with the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ON COMMIT PRESERVE ROWS option.</td>
</tr>
</tbody>
</table>

**Saving Reports as HTML Output**

WebFOCUS enables you to save your report output with all of the HTML formatting necessary to view the document using a browser. See *Viewing and Printing Reports and Graphs* on page 295 for an illustration.

WebFOCUS provides the following format options for reports that create HTML documents:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Related Viewing Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML document (.htm)</td>
<td>Saves report output as an HTML document, which can be opened in any web</td>
<td>Web browsers</td>
</tr>
<tr>
<td></td>
<td>browser that supports tables. This format is useful if users need to see</td>
<td></td>
</tr>
<tr>
<td></td>
<td>report output that is ready to display.</td>
<td></td>
</tr>
<tr>
<td>HTMTABLE file (.htm)</td>
<td>Saves report output as an HTML table that can be incorporated into an</td>
<td>Web browsers</td>
</tr>
<tr>
<td></td>
<td>existing webpage. This format is useful when you want to run a report</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dynamically in WebFOCUS using up-to-the-minute data, and view the report</td>
<td></td>
</tr>
<tr>
<td></td>
<td>output in a customized webpage.</td>
<td></td>
</tr>
<tr>
<td>HTML Active Technologies Reports</td>
<td>An active report is a report that is designed for offline analysis. For</td>
<td>Web browsers</td>
</tr>
<tr>
<td></td>
<td>more information, see <em>Using an Active Technologies Report</em> in the Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Technologies User's Guide.</td>
<td></td>
</tr>
</tbody>
</table>
**Procedure: How to View Reports as HTML**

If you choose HTML format from the Output Format toolbar in Report Painter, you can run your report and it will be saved as an HTML document.

1. In Report Painter, click the **Options** button. The Report Options dialog box opens.
2. Select an HTML Styled report format from the Output Format drop-down list.

The report appears in HTML format in your browser. For an illustration, see **Viewing and Printing Reports and Graphs** on page 295.

**Saving Reports as Print Display Output: PDF, PS**

The PDF HOLD and SAVE formats enable you to produce fully formatted PDF output (for example, headings, footings, and titles), correctly aligned on the physical pages. PDF format also supports StyleSheets that contain drill-down parameters and links to arbitrary URLs and embedded GIF images in report, page, and sort headings and footings.

For details about PDF format, see the **Creating Reports With WebFOCUS Language** manual.

When you create PDF output:

- If you issue a FILEDEF command for the file name, the PDF output is created in the file you specify in the Allocation. A FILEDEF command can be created by using the Allocation Wizard.

PS (PostScript format), a print-oriented page description language, is most often used to send a report directly to a printer. While used less frequently as an online display format, you can display PS report output on your monitor before printing it. For details, see **Choosing a Display Format** in the **Creating Reports With WebFOCUS Language** manual.

With the exception of drill-downs, all of the report formatting features that are supported for PDF are also supported for PostScript output.

For more information about PS format, see **Choosing a Display Format** in the **Creating Reports With WebFOCUS Language** manual.

**Reference: Required Software Configuration for PDF Report Display**

You need Adobe Acrobat Reader Version 5.0 or higher in order to display the PDF output in your browser. The appearance of the report in your browser looks precisely as it would when printed. The Adobe Acrobat Reader is an Internet standard that is available for Windows, UNIX, and Macintosh.
**Note:** You can view PDF reports with the full version of Adobe® Acrobat® product (that you can purchase), or with Acrobat Reader, which is free.

When creating a PDF HOLD file, the output is saved on the PC and can be viewed without using your browser. Acrobat Reader can be used directly.

If you did not install Adobe Acrobat Reader when you installed Developer Studio, it is available for download from http://www.adobe.com. This download is free of charge.

Acrobat Reader is used as a plug-in to your browser. You can view PDF-formatted reports with no additional setup or configuration.

Acrobat Reader is required only to display the PDF output. No third-party products are required to produce the PDF output.

**Reference: StyleSheet Features in PDF**

All StyleSheet features that are supported for PostScript behave identically for PDF, unless otherwise noted in this section. The output produced by PDF is identical to that produced by PostScript (subject to font availability).

PostScript fonts are used and can be scaled to any point size. The fonts that are supported are Courier (fixed width), Times (proportional width), and Helvetica (proportional width). The font styles that can be used are Normal (default), Bold, Italic, Underline, and combinations of these. Note that the font style Outline, which is available for PostScript, is not available for PDF. Also, the support for ISO Latin font encoding that is present in PostScript is not available for PDF.

The following StyleSheet features are not supported with PDF (HTML only):

- JAVASCRIPT (drill down to JavaScript)
- TARGET
- COLSPAN
- IMAGEALIGN
- IMAGEBREAK
- PAGEMATRIX, GUTTER

**Reference: Support for Drill-Downs in PDF Files**

PDF supports the creation of hyperlinks (drill downs). You can create a hyperlink to drill down to or from a PDF report, to drill down to a graph, or link to a webpage.
When you drill down from one report to another report, do not use the following characters:

- + (plus sign)
- # (number sign)
- & (ampersand)
- \ (backslash)

**Note:** PDF does not support drill downs when sorting across column.

**Reference:** Differences Between PDF and HTML Formats

The following are the differences between PDF and HTML formats:

**PDF**
- Has no PDF equivalent to HTML-formatted forms or HTML tables.
- Does not support dynamic features that require JavaScript code, such as OLAP reports.

**HTML**
- Supports JavaScript hyperlinks.

**Saving Reports Using Excel Formats**

You can display a report as one of several kinds of Microsoft Excel spreadsheets. The report opens in Excel within a web browser.

You can display a report as an:

- **Excel 2000 worksheet (format EXL2K).** The Excel 2000 format supports most StyleSheet attributes, allowing for full report formatting. The computer on which the report appears must have Microsoft Excel 2000 or higher installed.

  In addition, WebFOCUS supports two Excel 2000 variations: EXL2K FORMULA and EXL2K PIVOT. When either of these formats is specified additional processing is done.

- **Excel 2000 FORMULA (EXL2K FORMULA).** If you display a report using the format variation EXL2K FORMULA, the resulting spreadsheet will contain Excel formulas that calculate and display the results of any type of summed information (such as column totals, and row totals.)
Excel 2000 PivotTable (format EXL2K PIVOT). PivotTable is an Excel tool for analyzing complex data, much like WebFOCUS OLAP. Note that a report displayed as a PivotTable spreadsheet may not retain its formatting if you insert an additional field from the pivot cache, move a column position, or remove a field from the PivotTable.


Excel 97 spreadsheet (format EXL97). Excel 97 is an HTML-based display format that opens in Excel 97 or higher and supports report formatting and drill-downs. The computer on which the report appears must have Microsoft Excel 97 installed.

Note: Since only single line (single cell) column titles are supported in format EXCEL reports, any additional column title rows are treated as data. For example, if you have a report with a multi-line (multi-cell) column title and you sort the column, the second (and so on) column title rows will be sorted with the data. To avoid this, only select the data instead of the entire column when you select sorting options in Excel.

For details about displaying reports in Excel formats, see Choosing a Display Format in the Creating Reports With WebFOCUS Language manual.

Procedure: How to Create an EXL2K Report

The following example illustrates how to create a report in EXL2K format:

1. After selecting the CENTCOMP data source to report against, open Report Painter.
2. Type STORE NAME BY STATE in the heading area.
3. Move the insertion point to the report and add the following fields from the Object Inspector: STATE, REGION, STORE_CODE, and STORENAME.
4. Select the Store Name column and click the Sum button.
5. Select the State, Region, and Store ID columns and click they By button.
7. Set your report output format to Excel 2000 by selecting Excel 2000 (EXL2K) from the Output Format drop-down list.
8. In the Customize worksheet title field, type State Report.
9. Click OK, then click Run.
Notice that the tab name has been changed from the Excel default, Sheet1, to a more descriptive name, which you specified in the Customize worksheet title input field on the Format tab of the Report Options dialog box.

**Reference:** **Font Support for EXL2K Format**

The EXL2K format supports fonts that are available on both the server and the client. Be sure to verify the existence of a particular font on both the client and the server before saving a report in the EXL2K format.

**Reference:** **Unsupported Date and Numeric Formats**

The following formats are not supported in EXL2K. They will translate into Excel General format and possibly produce unpredictable results:

- YY, Y, M, D, JUL, and I2MT.
- Any date format with a Q (quarter).
- Any packed-decimal (P) date formats.
- Any alphanumeric (A) date formats.
- Fixed Dollar (N) formats.
Multiple format options. Only single format options are supported when using FORMAT EXL2K. For example, the formats I9C and I9B are supported, but I9BC is not.

**Reference:** Using Date Separators in Excel

In order to use a dash (−) as a separator between month, day, and year in Excel, you must change the default date separator for Windows®. This setting can be located under Regional Options in the Control Panel.

**Using PivotTables in Excel 2000**

The power of EXL2K format derives in large measure from its ability to take advantage of PivotTables. The PivotTable is a tool used in Microsoft Excel to analyze complex data much like the OLAP tool in WebFOCUS. It allows you to drag and drop data fields within a PivotTable, providing different views of the data, such as sorting across rows or columns.

Report requests can be created within WebFOCUS and sent as output to a fully formatted Excel PivotTable. The ON TABLE PCHOLD FORMAT EXL2K PIVOT command will generate an Excel PivotTable in your browser, as shown in the following image.

![Excel PivotTable Example](image)

When FORMAT EXL2K PIVOT is enabled, two data streams are created:

- The first data stream is the PivotTable file.
The PivotTable file (.xht) is an HTML file with embedded XML. The HTML file contains all the information that appears in your browser.

- The second data stream is the PivotTable cache file (.xml).

The PivotTable cache file is a metadata type of file. It contains all the fields specified in the procedure and links internally to the PivotTable file. The PivotTable cache file can contain data fields called CACHEFIELDS, which populate the PivotTable toolbar, but do not initially appear in the report. CACHEFIELDS can be dragged and dropped from the PivotTable toolbar into the PivotTable when required for analysis.

For more information about PivotTables, see the Creating Reports With WebFOCUS Language manual.

**Procedure: How to Populate a PivotTable**

1. After selecting the Car data source to report against, open Report Painter.
2. In the heading area, type *CAR File PivotTable*.
3. Press the Enter key to add another line to the heading, and type *Sum of Sales by Car*.
4. Move the insertion point to the report and add the following fields from the Object Inspector: SALES and CAR.
5. Select the CAR column and click the By button.

Ensure that the SALES column is a Detail field.

**Note:** The PRINT command must be used. The PivotTable is generated by the PRINT command in combination with the BY, ACROSS, PAGEFIELDS and CACHEFIELDS phrases. See Choosing a Display Format in the Creating Reports With WebFOCUS Language manual.

6. Select *Excel 2000 PivotTable* from the Output Format toolbar drop-down list.
The Object Inspector opens at the Pivot tab and shows the Available Fields list, as shown in the following image.

7. Click COUNTRY in the Available Fields box and drag it to the Page Fields dimension in the Pivot hierarchy.

8. Click MODEL in the Available Fields box and drag it to the Cache Fields dimension in the Pivot hierarchy.

9. Run your report.
The report displays the PivotTable in Excel 2000. All fields (including PRINT, BY, and ACROSS fields, and PAGEFIELDS and CACHEFIELDS) appear in the toolbar, as shown in the following image.

Note: Text fields and Timestamp fields may not be used in Excel PivotTables. In addition, ensure that there is no missing data for any fields.

Viewing a Report in Excel 97 Format

The EXL97 format allows you to view reports in Excel 97, which supports full styling and drill-down capability.

When you specify EXL97 format, an HTML-based file is generated with an extension of .e97. The appropriate MIME type is automatically assigned to designate Excel as the active application for this file type.

When using the Microsoft Internet Explorer browser and Excel 97, the Excel client opens in the background and the report launches in your browser. You may see the Excel application open and minimize while viewing your report. Leave Excel open when viewing the spreadsheet.
After the report is generated, you can save the document as an Excel spreadsheet locally on your PC. In Excel 97, you will be prompted to save the document as a Microsoft Excel Workbook with an .xls extension. This will save the file as a binary Excel document.

For more information about Excel 97 format, see Choosing a Display Format in the Creating Reports With WebFOCUS Language manual.


**Note:** Although format EXL97 is available, there are some limitations when compared to the Excel 2000 (EXL2K) format. See Limitations for FORMAT EXL97 on page 340. Future enhancements in the area of Excel integration will primarily be made to the EXL2K format. We recommend upgrading to Excel 2000 or higher so you can take full advantage of our Excel integration, as well as all future enhancements.

**Procedure:** How to View a Report in EXL97 Format

1. After selecting the Centord data source to report against, open Report Painter.
2. In the heading area, type *Plant Revenue in Excel 97 Format*.
3. Move the insertion point to the report and add the following fields from the Object Inspector: *PLANTLNG* and *LINEPRICE*.
4. Select the *Line Total* column in the Report Painter window and click the *Sum* button.
5. Select the *PLANTLNG* column and click the *By* button.
6. Select *Excel 97* from the Output Format toolbar drop-down list.
7. Run your report. The output is shown in the following image.

<table>
<thead>
<tr>
<th>A1</th>
<th>Plant Revenue in Excel 97 Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><strong>PLANTLNG</strong></td>
</tr>
<tr>
<td>3</td>
<td>Boston</td>
</tr>
<tr>
<td></td>
<td>$262,433,980.63</td>
</tr>
<tr>
<td>4</td>
<td>Dallas</td>
</tr>
<tr>
<td></td>
<td>$97,751,846.65</td>
</tr>
<tr>
<td>5</td>
<td>Los Angeles</td>
</tr>
<tr>
<td></td>
<td>$57,507,080.82</td>
</tr>
<tr>
<td>6</td>
<td>Orlando</td>
</tr>
<tr>
<td></td>
<td>$97,616,855.94</td>
</tr>
<tr>
<td>7</td>
<td>Seattle</td>
</tr>
<tr>
<td></td>
<td>$22,742,743.88</td>
</tr>
<tr>
<td>8</td>
<td>St Louis</td>
</tr>
<tr>
<td></td>
<td>$195,970,536.09</td>
</tr>
</tbody>
</table>
Reference: Limitations for FORMAT EXL97

- This format is only compatible with Excel 97 or higher. This format is not compatible with Excel 95 or any versions of Excel prior to Excel 95.

- Excel 97 does not support Cascading Style Sheets (CSS). Any styling specified in a report that uses CSS (SET HTMLCSS=ON) will not be respected in Excel 97. The WRAP=n feature is not supported with EXL97 since this feature requires CSS.

- WebFOCUS uses nested HTML tables when generating a heading, which may cause a problem in EXL97. As an alternative, specify HEADALIGN=BODY in your stylesheets for all HEADING objects, include FOOTING, SUBHEAD and SUBFOOT commands. This creates the heading as a separate table.

  Page numbers appear by default and are placed in the heading. We recommend that you turn page numbering off (SET PAGE-NUM=OFF).

- All numeric and date formatting options that are not supported for EXL2K are also not supported for EXL97. In addition, negative numbers displayed with brackets, trailing zeroes after the decimal, and leading zeros will not appear with EXL97 reports. Note that dates are typically translated into the Excel General format, which can cause problems with sorting and other Excel features.

- PivotTables are not supported with EXL97. (EXL2K PIVOT is the only valid PivotTable format.)

- If you are using a Netscape® browser, you may have to assign the .e97 extension to the Excel MIME type manually the first time you run a report.
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.

In this chapter:

- Starting Update Assist
- Update Assist (Step 1 of 4): Selecting a Segment and Fields to Update
- Update Assist (Step 2 of 4): Selecting Navigation and Style Options
- Update Assist (Step 3 of 4): Selecting Output File Options
- Update Assist (Step 4 of 4): Confirming Selections
- About Your Update Assist Application
- Editing Your Update Assist Application
- Calling an Update Assist Procedure From a WebFOCUS Report
- Usage Notes

Starting Update Assist

You can create applications with Update Assist that add records, update records, or delete records against any data source for which you have read/write access. Data navigation and input validation are automatic. This means you get an update application with no need to design forms, or to write navigation, validation, or update procedures.

To start using Update Assist, do the following:

1. Open a project.
2. Right-click any Master File in the project path and click Update Assist.
3. Follow the directions in the Update Assist dialog boxes.

The four Update Assist dialog boxes correspond to the four steps for generating Update Assist applications.
Update Assist (Step 1 of 4): Selecting a Segment and Fields to Update

In the first window of Update Assist, you select the segment and fields you want to update.

Note: Update Assist is only available with a single segment Master File. The following image shows the Segment and Fields window.
This window contains the following fields and options:

**Update Segment Options**

Contains the segment you selected. In order to enable changes to any of the fields in a segment, select the Add, Update, or Delete options.

**Field View Options**

Contains the fields in the segment that you selected in the Update Segment Options section. Once you have enabled changes to the segment as a whole, you can set change options for each individual field in the segment.

**Display Name**

Is how the field name is displayed on the form.

**Name**

Is the name of the database 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.

**Tip:** You can select multiple fields and then click once to change the Visible or Changeable settings.

**Validation**

Applies a validation technique which verifies the value a user enters in the field. This option is available only if Changeable is set to Yes.

The options for Validation are:

**Automatic.** The default validation option, validates the user 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 does not require the server to validate the data.
Range. Allows you to define a numeric range between which data is valid for the field. See How to Use a Range to Validate a Field on page 345. 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. To retrieve a list of all values currently in the database, click the Get Values button. You can edit this supplied list. For details, see How to Use a Static List to Validate a Field on page 346.

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. For details, see How to Use a Dynamic List to Validate a Field on page 347.

None. Does not perform a validation.

Required field. Specifies that the user must supply a value for the field.

Procedure: How to Rename a Segment or a Field

You can easily rename a segment or field as it is displayed to the user (this is called the Display Name).

1. Right-click the segment or field.
2. Click Rename.
3. Type the new name and press Enter.

Procedure: How to Resort Fields in the Segment & Field Options Window

You can change the order of the fields as they appear in the window.

1. In the upper-right corner of the Field View Options pane, click the arrow to the right of the alphabetical sort button.
2. To sort fields by:
   - Display name, select Display name from the drop-down menu.
   - Original name, select Name from the drop-down menu.
   - The order in which they appear in the Master File, select Original order from the drop-down menu.
**Note:** This does not affect the order of the fields in your application. In the application, the fields are sorted according to their order in the Master File.

**Procedure:** How to Use a Range to Validate a Field

When you choose the *Range* option to validate a field, Update Assist opens the Field Validation - Range dialog box. You use this option with numeric fields to specify a range of values for any information the user enters. The Field Validation - Range dialog box is shown in the following image.

![Field Validation - Range dialog box](image)

1. Enter a *From* value to indicate the beginning of the acceptable range of values.
2. Enter a *To* value to indicate the end of the acceptable range of values.
3. Click *OK*.
Procedure: How to Use a Static List to Validate a Field

When you choose the Static List option to validate a field, the Field Validation - List dialog box opens. Use this option to specify a static list of values that the user can select from a pull-down list. The Field Validation - List dialog box is shown in the following image.

1. To enter new acceptable field values, click the Add new item button, type the text for the value, and press Enter.

2. To edit an existing value, select it, make any changes, and press Enter.

3. To delete an existing value, select it, and click the Delete selected items button.

4. To change the order of the values, use the move item up in the list and the move item down in the list buttons.

5. When you are done, click OK.
**Note:** When populating a Static list, make sure you scan the data source for all possible values and enter them into the list. If you leave a value off the list that is in a current record and that record is selected for update, the value for the bound column will change to the first item on the Static List.

**Tip:** If a field is not required and you want to give your user the option to leave it blank, put an empty entry in as the first item in your Static list.

**Procedure:** **How to Use a Dynamic List to Validate a Field**

When you choose the *Dynamic List* option to validate a field, you specify a field in the data source that contains the possible values. At run time, a list of values is retrieved from that data source and the user can then select one of these values from a drop-down list.

The real power of Dynamic Lists is that you can add items to the lists in your Update Assist applications without having to make changes to the form code of the application. Static lists require you to edit the forms in your Maintain application using the Maintain Development Environment. For example, if you choose to use a flat file as the source of items in your lists, you can simply add items to the flat file or export a new flat file from your data source to change the list. You do not need to change a line of application code.

1. In the Open dialog box, select a Master File for the data source containing the values for the field and click *OK*. You can use any data source type supported by WebFOCUS.
2. The Field Validation - File dialog box opens, as shown in the following image. Select the name of the field in the data source that contains the values you want to validate against and click OK. (If you want to select a different data source, click Browse.)
Update Assist (Step 2 of 4): Selecting Navigation and Style Options

The second window of Update Assist is where you determine what the user interface for your Update Assist application will look like. The following image shows the Navigation Options window.

This window contains the following fields and options:

**Prompt user to enter database security information (DBA)**

Generates a page prompting the user to enter a password to access the data in the data source. Use this option if data source security is enabled.

The application will store the password in a cookie, so the user will only be prompted for it once.

**Key values selected via tree**

Generates a form in which the user selects records using a hierarchical tree control.

**Key values selected via combobox**

Generates a form in which the user selects records using a combo box.
Key values entered by user

Generates a form in which the user selects records by entering key values.

Note: This requires that the user knows the actual values for the key values.

No key values required

Generates a Maintain procedure to be called by another procedure (usually a WebFOCUS report) with the appropriate values to fill out the screen. To see how to create the WebFOCUS report that calls this type of Update Assist project, see Calling an Update Assist Procedure From a WebFOCUS Report on page 358.

Themes

Adds a color scheme for your Update Assist application. You can add your own Cascading StyleSheet (CSS) to the form. You automatically see a preview of the color scheme when you select a theme.

Preview

Displays the selected navigation, theme, and fields.
Update Assist (Step 3 of 4): Selecting Output File Options

In the third window of Update Assist, the names of the files that Update Assist will generate are listed. You can rename these files if necessary. The following image shows the Output File Options window.

This window contains the following fields and options:

**Output Filenames**

Lists the files that will be created by Update Assist.

**Rename**

Allows you to change the selected file name.
Update Assist (Step 4 of 4): Confirming Selections

In the last window of Update Assist, you confirm all of your settings and generate the application files. The following image shows the Summary window.

This window contains the following fields and options:

**Project summary**

Lists the files that Update Assist is creating and summarizes the operations the project can perform on the data source. To make changes, press the Back button until you reach the screen containing the information you want to change.

**Make this project use compiled Maintain procedures**

Select this option when you are finished debugging your project. Compiled procedures run more quickly, but it takes time to compile them.

**Automatically run this project when you press the 'Finish' button**

Clear this option if you do not wish to run the project immediately, for instance, if you wanted to perform further editing on your project files.
Open maintain file for edit

Immediately opens the Maintain file for editing in the Maintain Development Environment (MDE).

About Your Update Assist Application

Once you click Finish in the last Update Assist window, Update Assist generates the files needed for your application and deploys and runs the application, if specified. This section describes how to run your application from outside this window and about how Update Assist applications work.

Procedure: How to Run Your Update Assist Application

Once you have generated an Update Assist application, you can run it any time your WebFOCUS Server is running. To do this, navigate to the URL:

http://yourmachine/approot/appname/launchname.htm

where:

- **yourmachine**
  
  Is the name of your machine on the network. If you are at the machine where the WebFOCUS Server is running, you can type localhost.

- **appname**
  
  Is the name of your project.

- **launchname.htm**
  
  Is the name of the HTML launch file for the Update Assist application. The default value for launchname is the name of your Master File.

Reference: Files Generated By Update Assist

Update Assist generates the following files:

<table>
<thead>
<tr>
<th>File name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MasterFileName.htm</td>
<td>Is the launching HTML file for your application and contains user interface information.</td>
</tr>
<tr>
<td></td>
<td>where:</td>
</tr>
<tr>
<td>MasterFileName</td>
<td>Is the name of your Master File.</td>
</tr>
<tr>
<td>File name</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>MasterFileName_navbar.htm</code></td>
<td>Contains more user interface information and determines the appearance of the Tree, Combo Box, and Edit Box options. For more information on ways you can edit this file, see Customizing the Tree Control on page 355.</td>
</tr>
<tr>
<td><code>mntSignon.htm</code></td>
<td>Contains the window prompting you for your data source password (only if you selected Prompt user to enter database security information (DBA) in Step 3).</td>
</tr>
<tr>
<td><code>SegmentName.mnt</code></td>
<td>Contains the Maintain language code to display and modify the fields in that segment of the data source. You will see a Maintain procedure for each segment that you selected in Step 1.</td>
</tr>
<tr>
<td><code>SegmentName_validationInit.js</code></td>
<td>Contains the JavaScript code to automatically validate user-entered values (only if you selected Automatic validation for any field). You will see a JavaScript file for each segment that you selected in Step 1.</td>
</tr>
</tbody>
</table>

**Reference: Working With Empty or New Data Sources**

- **In tree navigation:** If you select the Add option for any data segments that do not contain data, the Tree will display a period (.) for the null segment. You can right-click on the period (.) to enter new data for that segment.

- **In combo box navigation:** For any data segments that do not contain data, the combo boxes display a [New] command. This option enables you to enter new records.

**Editing Your Update Assist Application**

When you click Finish in the Update Assist (Step 4 of 4) - Summary dialog box, the files that comprise your Update Assist project are generated. For a list of files, see Files Generated By Update Assist on page 353.

If you wish, you can perform further editing on these files.
Customizing the Tree Control

The Tree control option for Update Assist applications enables users to find the record they are looking for by navigating through a tree hierarchy. You can add the following JavaScript variables in the MasterFileName_navbar.htm file to control this hierarchy:

- **maxNodesPerFolder.** An integer value that controls the maximum number of nodes that can be displayed in each folder. If the number of nodes exceeds this number, the Tree control automatically splits the nodes using subfolders labeled with a range of the node values contained within.

- **maxFolders.** An integer value that determines the maximum number of folders that can be rendered. The total number of nodes that will be rendered will be divided among these folders.

These variables and the values you wish to set for them can be added to the area of code inside the MasterFileName_navbar.htm file that generates the tree, using the syntax

```javascript
maintainTree.maxNodesPerFolder =
and
maintainTree.maxFolders =
```

prior to the maintainTree.init() code.

For example:

```javascript
var maintainTree = new maintainTreeInfo(constructFields(),mCallBack);
linestyle = "white";
maintainTree.maxNodesPerFolder= 8;
maintainTree.maxFolders = 20;
maintainTree.init();
```

**Note:** If the maxNodesPerFolder and maxFolders values are set in a way that would cause a conflict, the setting for maxFolders takes precedence over maxNodesPerFolder.
Calendar Control for Date-Formatted Fields

A calendar icon appears next to changeable date-formatted fields. When a user clicks the calendar icon, a calendar appears, as shown in the following image. Any date selected on this calendar is entered into the date field. Users can also enter dates into the date field manually.

Date-Stamping Fields

Many DBMSs allow you to create a time stamp field. This automatically fills the field with the current date and/or time and saves the user having to do it. There are many reasons at an application level for doing this. The most common is to give reporting applications some way to track when a record was first created or when each change was entered.
**Procedure:** How to Date-Stamp a Field in an Update Assist Application

To date-stamp a field in an Update Assist application, so that when a user clicks *New*, the application can set the initial value of the field to the current date in the stack before it is displayed in the form:

1. Open the `SegmentName.mnt` file in the Maintain Development Environment.
2. Add the following line of code to the top of the Maintain, just above Case Top:
   ```
   MODULE IMPORT(MNTUWS);
   ```
   This imports the library of functions included with WebFOCUS Maintain.
3. Scroll down to the newrecord case and add the following code right below the first Stack Clear statement:
   ```
   COMPUTE TheDate/MDY = Today();
   COMPUTE stack.datefield = TheDate;
   ```
   where `stack` and `datefield` are the stack name and field name to which you want to assign the current date.

**Note:** If you have multiple fields that need to be set to the current date, you only need to set the variable `TheDate` once and can reuse it as many times as you need.

**Example:** Date-Stamping a Field in the MOVIES Data Source

If you wanted the Release Date field from the MOVIES data source to contain the current date, your code would look like this:

```
COMPUTE TheDate/MDY = Today();
COMPUTE Movinfo_stack.RELDATE = TheDate;
```

**Auto-numbering Fields in Update Assist Applications**

Some DBMSs allow you to create an *auto-number* field. This automatically fills the field with a sequence number that is the last record index plus one. This saves the user having to make up an arbitrary key for the record.
Procedure: **How to Auto-Number a Field in an Update Assist Application**

To auto-number a field in an Update Assist application, so that when a user clicks New, the application can set the initial value of the field to the next sequence number in the stack before it is displayed in the form:

1. Open the SegmentName.mnt file in the Maintain Development Environment.
2. Scroll down to the newrecord case and add the following code right below the first Stack Clear statement:

   ```
   Stack clear SegmentNameStk;
   For all next MasterFileName.SegmentName.autonum into SegmentNameStk;
   NextVal/I5 = SegmentNameStk(SegmentNameStk.FOCCOUNT).val + 1;
   Stack clear SegmentNameStk;
   ```

   **Note:** If you are using an external DBMS that directly supports Date Stamp field types, you will not need to use this technique.

Continuing to Display Current Values After a New Action

By default, Update Assist clears all text boxes and controls in the form on a New action. You can have the values stay in the text boxes by editing the SegmentName.MNT file.

For example, users of some types of applications may be entering many similar records, one after another, and would like to display a record, then essentially have the New action display a copy of the record which they can edit before clicking Save.

Procedure: **How to Continue Displaying Current Values After a New Action**

1. Open the SegmentName.MNT file and go to the newrecord case.
2. Comment out the line that clears the stack, using a double dollar sign ($$).

Calling an Update Assist Procedure From a WebFOCUS Report

One way to use an Update Assist procedure is to call it from a WebFOCUS report. You can set up the WebFOCUS report so that a user can click on a row in the report and open the Update Assist procedure with the data from the row of the report.

Procedure: **How to Call an Update Assist Maintain Procedure From a WebFOCUS Report**

1. Create an Update Assist procedure that includes the following selections:
   a. In the Update Assist (Step 2 of 4) - Navigation Options window, select No key values required for your user interface.
b. In the Update Assist (Step 3 of 4) - Output File Options window, note the names of your procedures. By default, Update Assist names them after the segment they are modifying in the data source.

2. In Report Painter, create a report using the same data source you used for the Update Assist project. The report must contain the key fields in the segment you want to update (if you do not want to view them in the results, you can make them invisible).

3. Right-click the column you want to make clickable in the report, then select Options from the 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 cannot drill down from the title of the column in the report, only by clicking on the column data.

5. In the Drilldown Type drop-down list, select Maintain Procedure. This specifies that when the user clicks on one of the last names in the report, a Maintain procedure is executed.

6. In the Procedure name field, type the name of the Maintain procedure you want to execute.
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:
      \textit{segname\_fieldname\_Edit}
      where:
      \textit{segname}
      Is the name of the segment that contains the key field.
      \textit{fieldname}
      Is the name of the key field (a key field uniquely identifies a record in a segment).
      \textit{\_Edit}
      Must be typed exactly as shown. This is the convention for naming variables in Update Assist.
   c. Click OK.
   d. Repeat for each parameter that is to be passed to the Maintain procedure.

   \textbf{Note:} You must pass all key fields needed to locate the record you want to update, otherwise, the Maintain procedure will not have enough information to uniquely identify a record. For example, in the CAR file, if you wanted to update a particular car model, you would need to pass the country and car, as well as the model type to locate the model field.

8. In the Drill Down tab, ensure the key field is selected in the list of parameters.
9. Click OK.
10. Close your procedure and save it.

When you run your report, you will see that all of the items in the selected column of the report are underlined and clickable. Clicking any item on the report opens the Update Assist form with the information for that item already filled in.
Calling an Update Assist Project From a WebFOCUS Report Example

This example describes how to create a report in 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 an employee from the empdata data source. See Creating an Update Assist Project For the Empdata Data Source on page 362.

2. Create a report in Report Painter that contains a simple list of the employees in the empdata data source. See Creating a Report With the Empdata Data Source on page 364.

When you are done, you will have a WebFOCUS report that displays a list of customers in the employee data source. Clicking on the PIN adjacent to the last name of someone in this report will bring up a form where you can change information about the employee, or delete the employee from the data source. The result is shown in the following image.
Example: Creating an Update Assist Project For the Empdata Data Source

The following is an example of creating an Update Assist project for the empdata data source.

1. Create a project.
2. Add the IBISAMP directory to the viewable directories for the project.
3. Add the empdata Master File to your project.
4. Right-click the empdata Master File and click Update Assist in the pop-up menu.
   
   The Update Assist (Step 1 of 4) - Select Segment & Field Options window opens.

5. In the Update Segment Options section, set Update to Yes.
6. In the Field View Options section, set Changeable to Yes for all fields except PIN, which is a key field. The window is shown in the following image.

7. Click Next.
   
   The Update Assist (Step 2 of 4) - Navigation Options window opens.

8. Select No key values required and a color scheme from the supplied examples and click Next.
The Update Assist (Step 3 of 4) - Output File Options window opens. You should only see one file here, empdata.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.

9. Click Next.

The Update Assist (Step 4 of 4) - Summary window opens.

10. Review the summary of options you have chosen, and ensure Automatically run this project when you press the 'Finish' button is selected.

11. Click Finish.

WebFOCUS Maintain creates the .mnt files based on the options you selected in Update Assist, and deploys them. During deployment, the Deploying Application window opens.

12. 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, as shown in the following image.
**Example: Creating a Report With the Empdata Data Source**

1. Create a procedure that uses 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. Type update in the File Name field, select Report Painter from the Create with drop-down list, and click Open.
      
      The Open dialog box opens.

   c. Select empdata and click Open.
      
      Report Painter opens.

2. Place the LASTNAME, FIRSTNAME, DEPT, and TITLE fields in the report. Select the PIN field and click By to sort by the PIN field.

3. Right-click the PIN field on the report, and select Options from the 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 cannot drill down from the column title PIN, only from the data in the report.

6. In the Drill down Definition drop-down menu, select Maintain Procedure. This specifies that when an user clicks on one of the last names in the report, a Maintain procedure is executed.

7. Select empdata.mnt from the Procedure Name drop-down list (or, if you renamed your procedure when you created your Update Assist application, enter the new name).
8. Specify the parameter that gets passed to the Maintain procedure empdata:
   a. Click Add in the With Parameters section.
      The Drill Down Parameter dialog box opens.
   b. Select EMPDATA_PIN_Edit from the Parameter name box.
   c. Select the Field option and select EMPDATA.EMPDATA.PIN in the Parameter value section.
   d. Click OK.
The Drill Down Parameter dialog box opens, as shown in the following image.

9. Close your procedure and save it.
When you run your report, you will see all of the PIN values in the report are underlined and clickable, as shown in the following image.

![WebFOCUS Report](image)

<table>
<thead>
<tr>
<th>PIN</th>
<th>LASTNAME</th>
<th>FIRSTNAME</th>
<th>DEPT</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000010</td>
<td>VALINO</td>
<td>DANIEL</td>
<td>MARKETING</td>
<td>MARKETING EXECUTIVE</td>
</tr>
<tr>
<td>000000020</td>
<td>BELLA</td>
<td>MICHAEL</td>
<td>MARKETING</td>
<td>INDUSTRIAL MARKETER</td>
</tr>
<tr>
<td>000000030</td>
<td>CASSANOVA</td>
<td>LOIS</td>
<td>SALES</td>
<td>SALES MANAGER</td>
</tr>
<tr>
<td>000000040</td>
<td>ADAMS</td>
<td>RUTH</td>
<td>MARKETING</td>
<td>MARKETING DIRECTOR</td>
</tr>
<tr>
<td>000000050</td>
<td>ADDAMS</td>
<td>PETER</td>
<td>SALES</td>
<td>EXECUTIVE MANAGER</td>
</tr>
<tr>
<td>000000060</td>
<td>PATEL</td>
<td>DORINA</td>
<td>MARKETING</td>
<td>MARKETING DIRECTOR</td>
</tr>
<tr>
<td>000000070</td>
<td>SANCHEZ</td>
<td>EVELYN</td>
<td>ACCOUNTING</td>
<td>MANAGER</td>
</tr>
<tr>
<td>000000080</td>
<td>SO</td>
<td>PAMELA</td>
<td>MARKETING</td>
<td>SENIOR SALES EXEC.</td>
</tr>
<tr>
<td>000000090</td>
<td>PULASKI</td>
<td>MARIANNE</td>
<td>PERSONNEL</td>
<td>EMPLOYEE COORDINATOR</td>
</tr>
<tr>
<td>000000100</td>
<td>ANDERSON</td>
<td>TIM</td>
<td>ACCOUNTING</td>
<td>SUPERVISOR OF AP/AR</td>
</tr>
<tr>
<td>000000110</td>
<td>RUSSO</td>
<td>ANTHONY</td>
<td>CUSTOMER SUPPORT</td>
<td>PRODUCT DISTRIBUTOR</td>
</tr>
<tr>
<td>000000120</td>
<td>WANG</td>
<td>KATE</td>
<td>CONSULTING</td>
<td>CORPORATE CONSULTANT</td>
</tr>
<tr>
<td>000000130</td>
<td>CVEK</td>
<td>MARCUS</td>
<td>PROGRAMMING &amp; DVLPMT</td>
<td>EXEC SECRETARY</td>
</tr>
<tr>
<td>000000140</td>
<td>WHITE</td>
<td>VERONICA</td>
<td>CUSTOMER SUPPORT</td>
<td>MANAGER</td>
</tr>
<tr>
<td>000000150</td>
<td>WHITE</td>
<td>KARL</td>
<td>CONSULTING</td>
<td>SENIOR CONSULTANT</td>
</tr>
<tr>
<td>000000160</td>
<td>HIRSCHMAN</td>
<td>ROSE</td>
<td>CUSTOMER SUPPORT</td>
<td>MNGR OF CUST SUPPORT</td>
</tr>
<tr>
<td>000000170</td>
<td>MORAN</td>
<td>WILLIAM</td>
<td>ADMIN SERVICES</td>
<td>ADMINISTRATOR</td>
</tr>
<tr>
<td>000000180</td>
<td>GORDON</td>
<td>LAURA</td>
<td>Admin Services</td>
<td>Asst Administrator</td>
</tr>
<tr>
<td>000000190</td>
<td>MEDINA</td>
<td>MARK</td>
<td>Sales</td>
<td>Sales Specialist</td>
</tr>
<tr>
<td>000000200</td>
<td>LASTRA</td>
<td>KAREN</td>
<td>Sales</td>
<td>VICE PRES</td>
</tr>
<tr>
<td>000000210</td>
<td>DUBOIS</td>
<td>ERIC</td>
<td>Sales</td>
<td>Sales Executive</td>
</tr>
<tr>
<td>000000220</td>
<td>LEWIS</td>
<td>CASSANDRA</td>
<td>Admin Services</td>
<td>Marketing Supervisor</td>
</tr>
<tr>
<td>000000230</td>
<td>NOZAWA</td>
<td>JIM</td>
<td>Personnel</td>
<td>VICE PRES</td>
</tr>
<tr>
<td>000000240</td>
<td>KASHMAN</td>
<td>XOLANDA</td>
<td>Personnel</td>
<td>Exec Secretary</td>
</tr>
</tbody>
</table>

Clicking a PIN on the report opens the Update Assist form with the information for that name already filled in.

**Usage Notes**

The following are known issues when using WebFOCUS Update Assist:

- Update Assist will not prevent the use of special characters or wildcard designations such as $* when entering data. Entering such character combinations can cause unexpected results.
- When renaming Update Assist HTML files, the ampersand character (&) is not supported.
Usage Notes

- Using the browser Refresh action while running an Update Assist application can cause unexpected results and is not recommended.

- Update Assist does not allow updates of cross-referenced segments. You must run the Update Assist on the individual Master Files and create separate update procedures.
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 Files 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.

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.

**In this chapter:**
- Text Editor
- The Other Component
- The Comment Component
- Creating a Text File or a Procedure Component as Text
- Opening Application Components as Text
- Finding and Replacing Text
- Changing Text Color and Case Size
- Adding Headings and Footings
- Using Bookmarks to Move Within a File
- Running a Procedure From the Editor
- Opening a Graphical Tool From the Text Editor

**Text Editor**

Since the Text Editor is fully integrated with the graphical toolset, changes you make to source code are immediately reflected in the graphical tools, and vice versa.

The Text Editor window includes:
A menu bar, which provides access to all options available in the Text Editor.

The editor window, which contains the open file, displaying color-coded text.

The editor toolbar, which contains icons for frequently performed tasks, such as, saving and printing, cutting, copying, and pasting, undoing and redoing, finding text, toggling bookmarks, finding bookmarks, and clearing bookmarks, and accessing online help.

When you are editing a procedure, the following additional icons appear: Report, Graph, Define, Join, Set, Allocation, Use, and SQL. 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 382.

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 378.

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 382 for details.

You can access any text editor that is registered with your operating system based on the program that is set up to be the default program that opens the selected file type. For example, in Windows Explorer, select a .fex file, then right-click and select Properties from the context menu. Click Change and select Notepad as the default program to open this file type if it is not already associated with an application. The option to open procedures (.fex files) with Notepad is available through Developer Studio.

This option is valuable for developers who want to take advantage of the WebFOCUS language and use the editor of their choice.

Accessing Text Editors

You can access any text editor by right-clicking a file within the Developer Studio Explorer. A context menu for the file appears with the following default options:

- **Open in (Task Viewer/Text Editor/Registered Program)** to edit the components of the procedure. Depending on this setting, the following options order will change.

- **Open in (Task Viewer).** Depending on the product settings, this option may show Open in (Text Editor) or Open in (Registered Program), for example, Notepad.
- **Open in Text Editor** to edit the procedure code using the Developer Studio Text Editor.
- **Open in (Registered Program)** to edit the procedure code using an external editor, such as Notepad.
- **Open in auto detected tool** to open the procedure in the tool used to create it (for example, Report Painter, HTML Composer, and so on). You must use this option if you need to open the file in the HTML Composer because it is the only way to open procedures created in the tool used to create them. You cannot open procedures created with those tools from the Task Viewer.

**Procedure: How to Set Up the Default Action for the Open Option**

1. In the Explorer window, select **Options** from the Window menu. The Developer Studio Options dialog box opens at the General tab.
2. From the Default file editor drop-down list, select one of the following:
   - **Edit in DevStudio tool** to edit a procedure in the Developer Studio Task Viewer.
   - **Edit in Text Editor** to edit a procedure in the Developer Studio Text Editor.
   - **Edit in Windows registered tool** to edit a procedure in Notepad, Wordpad, or another program as registered with the operating system.

**The Other Component**

The Other component uses the Text Editor as a scratch pad on which you can create and edit a procedure or a component of a procedure. You can access the Other component from the following button on the component toolbar.

The Other component is particularly useful when you want to add code that is not represented by a graphical tool (for example, Dialogue Manager code for managing the flow of control within a procedure). For information on Dialogue Manager, see *Managing Flow of Control in an Application* in the *Developing Reporting Applications* manual.

If Developer Studio recognizes the code you type as a report or graph request or a Define or Join command, the next time the procedure opens, the appropriate component type (for example, Report, Graph, Define, or Join) displays in the Procedures window. Similarly, if you type text that is preceded by a dash and asterisk (-*), 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 372). For related information, see *Creating a Reporting Procedure* on page 215.
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 click **Open** from the shortcut menu. The Procedure window opens.

2. Click 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, as shown in the following image.

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 a component connector (yellow diamond), then drag the arrow pointer to the Other option on the component toolbar. The Text Editor opens.

2. Type your comment, beginning with the characters -*. Each line of the comment must begin with -*.

3. Close the Editor and update the procedure when prompted. The text is added to the procedure as a new Comment component.

**Adding and Removing Comments in the Text Editor**

A comment option is available in the Text Editor, which enables you to add and remove comments for selected lines of WebFOCUS code. You may use the Text Editor to create, view, and edit the source code for procedures.

The comment option gives you the ability to comment out an entire block of code, without manually typing in a dash and asterisk (*-) on every line.

**Procedure: How to Add or Remove Comments in the Text Editor**

The option to add or remove comments is available from the stand-alone Text Editor and the Text View of the Procedure Viewer.

**Note:** The comment option is not available while viewing the Source code from Report Painter.

1. To open the procedure in a Text Editor:
   a. In the Explorer view, right-click a Procedure file (.fex) and select *Edit in Text Editor*, as shown in the following image.
b. In the Procedure Viewer, right-click a procedure component (for example, Report, Graph, Other), and select *Edit Text*, as shown in the following image.

![Image of Procedure Viewer with right-click options]

c. In the Explorer view, select *Text Editor* from the Command menu, as shown in the following image.

![Image of Command menu with Text Editor option]

Select a procedure from the Open dialog box, or create a new procedure file (.fex).

The Text Editor opens.

2. Select the portion of WebFOCUS code that you would like to comment and select *Add Comment* from the Edit menu.

**Note:** You may apply comments to a block of code or to a single line.
The following image shows the Add Comment option.

A dash and asterisk (\*-\*) is added to the selected block of code.

```
BY &ProductCategory.(OR.<CD Players,CD Players>,<Camcorders,Camcorders>,<Camera
-\* BY &ProductType.(AND.(ACCEPT PROTYPE FROM CENTORD)).Select Product Type.|AC
```

**Note:** Comments (-*) are added to the beginning of the selected lines.

3. To remove the comment, select the commented lines of code and select Remove Comment from the Edit menu.

The comment (-*) is removed from the beginning of the selected lines of code.

**Tip:** You may also select the Add and Remove Comment selections from the right-click menu and by using the shortcut keys. Ctrl+M adds a comment. Ctrl+R removes a comment.

**Creating a Text File or a Procedure Component as Text**

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 382.

You also can create a text component in a Procedure using the Other component.

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

1. Choose Text Editor from the Command menu.
2. In the Open dialog box, enter a new file name in the File name box. For a file other than a procedure (.fex) or a Master File (.mas) include an extension.

3. Choose a file type from the File of type list. For a procedure (.fex) or a Master File (.mas), the file extension is picked up from this selection.

4. Click the Open button. The Text Editor opens.
   
   Type your text and close the Text Editor. When prompted, save your file.

   The new file is saved in the selected folder with the appropriate extension.

**Procedure: How to Create a Text Component in a Procedure**

1. Create a new procedure:
   
   - With the Procedures folder highlighted, select New from the File menu.
   
   - or
   
   - Right-click the Procedures folder and select New from the pop-up menu, then select Procedure.

   The Add Procedure dialog box opens, as shown in the following image.

   ![Add Procedure Dialog Box](image)

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 Procedure Viewer opens.
4. Click a component connector (yellow diamond) and select Other from the menu.
   The Text Editor opens.

5. Type your text and close the Text Editor. When prompted to update the procedure (FOCEXEC),
   click Yes.

6. Save and 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 Text
Editor option in the Command menu or from a project folder.

Procedure: How to View an Application Component as Text

You can use any of the following three methods to expose the underlying code or text for the
selected file in the editor.

1. Right-click a file, such as a procedure or Master File, in an Explorer folder.
2. Select Edit As Text from the shortcut menu.
3. Highlight a file, such as a procedure or Master File, in an Explorer folder.
4. Select Edit as Text from the File menu.
5. Choose Text Editor from the Command menu. The Open dialog box appears.
6. Specify the type of file you want to view or edit. FOCEXEC is the default.
7. Select a file, then click Open.

Procedure: How to View a Procedure Component as Text

In addition to viewing the code for an entire procedure from a folder, you can examine and edit
the code for an individual procedure component.

1. Right-click a procedure folder in the Explorer and select Open from the shortcut menu.
2. Right-click a procedure component (for example, Report, Graph, Other), then select Edit As
   Text.

You will see the underlying code for the selected procedure component in an Editor window.
Finding and Replacing Text

The Find dialog box enables you to search a file for specified text or numbers. You can also include special characters (for example, paragraph marks and symbols) in your search criteria. Replace enables you to search for a word and replace it with another.

Options available to narrow or speed your search are:

- **Match Case.** Finds only those occurrences with the exact combination of uppercase and lowercase letters specified in the Find What box.
- **Regular Expression.** Searches for text using the wildcard characters *, (matches any number of characters) and a question mark, ?, (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 381.

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.

<table>
<thead>
<tr>
<th>Syntax Coloring Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Regular or default text in the Text Editor.</td>
</tr>
<tr>
<td>Text Selection</td>
<td>Text that has been highlighted in the Text Editor.</td>
</tr>
<tr>
<td>Number</td>
<td>Numbers in the Text Editor.</td>
</tr>
<tr>
<td>Operator</td>
<td>Operators in the Text Editor.</td>
</tr>
<tr>
<td>Comment</td>
<td>Any text that follows a hyphen (-) and an asterisk (*). For Master Files (.mas), any text that follows a dollar sign ($).</td>
</tr>
<tr>
<td>Keyword</td>
<td>Words in the FOCUS language.</td>
</tr>
</tbody>
</table>
**Syntax Coloring Option** | **Description**
--- | ---
End of Line | Text designating the last line of syntax.
Double Quoted String | Any text between double quotation marks.
Single Quoted String | Any text between single quotation marks.

**Procedure: How to Change the Foreground and Background Text Color**

1. With the Text Editor open, select *Options* from the View menu.
   
   The Editor Options dialog box opens.

2. Click *Font/Color*. The Font and Color Settings dialog box opens.

3. Under the Color group box, highlight the syntax coloring option you want to change.

4. Deselect the Automatic check box next to the Foreground and Background drop-down lists.
   
   The drop-down lists become active.

5. Select a Foreground color and Background color from the respective drop-down lists.

6. Click *OK*.

7. Click *OK* to close the Font and Color Settings dialog box.

8. Click *OK* to close the Editor Options dialog box.

**Procedure: How to Change the Font**

1. With the Text Editor open, select *Options* from the View menu.

   The Editor Options dialog box opens.

2. Click *Font/Color*. The Font and Color Settings dialog box opens.

3. Click *Choose Font*.

   The Font dialog box opens.

4. Select a font and type from the Font drop-down list and a size from the Size drop-down list.

5. Click *OK*.

6. Click *OK* to close the Font and Color Settings dialog box.

7. Click *OK* to close the Editor Options dialog box.
Procedure: How to Change the Text to Uppercase or Lowercase

1. In the Text Editor, highlight the text you want to change.
2. Do one of the following:
   - Select Change 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 page. They do not appear when the report is displayed in the web browser.

Procedure: How to Add Headings and Footings

1. From the File menu, select Header and Footer. The Header and Footer dialog box displays.
2. Enter the heading text in the Header field or the footing text in the Footer field. You can do this by typing text you want included, or by clicking one of the buttons located on the right of the dialog box:
   - Insert Filename includes the name of the procedure.
   - Insert Page Number includes the page number.
   - Insert Number of Pages includes the total number of pages.
   - Insert Date includes the date.
   - Insert Time includes the time.
   - Default restores the Header and Footer fields to the default display.
3. Select a justification for the heading and footing from the Header and Footer section:
   - Left left-justifies the heading or footing.
   - Center centers the heading or footing.
   - Right right-justifies the heading or footing.
4. Click OK.

**Using Bookmarks to Move Within a File**

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 and 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. The results of a report or graph are displayed in the browser. For other types of components, results simply take effect (for example, virtual fields and fields from the joined data sources will be available for selection in the Fields windows).

Procedure:  How to Run a Full Procedure as Code

1. Open the procedure in the Text Editor.
2. Run the procedure using any of the following methods.
   - Select Run from the File menu.
   - Click the Run icon on the toolbar.
   - Press the F5 shortcut key.

The results appear in the browser.

Procedure:  How to Run a Procedure Component as Code

1. Select the procedure, right-click it, and select Edit Text from the pop-up menu.
2. Place the cursor within or highlight the specific text you wish to run.
3. Right-click and select Run tool (where the tool corresponds to the type of code you have selected, for example, Report, Define, Join, Set, and so on).

Procedure:  How to Run Several Components From the Text Editor

1. Open a procedure in the Text Editor or Other component.
2. Highlight the code to be run.
3. Right-click and select Run.

Opening a Graphical Tool From the Text Editor

Since the Text Editor is fully integrated with the graphical toolset, you can open the following tools from the editor or the Other component: Set, Allocation, Use, Join, Define, Report, Graph. The location of the cursor in the procedure code determines where the code is returned in the Editor window. When you close a tool that you opened from the Text Editor, you return to the Text Editor. Any edits you saved in the graphical tool are reflected in the code.
Procedure: **How to Add Code With a Graphical Tool From the Text Editor**

In the Text Editor or the Other component:

1. Open a procedure or a procedure component.
2. Position the cursor on a blank line at the point where you want to insert the code generated by the tool you are about to open.

   **Tip:** If the cursor is not located on a blank line, the new code will be inserted before the block of contiguous code in which the cursor is positioned.

3. Click one of the following icons on the toolbar: Set, Allocation, Use, Join, Define, Report, Graph. The corresponding graphical tool opens.
4. Create the procedure component and close the tool.

You return to the Text Editor, where the new code has been inserted.

Procedure: **How to Edit Code With a Graphical Tool From the Text Editor**

In the Text Editor or the Other component:

1. Open a procedure or a procedure component.
2. Position the cursor on the line containing the code you want to edit, and click the Launch FOCUS Tool button on the toolbar.

   The appropriate graphical tool opens.

3. Edit the procedure component and close the tool.
4. When you are asked if you want to update the procedure, click Yes.

You return to the Text Editor, where the code has been revised.
WebFOCUS Alerts enable Managed Reporting Developers to distribute an entire report or selected sections of a report only when certain test conditions are met. The Alert tool is available from WebFOCUS Managed Reporting.

**In this chapter:**
- Alert Reporting Overview
- Components of an Alert
- Creating an Alert
- Checking Scheduled Alerts

### Alert Reporting Overview

In the business climate today, cutting costs and increasing profitability is critical. Organizations create and distribute reports for evaluating and taking action when management defined conditions are not met. The evaluation can largely be automated using Alert reporting.

Alert reporting automates the evaluation of a defined condition to determine whether a report should be submitted. Recipients of the report know before they open it that they are receiving this report because there is an exception in their area of responsibility.

A data-driven Alert is an event that is prompted by a guideline you define. This guideline can vary from simple to complex. For example, a simple guideline might be to alert a sales manager when an order that exceeds $1 million is entered. An example of a more complex guideline might be to alert a department manager when cumulative expenses exceed budget for any category. The report sent to the manager as a result of the Alert could show the most recent transactions, the amounts, and the sources.

Alert reporting is an integrated feature of WebFOCUS Managed Reporting and ReportCaster. An Alert report can be run interactively or by authorized users. You can use ReportCaster to schedule and distribute an Alert report. Burst capability can send targeted information to people who need it, so that there is less information for each person to analyze.

Alerts work with all email clients. They are supported on the following mobile devices: iPhone®, iPad®, BlackBerry®, and email-enabled mobile phones. Alerts can also be sent to any PC or laptop computer with email capability.
Alerts provide the following:

- **Alert tools.** You can quickly set up test conditions (rules) that determine if an Alert condition should be triggered (true). This is called the Alert test. When the Alert test is triggered (true), the Alert result is run. An Alert result can be an existing report procedure or you can create the Alert. This is called the Alert result.

- **Scheduling flexibility.** Optionally, you can schedule an Alert to run as often as every minute to evaluate the Alert test conditions.

## Components of an Alert

The Alert feature requires the following components:

- **WebFOCUS Reporting Server.** This server is responsible for processing the procedure, accessing and retrieving the requested data, and creating the report when the Alert test is triggered (true).

- **Managed Reporting.** This environment is used to create and manage Alerts.

An Alert consists of the following:

- Alert test
- Alert result

An Alert test is required to check whether or not a set of defined test conditions (rules) is met (true or false). An Alert test can be a *WebFOCUS Test*, which is a simple TABLE request that you create to define the condition to test for. Alternatively, an Alert test can be a *Test for File Existence*, which checks for the existence of a file that is accessible to the WebFOCUS Reporting Server to which the Alert test is submitted. The file does not need to be in the APP PATH. It just needs to be in a location where it can be found by the Reporting Server.

**Note:** The *Test for File Existence* file does not need to be an actual file with an extension. Testing for the existence of a folder is also valid.

If the Alert test is triggered (true) and returns records, the Alert result request is processed. You have the option of creating the result report or selecting an existing Managed Reporting report you are authorized to run.

**Alerts Created in Developer Studio.** If you select *WebFOCUS Test* in Developer Studio, you are prompted to create the Alert test using the reporting tool. You can also choose the Editor. If you select the Editor, be sure to keep the TABLE request simple. The TABLE request should only contain the fields to evaluate the condition and an ON TABLE HOLD statement. When you save the request, it is validated to ensure you have included the ON TABLE HOLD statement.
You can create the Alert result or select an existing Managed Reporting report.

**Considerations:** One consideration when creating an Alert is the flexibility of editing the Alert result. It is recommended that you select an existing report for the Alert result request, as it is available to edit, run interactively, or schedule independently of the Alert. If you create the Alert result using Developer Studio within the Alert Wizard, you need to use the Alert Wizard and select the open option again to edit the request, because the Alert test and the report to be run are contained within the Alert procedure.

Once an Alert is created using the browser or Developer Studio, it is stored in the WebFOCUS Repository.

**Reference:** Alert Test Sample Code Created in Developer Studio With the Editor

In the sample code that follows, an Alert test is created with the Editor and a report to be executed is created with Report Painter. To keep the sample code simple, styling and report options, such as report headings and conditional styling, are not included in this request.

Notice that after the first TABLE request there is a –IF statement that checks if the number of lines in the report is equal to zero (–IF &LINES EQ 0). This checks to see if the Alert should be triggered (true).

If the number of lines is greater than zero, then there are records that satisfy the Alert test and the report is executed. This test condition determines if the second TABLE request, to execute the report, should be triggered (true).
The following is the Alert test sample code:

```plaintext
/* Created by Alert Wizard
   * version 1.0
   * start of the test report
   TABLE FILE CENTINV
   SUM CENTINV.INVINFO.QTY_IN_STOCK
   BY CENTINV.INVINFO.PRODCAT
   BY CENTINV.INVINFO.PRODTYPE
   BY CENTINV.INVINFO.PRODNAME
   BY CENTINV.INVINFO.PROD_NUM
   WHERE CENTINV.INFO.QTY_IN_STOCK LT 5000;
   ON TABLE HOLD
   END
   /* end of the test report
   /* start of the test
   -RUN
   -IF &LINES EQ 0 GOTO ALERTEXIT;
   /* end of the test
   /* start of the output report
   TABLE FILE CENTINV
   PRINT
   CENTINV.INVINFO.QTY_IN_STOCK
   CENTINV.INVINFO.PRODCAT
   CENTINV.INVINFO.PRODTYPE
   CENTINV.INVINFO.PRODNAME
   CENTINV.INVINFO.PROD_NUM
   WHERE CENTINV.INFO.QTY_IN_STOCK LT 5000;
   ON TABLE SUBHEAD
   "Products with Inventory Below 5000"
   ON TABLE SET PAGE-NUM NOLEAD
   ON TABLE NOTOTAL
   ON TABLE PCHOLD FORMAT HTML
   ON TABLE SET HTMLCSS ON
   END
   /* end of the output report
   /* start of the options
   /* ALERTEXIT
   /* End Alert Wizard
```
The Life of an Alert

Authorized users can run an Alert test interactively and optionally schedule it to be distributed by ReportCaster.

The following image shows basic Alert processing.

1. The Managed Reporting developer, using Alert Assistant or the Alert Wizard, creates an Alert test and designs the report to be run if the Alert condition is true.
2. If the Alert test is true, the WebFOCUS Reporting Server runs the Alert result.
3. If the Alert test is false, the Alert result is not run.
The following image shows scheduled Alert processing.

1. The Managed Reporting developer, using Alert Assistant or the Alert Wizard, creates an Alert test and designs a report to be run if the Alert condition is true.
2. If desired, an Alert schedule is created using ReportCaster.
4. When the ReportCaster Distribution Server processes a scheduled Alert procedure, it is submitted to the Reporting Server to evaluate if the Alert test is true.
   a. If the Alert test is false, the Alert is returned to the queue for the next scheduled time.
   b. If the Alert test is true, the Reporting Server runs the Alert result and returns the results to the ReportCaster Distribution Server. The ReportCaster Distribution Server distributes the Alert result to the recipients, as specified in the schedule.
5. The ReportCaster Distribution Server distributes the Alert result to the specified recipients, and allows you to specify various scheduling options.
6. The report is received by the recipient.
Creating an Alert

To facilitate the creation and scheduling of Alerts, the Alert Wizard guides you through a series of windows that complete the following tasks:

- Create an Alert test.
- Create a new report to be run or select an existing report.

If you are logged on to Managed Reporting as a Managed Reporting developer, you can access Alert Wizard through a Published Content or Private Content folder or subfolder within the tree.

When you use the reporting tool through Developer Studio to create an Alert test, a subset of the tabs and options is available, because the Alert test is used only to state the conditions or rules, of the test. Users do not see this report. This should be a simple report that states the selection criteria and an ON TABLE HOLD command. For sample code examples, see Alert Test Sample Code Created in Developer Studio With the Editor on page 387.

When an Alert test is evaluated and the result is true, a report (the Alert result, which is the second report you create) is executed. This is the report users see. Therefore, you can add styling to this report. More options are available in Report Painter when you are creating an Alert result.

When you create an Alert result report, it is not shown to the Managed Reporting end user by default since all content is Private by default.

Procedure:  How to Create an Alert in Developer Studio

As you use the Alert Wizard to create reports, you may see messages that state Report Saved. Your changes for the report have been saved but not written to the tree until the Alert is saved.

To save the Alert, click Finish in the Alert Wizard. If you exit the Alert Wizard before you click Finish, the Alert is not saved.

The following steps provide options for creating an Alert in Developer Studio.

1. Open WebFOCUS Developer Studio, and expand the tree to show the folders and subfolders.
2. Right-click a folder or subfolder. Select New, and then Alert from the context menus. The New Alert Wizard Procedure window opens, as shown in the following image.

3. Enter the name for the Alert, check any necessary options, and click Create. The Alert Wizard Welcome window opens.
4. Click Next.

The Alert Wizard, Alert Test window opens, as shown in the following image.

5. Select the Alert test.

- If you select WebFOCUS Test (the default), click Next. Proceed to step 6.
- If you want to check for a file, click the Test for File Existence option button.
  
  Type the full path to the file on the WebFOCUS Reporting Server that you want to check. Do not use a file name or a file path that includes spaces.

  Click Next.

  If the file is located, the Alert report is sent. Proceed to step 14 to create the Alert report (event procedure).

  **Note:** Testing a folder for existence is also valid. You can test a folder for existence if it is available on the WebFOCUS Reporting Server.

6. Select a tool to create an Alert test.

Select either the Procedure Viewer or Editor to create an Alert test.

- If you choose Procedure Viewer (recommended), proceed to step 7.
- If you choose Editor, proceed to step 11.

7. From the WebFOCUS Table List window, select a Master File and click OK.
The Procedure Viewer opens, where you can create selection criteria for your Alert test. If you do not see the reporting tool, double-click the Report icon in the Procedure Viewer.

8. Select your options. When you have finished, click Exit.

**Note:** When creating an Alert test, amper variables in a WHERE condition must have a value specified in the schedule. This is because Alerts run unattended and are not able to collect a value from a user. For more information, see the ReportCaster manual.

9. You are prompted to save your changes. Click Yes and close the Procedure Viewer.

You return to the Alert Wizard, New or Existing Test window.


11. If you chose Editor in step 6, the Editor window opens. Type your Alert test in the editor window.

**Note:**

- You must include an ON TABLE HOLD command in your Alert test.
- You can access graphical tools, such as the Join and Define tools from the Developer Studio text editor.

12. When you have completed the procedure, close the Text Editor. The Save Current Changes window opens.

13. Click Yes to save your Alert test. You return to the Alert Wizard, New or Existing Test window.
14. Click Next.

The Alert Wizard, Alert Event window opens, as shown in the following image. You can indicate which report to distribute when the Alert test is true. It can be a report you create or one that already exists.

- If you choose New WebFOCUS Report, click Next and proceed to step 15.
- If you choose Select WebFOCUS Report, click Next and proceed to step 17.

For information on which option to select, see the Considerations note in *Components of an Alert* on page 386.
15. From the Alert Wizard, Create the Report window, create a new report with the Procedure Viewer or Editor. The Alert Wizard, Create the Report window is shown in the following image.

![Alert Wizard, Create the Report window](image)

**Note:** From the Procedure Viewer Alert Event window, you can select a Report or a Graph.

- Selecting a *Report* opens Report Painter to create a more complex request.
- Selecting a *Graph* opens InfoAssist to create a robust graph.

16. When you have completed the report, click **Save**. Proceed to step 18.

17. Select your report from the tree, and click **Save**, as shown in the following image.
The Alert Wizard, Finish window opens.

18. Click *Finish*.

**Note:** The new Alert can be found under the folder, or subfolder, in the tree, identified by the Alert icon 🚨. Your Alert will automatically be executed if the test report is true. It is not necessary to schedule the Alert.

19. From the tree, right-click the newly created Alert, click *Schedule*, and then *Email.*
ReportCaster opens in a new browser window, as shown in the following image.

20. Update the fields as desired. Ensure that you have the correct Path, Procedure, and Server Name.

21. Click Alert.
The Alert Options dialog box opens, as shown in the following image.

22. Select one of the following options:

- **Automatically Reset.** This option continues to run the schedule at the specified time and checks to confirm that the Alert test is no longer true before the Alert test is evaluated again. It is best to use this option when you do not want to receive the Alert result again until after the exception condition has been addressed and occurs again. For example, when the quantity in inventory is greater than 20,000, the Alert is triggered. You do not want to receive the report again until after the inventory is less than or equal to 20,000 for all products and then goes back over 20,000 again.

- **Continue after Alert.** This option continues to run the schedule at the specified time and distribute the report for each time the Alert test is true. It is best to use this option when you want to be notified on the interval specified in the schedule that the Alert test is true.

- **Deactivate Schedule After Alert.** This option deactivates the schedule after the Alert is triggered. The schedule has to be activated again for the Alert test to run after the Alert is triggered. It is best to use this option for one-time Alert tests. This is the default value.

- **Delay.** This option is best used when you want to allow a specific period of time to address the Alert test but want to be notified again if the Alert test has not been met. You can specify to restart the Alert after a maximum of 99 hours, days, weeks, months, or years.
As you complete the creation of the schedule, remember to consider the burst option to distribute specific report values, instead of the entire report. When a report is burst, the distribution information must be provided in a distribution list, distribution file, or dynamic list for email distribution and an access list for library distribution. The email and library distribution options are the most effective for Alert reports because of the push versus pull notification through email. For more information on using the burst option in ReportCaster, see the ReportCaster manual.

23. Click Distribution and type the email addresses you want the report sent to.

24. Click Properties in the Show group and type a title and summary for your scheduled report.

25. Click Recurrence and specify when and how often to run the schedule.

26. Click Save & Close.

The scheduled report now appears in the tree, as shown in the following image.

![Image showing the tree view of scheduled reports]

Checking Scheduled Alerts

When an Alert test is false (and therefore not activated), the Alert result report is not run. When an Alert schedule runs, information about the Alert test evaluation (true or false) and any error or warning messages that occur during schedule or Alert processing are written to the log file.

In order to track the completion of a scheduled Alert, or errors that occurred during processing without checking the log report for the Alert schedule, we recommend that you use the Schedule Notification option.
In the Scheduling tool, the Notification tab allows you to specify:

- The terms for supplying notification (Never, which is the default value, Always, or On Error).
- The level of detail in the notification (full notification or brief notification).
- The email addresses to distribute the notification information to.
Checking Scheduled Alerts
With WebFOCUS Managed Reporting, a developer can easily transform complex views of data into simple objects labeled with common business terminology that every user can understand, such as Weekly Sales or Revenue. Developers create Reporting Objects that present available data using terms and formats meaningful to users. Users can then use Reporting Objects as templates for the creation of a wide range of reports and graphs.

**In this chapter:**
- Reporting Objects
- Components of a Reporting Object
- Running a Reporting Object From Developer Studio
- Creating a Reporting Object Which Uses a HOLD File

---

**Reporting Objects**

Reporting Objects are stored in folders in the Repository at any level except the root. They can include selection criteria (WHEREs), JOINs, virtual fields (DEFINEs), filters, and other WebFOCUS statements, as well as reports or graphs supplied as templates.

**Procedure: How to Create a Reporting Object**

1. Launch Developer Studio and navigate to the Repository.
2. If the folder which will contain the Reporting Object already exists, proceed to step 4. Otherwise, create the folder by right-clicking the desired location in the Repository and selecting New, then Folder, or right-click Repository and select New Folder to create the folder directly under the root.
The New Folder dialog box appears, as shown in the following image.

3. Type the name of the Reporting Objects folder (the name automatically appears in the Title field), optionally select a server and an application path, and click Create.

   **Note:** If you do not select a server or an application path, each will be set to the default server and application paths for your Developer Studio client. For more information, see the Managed Reporting manual.

4. Right-click the folder which will contain the Reporting Object and select New, then Reporting Object.
The New Reporting Object dialog box appears, as shown in the following image.

![New Reporting Object Dialog Box](image)

The Look in location is set to the folder you selected, although you can change it to another location.

5. Type the name of the new Reporting Object in the File name field and click File to select the Master File on which your Reporting Object will be based.
The WebFOCUS Table List dialog box appears, as shown in the following image.

![WebFOCUS Table List](image)

**Note:** Depending on your environment, you may be prompted for WebFOCUS Server credentials when you click the *File* button.

6. Select a table and click *OK* or, if you would like to create a new table, click *Create New*.

7. You may also make the following optional selections when creating a new Reporting Object:

   - **Prompt for Parameters** determines whether or not users will be prompted for parameters (values) when they create private reports from this object. This is selected by default.
   - **Schedule Only** determines whether or not the Reporting Object runs as a scheduled report only. By default, this is not available.
   - **Run With OLAP** determines whether or not the Reporting Object is OLAP-enabled. By default, this is not selected.
   - **Only allow user to Run** determines whether or not only a specific user can run the Reporting Object. By default, this is not selected.
   - **Allow user to Run** determines whether or not running the Reporting Object is restricted to a specific user. By default, this is not selected.
   - **Do not show on User’s List** determines whether or not the Reporting Object is restricted from appearing in user lists. By default, this is not available.
- Only Run as Deferred Report sets the report to run as a deferred report. By default, this is not selected.

The Data Server tab allows you to manage the data server and the application path settings from which the Reporting Object retrieves information.

8. When you have selected all the desired options, click Create.

The Reporting Object tool appears, as shown in the following image.

![Reporting Object Tool](image)

Note: Depending on your environment, you may be prompted for WebFOCUS Server credentials when you select the Create button.

9. Double-click a Reporting Object component to begin customizing the object.

Procedure: How to Set a Different Data Server for a Reporting Object

By default, the data server, which defines the Reporting Object, is the same data server from which it is accessed by users. However, you may wish to create a Reporting Object, which draws upon limited information from another data source, for example, to allow Sales managers to combine inventory information with personnel information to determine personnel performance.
To set a different data server for a Reporting Object:

1. While creating a new Reporting Object, select the Data Server tab, as shown in the following image.

If the Reporting Object already exists, right-click the object and select Properties, then click the Data Server tab on the dialog box that appears.
2. To change the application path but keep the current server, select *Application Path*. Move applications from the Available box to the Selected box by double-clicking them or by selecting them and using the movement arrows. Use the up and down arrows to change the preferred order of the application paths.

3. To change the server, check *Server* and select from the available servers listed. Follow the directions above to set the application path of the new server.

**Properties of a Reporting Object**

To view or edit the properties of a Reporting Object, right-click the object and select *Properties*. The Reporting Object Properties dialog box opens, as shown in the following image.

![New_Reporting_Object.fex Properties](image)

- **Name:** New_Reporting_Object
- **Title:** New_Reporting_Object
- **Parent Folder:** IBFS:/localhost/WFC/Repository/Public/ReportingObjects
- **Full Path:** IBFS:/localhost/WFC/Repository/Public/ReportingObjects/New_Reporting_Object.fex
- **Size:** 300 bytes
- **Status:** Private/User: admin
- **Run:** immediately
- **Created On:** Thursday, October 18, 2012, 06:08:03 PM
- **Created By:** admin
- **Last Modified On:** Thursday, October 18, 2012, 06:08:03 PM
- **Last Modified By:** admin
- **Last Accessed On:** Thursday, October 25, 2012, 05:05:59 PM
- **Last Accessed By:** admin
- **Prompt for Parameters**
- **Run With OLAP**
- **Allow user to Run**
- **Only Run as Deferred Report**

[OK] [Cancel] [Apply] [Help]
In the Properties dialog box, the General tab displays the parent folder (location), file name, size, status, and run option of the object, along with the user and date of the creation, last access, and last modification of the object. You can also edit the object description and name and enable the following options:

- **Prompt for Parameters** determines whether or not users will be prompted for parameters (values) when they create reports or graphs. This is selected by default.

  If prompting is enabled globally, then, even when this option is not selected, you will still be prompted for any unresolved variables. These unresolved variables can be resolved by using the defaults or by passing them on.

  **Note:** If you create a graph with an optional parameter, you must clear the Prompt for Parameters option. If you do not clear the Prompt for Parameters option, you will be prompted, at run time, to select the optional parameter.

- **Schedule Only** determines whether or not the Reporting Object runs as a scheduled report only. By default, this is not available.

- **Run With OLAP** determines whether or not the Reporting Object is OLAP-enabled. By default, this is not available.

- **Only allow user to Run** determines whether or not only a specific user can run the Reporting Object. By default, this is not available.

- **Allow user to Run** determines whether or not running the Reporting Object is restricted to a specific user. By default, this is not available.

- **Do not show on User’s List** determines whether or not the Reporting Object is restricted from appearing in user lists. By default, this is not available.

- **Only Run as Deferred Report** sets the report to run as a deferred report.
The Data Server tab allows you to manage the data server and the application path settings from which the Reporting Object retrieves information. The Data Server tab is shown in the following image.

To change the server, check **Server** and select one of the available servers. Move application paths from the Available box to the Selected box by double-clicking them or by selecting them and using the movement arrows. Use the up and down arrows to change the preferred order of the application paths.
Components of a Reporting Object

The Summary tab, as shown in the following image, allows you to enter a brief description of the Reporting Object.

### Components of a Reporting Object

A Reporting Object may contain any of the following types of components, which are executed in order of precedence as listed:

- Preprocessing Other (custom code that runs before all other components)
Joins
 Defines
 Filters, when applicable
 Where statements
 Report or graph templates
 Postprocessing Other (custom code that runs after all other components)

**Note:** Filters are only applied if the user of the Reporting Object activates them when the report or graph based on the Reporting Object is executed.

Users do not see the WHERE statements, DEFINE statements, or joins you create, which are issued automatically every time a user accesses the Reporting Object. You can, however, provide report or graph templates for users, including headers, footers, and styling features, which users can see and customize for their own needs.

To view or edit a Reporting Object, double-click it, or right-click it and select *Edit in Reporting Object tool*. The Reporting Object tool displays, as shown in the following image.

![Reporting Object Tool](image)

**Preprocessing Other Component**

The Preprocessing Other component contains any custom code that must run before all other components. You can use the text editor in the component to create and edit code, or to paste the code you have copied from another application.

To create or edit a Preprocessing Other component, right-click the component and select *Edit*. The text editor appears.
See *Editing Application Components as Text in Developer Studio* on page 369 for more information on using the test editor. For more information about using the WebFOCUS language to code procedures, see the *Creating Reports With WebFOCUS Language* manual.

**Note:**

- Double-clicking the component or right-clicking and selecting *Open* will open the procedure in the Procedure Viewer instead of the text editor.
- Syntax and error checking are not performed on the commands you enter until the Reporting Object is executed.
- Commands in the Preprocessing Other component should not return a displayable answer set.
- InfoAssist previews in the Report component and the Graph component are disabled if the Preprocessing Other component contains the syntax `-OLAP ON`

To avoid this issue, enable OLAP through the Reporting Object properties instead of the Preprocessing Other component. For more information on the Reporting Object properties, see *Properties of a Reporting Object* on page 409.

**Joins**

A join is a temporary connection between two or more data sources that share at least one common field. After you join two data sources, each time WebFOCUS retrieves a record from the first data source (host file), it also retrieves the matching records from the second data source (target file).

You use the Join tool to link data sources. The Join tool provides a graphical method for creating and manipulating joins. Instead of coding a join using the WebFOCUS language, you use the Join tool to drag a field from one data source and link it to a field in another data source.

When you access the Join tool, the window displays a field list for the data source associated with the Reporting Object (which you specified when you created the Reporting Object) and a field list for any data source you add to the window. An icon representing the data type appears before each field name in the Master File.

**Note:** When joining two FOCUS data sources, the cross-referenced field of the target file must be indexed.
Since every procedure creates a new session on the Reporting Server when it is run, the duration of your connection is always limited to a single procedure. Therefore, any Join issued at the beginning of a procedure will be in effect only for those report requests that are called in the same procedure.

To open the Join component in the Join tool, double-click the component or right-click and select Open. To view the Join as a text file, right-click the component and select Edit instead.

For a full description of joining data sources using the browser, see the Creating Reports With WebFOCUS Language manual. For a full description of joining data sources using Developer Studio, see Joining and Merging Data Sources on page 141.

**Virtual Fields (DEFINE Statements)**

A virtual field is a field whose value is not stored in the data source but can be calculated from the data that is there. A virtual field takes up no storage space. It is created only when a user accesses the Reporting Object that includes the virtual field.

You use the Define tool to create a virtual field. Using this tool, you specify the virtual field name, format, and the expression that defines the field. An expression enables you to combine fields, constants, predefined functions, and operators into an operation that produces a single value.

When you define a virtual field, you must specify a field format type, length, and appropriate display options.

To open the Define component in the Define tool, double-click the Define component or right-click the component and select Open. To view the Define statement as a text file, right-click the component and select Edit instead.

For more information about defining fields, and a full description of the Define tool, see Creating Temporary Fields on page 273.

**Filters**

Filters are selection criteria (WHERE statements) which enable users to quickly select predefined criteria that limit the data included in a report or graph. You create the filters for users, who then select the filters for application without having to create their own. Filters differ from other selection criteria in that filters run only when the user selects them at run time. If the user turns off the filter, the data previously excluded from the display is again available.

In order to create a filter, you must create a filter group and define the filters that make up the group.
Procedure: How to Create a Filter Group

1. Double-click the Reporting Object to which you want to add the filter group.
2. Right-click the Filters folder and select New Group.
   The New Filter Group dialog box opens.
3. In the Name field, type a descriptive name for the filter group folder.
   You should make the filter group name as explicit as possible, because your users depend on this name to select the correct filter to apply to their reports.
4. Click OK.
   You return to the Reporting Object tool. The filter group folder appears below the Filter component folder.

Procedure: How to Define a Filter

1. Right-click the filter group folder to which you want to add a filter and select New Filter.
   The New Filter dialog box opens.
2. In the Name field, type a descriptive name for the filter and click OK.
   You should make the filter names as explicit as possible, because users depend on these names to select the correct filter to apply to their reports.
   The Basic Expression Builder dialog box opens.
3. Use the Basic Expression Builder to build the filter.
   For more information on using the Basic Expression Builder, see Using the Expression Builder in the Creating Reports With Report Painter manual.
4. When you have created the filter, click OK.
   WebFOCUS saves the filter and returns you to the Reporting Object tool. The new filter appears below the filter group folder.

Note:

- To open the filter in the text editor instead of the Basic Expression Builder, right-click the Filter icon and select Edit.
- Syntax and error checking are not performed on the commands you enter until the Reporting Object is executed.
For details on the Basic Expression Builder, see *Using the Expression Builder* in the *Creating Reports With Report Painter* manual.

**Procedure:** How to Rename a Filter or Filter Group

1. Right-click the filter or filter group and select *Rename*.
2. Type the new name in place, then press Enter or click the mouse outside the field to finish renaming.

**Reference:** Filters Considerations With Amper Auto Prompting

When using Amper Auto prompting, users may be prompted to supply amper variable values in filters they did not select, or not prompted to supply amper variable values in filters they did select. For more details, see *Considerations for Amper Auto Prompting With the Preprocessing and Postprocessing Other Components* on page 421.

**Selection Criteria (WHERE Statements)**

You limit the data a user can report from by defining selection criteria, which restrict data to only the records that meet conditions you specify. You use the Basic Expression Builder to define selection criteria.

Selection criteria, for example, filters, limit the amount of data a user can access to build a report. When a user opens a Reporting Object that contains selection criteria, the selection criteria runs and excludes those data values that do not meet the criteria. A filter, on the other hand, runs only when a user selects the filter at run time.

**Procedure:** How to Create Selection Criteria

1. Double-click the Reporting Object to which you want to add the selection criteria.
2. Double-click the *Where* icon, or right-click the icon and select *Open*.
   
   The Basic Expression Builder appears.
3. Use the Basic Expression Builder to build the selection criteria.
   
   For details on the Basic Expression Builder, see *Using the Expression Builder* in the *Creating Reports With Report Painter* manual.
4. Click *OK* to save the selection criteria and return to the Reporting Object tool.
Components of a Reporting Object

**Note:**

- To open the filter in the text editor instead of the Basic Expression Builder, right-click the Filter icon and select *Edit*.
- Syntax and error checking are not performed on the commands you enter until the Reporting Object is executed.

**Report Component**

You use the Report component to create a report template and save it in a Reporting Object. When a user selects this Reporting Object in the browser, the report template you created is displayed. Users can modify the template to create a report that suits their needs and then save the new report.

Double-click the Report component or right-click the component and select *Open*. InfoAssist opens, allowing you to create a report template.

For more information on using InfoAssist, see the *WebFOCUS InfoAssist User’s Manual*. 
13. Reporting Objects in Developer Studio

Note:

- To open the report in the text editor instead of InfoAssist, right-click the Report component and select Edit. Manual changes to the code may make the report incompatible with InfoAssist.

- Syntax and error checking are not performed on the commands you enter until the Reporting Object is executed.

- If the Reporting Object contains both a report and a graph, the user is asked to select whether to run the report or the graph at run time. The Reporting Object cannot run both simultaneously.

- When using Amper Auto prompting (the Reporting Object properties have Prompt for Parameters selected), there are scenarios in which users are prompted to provide amper variable values in selection criteria (WHERE) statements in the Reporting Object that the user did not save with their report. This is due to internal processing of private reports created from Reporting Objects. You can avoid this issue by using the Reporting Object Filter component to predefine selection criteria that users can select to include in their reports. Only the Filter(s) that users select for inclusion in their reports are evaluated by the WebFOCUS Amper Auto prompting facility.

- When you exit InfoAssist from Developer Studio, InfoAssist displays a prompt asking you to save your changes. This prompt does not specify the name of the Reporting Object. If you click Yes, changes are passed on to Developer Studio, but are not saved until you save the Reporting Object itself in Developer Studio.

Graph Component

The Graph component works exactly like the Report component. You use the Graph component to create a graph template and save it in a Reporting Object. When a user selects this Reporting Object in the browser, the graph template you created is displayed. Users can modify the template to create a graph that suits their needs and then save the new graph.

Double-click the Graph component or right-click the component and select Open. InfoAssist opens, allowing you to create a graph template.

For more information on using InfoAssist, see the WebFOCUS InfoAssist User’s Manual.
Components of a Reporting Object

**Note:**

- To open the graph in the text editor instead of InfoAssist, right-click the Graph component and select *Edit*. Manual changes to the code may make the graph incompatible with InfoAssist.

- Syntax and error checking are not performed on the commands you enter until the Reporting Object is executed.

- If the Reporting Object contains both a report and a graph, the user is asked to select whether to run the report or the graph at run time. The Reporting Object cannot run both simultaneously.

- When using Amper Auto prompting (the Reporting Object properties have *Prompt for Parameters* selected), users may be prompted to provide amper variable values in selection criteria (WHERE) statements in the Reporting Object that the user did not save with their graph. This is due to internal processing of private graphs created from Reporting Objects. You can avoid this issue by using the Reporting Object Filter component to predefined selection criteria that users can select to include in their graphs. Only the Filter(s) that users select for inclusion in their graphs are evaluated by the WebFOCUS Amper Auto prompting facility.

- When you exit InfoAssist from Developer Studio, InfoAssist displays a prompt asking you to save your changes. This prompt does not specify the name of the Reporting Object. If you click Yes, changes are passed on to Developer Studio, but are not saved until you save the Reporting Object itself in Developer Studio.

**Postprocessing Other Component**

The Postprocessing Other component contains any custom code that must run before all other components. You can use the text editor in the component to create and edit code, or to paste the code you have copied from another application.

To create or edit a Postprocessing Other component, right-click the component and select *Edit*. The text editor appears.

See *Editing Application Components as Text in Developer Studio* on page 369 for more information on using the test editor. For more information about using the WebFOCUS language to code procedures, see the *Creating Reports With WebFOCUS Language* manual.
**Note:**

- Double-clicking the component or right-clicking and selecting *Open* will open the component in the Procedure Viewer instead of the text editor.
- Syntax and error checking are not performed on the commands you enter until the Reporting Object is executed.
- Commands in the Postprocessing Other component should not return a displayable answer set.
- InfoAssist previews in the Report component and the Graph component are disabled if the Postprocessing Other component contains the syntax:

  ```
  -OLAP  ON
  ```

  To avoid this issue, enable OLAP through the Reporting Object properties instead of the Postprocessing Other component. For more information on the Reporting Object properties, see *Properties of a Reporting Object* on page 409.

**Reference:** Considerations for Amper Auto Prompting With the Preprocessing and Postprocessing Other Components

When using Amper Auto prompting, users may be prompted to supply amper variable values for parameters that the private user content does not reference. These scenarios are directly related to the internal processing of private content based on Reporting Objects.

Auto prompting considerations and expected behavior must be understood when referencing amper variables within a Reporting Object. For information on the Amper Auto prompting facility, see the *Developing Reporting Applications* manual. The specification of values for the amper variables in the Reporting Object Other component using –DEFAULT, –DEFAULTH, or –SET commands will assign a default value for amper variables in a report request to avoid a FOC error for not specifying a value. A consideration for –DEFAULTH and –SET is that Amper Auto prompting does not prompt for amper variables that have values assigned using –DEFAULTH and –SET.

Whether a user is prompted by the Amper Auto prompting facility for amper variables assigned a default value using the –DEFAULT command is dependent on the setting of the WebFOCUS Client configuration parameter, IBIMR_prompting. When IBIMR_prompting is set to XMLPROMPT, users are prompted for amper variable values and defaults will be displayed in fields where the default values have been specified. When IBIMR_prompting is set to XMLEXCEL, users are not prompted for variables when all amper variables have been assigned a default value. For more information on setting the IBIMR_prompting parameter, see the *WebFOCUS Security and Administration* manual.
Running a Reporting Object From Developer Studio

You can run a Reporting Object or any individual component of it in Developer Studio. To run a Reporting Object, select the object and click the Run button or the Run Deferred button from the Developer Studio toolbar, or right-click the object and select Run. The Run option in the Reporting Object tool is enabled regardless of the setting of the Reporting Object Only Run as a Deferred Report property.

Expected Behaviors of Running a Reporting Object

When you select and run a Reporting Object, the expected behavior is as follows:

- When you define a Report or Graph component, all components of the Reporting Object run.
- When you define a Report and Graph component, an HTML page appears that enables you to select which one to run.
- When you define filters in a Reporting Object, an HTML page appears that enables you to select from the defined filters of the Reporting Object.
- When you define parameters in a report or graph and the Reporting Object Prompt for Parameters property is specified, an HTML page appears that prompts you for parameter values.
- Reporting Object components are validated prior to run time. This means that errors are found and corrected more easily since errors in individual components are detected prior to run time.
- If you attempt to run a Reporting Object that is created without using the Report or Graph components, you will receive the message: 
  
  EDA no data

Creating a Reporting Object Which Uses a HOLD File

Suppose your company has a data warehouse and you need a Reporting Object that uses an extract from that data warehouse. In your Preprocessing Other component, you create a HOLD file from that data warehouse, and then you use that HOLD file in each Reporting Object component. You delete the HOLD file in the Postprocessing Other component so it is not mistakenly used again the next time the Reporting Object is used.

This task has the following steps:

1. Create a HOLD file in the Preprocessing Other component.
2. Create a new Join component with two Joins.
3. Use the Report component to create a new report in InfoAssist.
4. Delete the Join in the Postprocessing Other component.
5. Run the Reporting Object, then view the object source to verify that the output is correct.

**Example:** Creating a HOLD File With the Preprocessing Other Component

1. Create a Reporting Object with the BASEAPP and IBISAMP selected as the application paths and the EMPLOYEE table selected as the MASTERFILE.
   
   For more information on creating a Reporting Object, see *How to Create a Reporting Object* on page 403.

2. Right-click the Preprocessing Other component and select *Edit*.

3. Type the following code:

   ```plaintext
   APP HOLD IBISAMP
   TABLE FILE EMPLOYEE
   PRINT EMPLOYEE.EMPINFO.DEPARTMENT AS 'NEWDEP'
   EMPLOYEE.EMPINFO.CURR_SAL AS 'NEWCURR_SAL'
   BY EMPLOYEE.EMPINFO.EMP_ID AS 'EMP_ID'
   BY EMPLOYEE.EMPINFO.HIRE_DATE AS 'NEWHIRE_DATE'
   ON TABLE HOLD AS WINFILE FORMAT FOCUS INDEX EMP_ID
   END
   ```

4. Click the *Save* button, then close the text editor.

5. To verify that the component works properly, select the Preprocessing Other component and click the *Run* button.

   The following output is produced:

   ```plaintext
   0 NUMBER OF RECORDS IN TABLE= 12 LINES= 12
   ```

**Example:** Creating a New Join Component With Two Joins

1. Double-click the Join component to open it.

   The Join component displays the EMPLOYEE table.

2. Click the *Add File* button, then select the WINFILE table and click *OK*.

3. Click the *Add File* button, then select the JOBFILE table and click *OK*.

4. Click the *Run Join* button in the toolbar to verify the Join.
The following output is produced in the Check dialog box:

```
SET XRETRIEVAL=OFF
APP HOLD IBISAMP
TABLE FILE EMPLOYEE
PRINT EMPLOYEE.EMPINFO.DEPARTMENT AS 'NEWDEP'
EMPLOYEE.EMPINFO.CURR_SAL AS 'NEWCURR_SAL'
BY EMPLOYEE.EMPINFO.EMP_ID AS 'EMP_ID'
BY EMPLOYEE.EMPINFO.HIRE_DATE AS 'NEWHIRE_DATE'
ON TABLE HOLD AS WINFILE FORMAT FOCUS INDEX EMP_ID
END
SET XRETRIEVAL=ON
JOIN
EMPLOYEE.EMPINFO.EMP_ID IN EMPLOYEE TO MULTIPLE WINFILE.SEG01.EMP_ID IN
WINFILE
TAG J1 AS J1
END
JOIN
EMPLOYEE.PAYINFO.JOBCODE IN EMPLOYEE TO MULTIPLE JOBFILE.JOBSEG.JOBCODE
IN JOBFILE TAG J2 AS J2
END
```

5. Click OK to close the Check dialog box.

6. Close the component, clicking Yes to save the current changes.

**Example:** Creating a Report With the Report Component

1. Double-click the Report component.
   
   InfoAssist launches.

2. In the Data pane, from the Measures/Properties segment, drag and drop the NEWCURR_SAL field onto the canvas.

3. In the Data pane, from the Dimensions segment, drag and drop the EMP_ID field onto the canvas.

4. In the Data pane, from the Dimensions segment, drag and drop the NEWHIRE_DATE field onto the canvas.

5. In the Data pane, from the Dimensions segment, drag and drop the NEWDEP field onto the canvas.

   The report displays as a table with four columns.

6. Close the component, saving your changes.

7. Save the changes to the Reporting Object.

8. To verify that the component works properly, right-click the Reporting Object and select Run.
The report you created in InfoAssist appears in the browser.

**Example:** Deleting the Join in the Postprocessing Other Component

1. Right-click the Postprocessing Other component and select *Edit*.

2. Type the following:

   ```
   ? JOIN
   JOIN CLEAR *
   ? JOIN
   ```

3. Close the component, saving your changes.

4. Save the changes to the Reporting Object.

5. To verify that the Reporting Object works properly, select the object and click the *Run* button.

   The report you created in InfoAssist appears in the browser.

6. Right-click the report webpage and select *View source*.

7. Scroll to the bottom of the source window.

   The following comment displays:

   ```
   <!--
   0 NUMBER OF RECORDS IN TABLE= 12 LINES= 12
   0 NUMBER OF RECORDS IN TABLE= 12 LINES= 12
   0 HOLDING HTML FILE ON PC DISK ...
   JOINS CURRENTLY ACTIVE
   HOST                              CROSSREFERENCE
   FIELD        FILE      TAG     FIELD     FILE      TAG AS ALL WH
   -----        ----      ---     -----     ----      --- -- --- --
   EMPLOYEE.EM>EMPLOYEE        WINFILE.SEG> WINFILE  J0 J0 Y N
   EMPLOYEE.PA>EMPLOYEE        JOBFILE.JOB> JOBFILE  J1 J1 Y N
   0 NO JOINS CURRENTLY IN EFFECT
   -->
   ```
<p>| <strong>Glossary</strong> |
|-----------------|---------------------------------------------------|
| <strong>.fex</strong>         | The file extension for a procedure.               |
| <strong>Access File</strong> | Contains information needed by WebFOCUS to access the data in a data source. The Access File includes the real name and location of the data source. |
| <strong>Alert result</strong>| A report procedure that is triggered by an Alert test if a set of defined conditions (rules) is met (true). |
| <strong>Alert test</strong>  | A test that checks whether or not a set of defined conditions (rules) is met (true or false). If the test is triggered and the set of conditions has been met (true), it generates an Alert result. |
| <strong>Alert tool</strong>  | A tool you can use to set up a test condition that determines if an alert should be triggered. This helps to determine when and under what conditions a report should be submitted. For example, an alert would be triggered when an order exceeds $1 million. |
| <strong>APPROOT</strong>     | A variable in the directory path on the WebFOCUS Reporting Server. The default setting for APPROOT is <code>install_drive:\ibi\apps</code>. |
| <strong>Basic Expression Builder</strong> | A tool you can use to create expressions quickly, by selecting fields, relations, operators, and values from lists. You can base selection criteria on a specified value, a variable value, or a field value. You can access the Basic Expression Builder either by clicking the Assist button within the WHEN dialog box in Report Painter, or from a WHERE-based Join. |
| <strong>calculated value</strong> | A temporary field that is calculated after all records have been selected, sorted, and aggregated. |
| <strong>component</strong> | A piece of a procedure, such as a report, graph, or Join. Components are stored within a .fex file. |
| <strong>Create Synonym tool</strong> | A tool you can use to create synonyms for remote data sources configured with the WebFOCUS Reporting Server, or for tables that reside on a subserver. |
| <strong>Creation tool</strong> | A tool you can use to create a procedure, such as a report, graph, Join, SQL report, or DEFINE function. Examples of a creation tool include Report Painter, Composer, SQL Report Wizard, and Procedure Viewer. |
| <strong>cube</strong> | A multi-dimensional data source. |
| <strong>Define tool</strong> | A tool that you use to create virtual fields as a component in a reporting procedure. Define the virtual field by assigning a format and typing an expression, or by composing it using the calculator and the fields and functions listed in the tabbed panes in the Define tool window. |
| <strong>delimited flat file</strong> | A data file that you can upload for use in the WebFOCUS reporting tools, such as a .txt, .csv, .ftm, .dat, or .tmp file. |
| <strong>deployment</strong> | The process where Developer Studio takes partitioned files and moves them to the target servers, where they make up an application that users run on the web. |
| <strong>Engine tool</strong> | A tool you can use to create ENGINE statements. These statements control connection attributes and override parameters. |
| <strong>Expression Builder</strong> | A tool you can use to create expressions quickly, by selecting fields, relations, operators, and values from lists. You can base selection criteria on a specified value, a variable value, or a field value. You can access the Expression Builder by clicking Where, If, or Where Total from the Where/If drop-down menu. |</p>
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>A selection criterion, or WHERE statement, that you can use to quickly select predefined criteria that limits the data included in a report or a graph.</td>
</tr>
<tr>
<td>graph</td>
<td>A graphical representation of data within a Master File. You can use the Graph tool to create graphs, and these graphs can be inserted in the page layout.</td>
</tr>
<tr>
<td>hierarchy</td>
<td>A parent/child relationship between fields. The relationship is defined in a Master File.</td>
</tr>
<tr>
<td>HOLD file</td>
<td>A type of output file. The output of a report request stored in a file that you can use as input to another WebFOCUS procedure.</td>
</tr>
<tr>
<td>Information Builders File System (IBFS)</td>
<td>A logical addressing system used by WebFOCUS to store and retrieve objects.</td>
</tr>
<tr>
<td>Join</td>
<td>A temporary connection between two or more data source files that share at least one common field. Once files are joined, each time a record is retrieved from the first file (host file), it also retrieves the matching records from the second file (cross-referenced file).</td>
</tr>
<tr>
<td>Maintain File</td>
<td>Contains commands that can access, read, and manipulate the information in a data source.</td>
</tr>
<tr>
<td>Master File</td>
<td>Describes the data source from where you are reporting. The Master File is a map of the data source and all of the fields. By looking at the Master File, you can determine what fields are in the data source, what they are named, and how they are formatted. You can also determine how the fields in the data source relate to each other.</td>
</tr>
<tr>
<td><strong>output file</strong></td>
<td>Contains the output of a report request. You can capture the output multiple times in different formats. Each output file is classified as either Save or Hold.</td>
</tr>
<tr>
<td><strong>partitioning</strong></td>
<td>The process where 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. A partition is defined in a deployment scenario.</td>
</tr>
<tr>
<td><strong>procedure</strong></td>
<td>A series of components that you can execute one after another. For example, a procedure can consist of a Join, Set, and report.</td>
</tr>
<tr>
<td><strong>project</strong></td>
<td>A reporting application consisting of different kinds of files. This can be a stand-alone application in the development environment or a web-based application implemented through the deployment feature.</td>
</tr>
<tr>
<td><strong>report request</strong></td>
<td>A series of commands and phrases that defines the contents and format of a report.</td>
</tr>
<tr>
<td><strong>ReportCaster</strong></td>
<td>A WebFOCUS tool. ReportCaster is a scheduling and distribution application that centralizes the execution and distribution of WebFOCUS reports, the contents of URLs, and files. ReportCaster supports multiple administrators and provides a single point of control for managing the information required to run an organization.</td>
</tr>
<tr>
<td><strong>reporting application</strong></td>
<td>A self-contained program with an interface that you can use to accomplish reporting tasks.</td>
</tr>
<tr>
<td><strong>Reporting Object</strong></td>
<td>A tailored view of a set of data that a Managed Reporting Administrator creates and saves to a group folder. The data contained in a Reporting Object is used to create personal reports quickly and in compliance with the reporting rules and guidelines of your company.</td>
</tr>
<tr>
<td><strong>SAVE file</strong></td>
<td>A type of output file. The output of a request stored in a file that you can use as input to a variety of applications, programming languages, and other WebFOCUS facilities, depending on the format type you specify. The default Save format is ALPHA, an ASCII text file containing your data values as printable characters.</td>
</tr>
<tr>
<td><strong>source control</strong></td>
<td>Manages access to source code and keeps track of all code changes.</td>
</tr>
<tr>
<td><strong>synonym</strong></td>
<td>Defines a unique logical name (also known as an alias) for each web services operation. Synonyms are useful because they insulate client applications from changes to the location and identity of a request. You can move or rename a request without modifying the client applications that use it. You need to make only one change, redefining the request synonym on the server. They provide support for the extended metadata features of the server, such as virtual fields and security mechanisms. Creating a synonym generates a Master File and an Access File.</td>
</tr>
<tr>
<td><strong>temporary field</strong></td>
<td>A field whose value is not stored in the data source. You can calculate a temporary field from the data that is there, or assign it an absolute value. A temporary field takes up no storage space in the data source, and is created only when needed.</td>
</tr>
<tr>
<td><strong>Text Editor</strong></td>
<td>A tool you can use to view the code, launch graphical tools, and to create procedures and other types of application files.</td>
</tr>
<tr>
<td><strong>Update Assist</strong></td>
<td>A tool you can use to create applications that add records, update records, or delete records against any data source for which you have read/write access. To access Update Assist, open a project, right-click any Master File in the project path, and click Update Assist.</td>
</tr>
<tr>
<td><strong>variable</strong></td>
<td>Used to slice data. You can use a variable in data selection and in some cases, to populate a drop-down list box.</td>
</tr>
<tr>
<td><strong>virtual field</strong></td>
<td>A temporary field that is evaluated as each record that meets the selection criteria is retrieved from the data source.</td>
</tr>
<tr>
<td><strong>virtual folder</strong></td>
<td>A folder that applies a logical structure to a project, but does not actually exist as a physical directory. When you create a project, the project node populates with virtual folders labeled HTML files, Maintain Files, Master Files, Procedures, and Other.</td>
</tr>
<tr>
<td><strong>WebFOCUS Administration Console</strong></td>
<td>The interface that administrators use to manage the WebFOCUS environment and configuration settings.</td>
</tr>
<tr>
<td><strong>WHERE expression</strong></td>
<td>An expression that displays only those field values that meet your needs.</td>
</tr>
<tr>
<td><strong>WHERE Expression Builder</strong></td>
<td>A tool you can use to create a WHERE expression. The WHERE Expression Builder is only available in the Join tool and when two or more files are joined together. Creating a WHERE expression in the Join tool creates a conditional Join.</td>
</tr>
<tr>
<td><strong>Workspace file</strong></td>
<td>Contains projects that are stored on your PC. You can control and limit projects that are visible by using Workspace files in the Projects area of Developer Studio Explorer.</td>
</tr>
</tbody>
</table>
Index

-HTMLFORM command 115
deploying files 115
? DEFINE command 287
? JOIN command 156, 167
.GFA control file 27, 100
deleting 100

A

Access Files 44
creating synonyms 44
ACROSS phrase 291
adding/removing comments 373
Advanced Allocation dialog box 259
advanced fields 194
aggregated data 201
Alert Assistant 391
accessing 391
Alert Log 400
alert tests 386, 391
creating 391
external 391
running batch or script files 391
scheduling 391
Alert Wizard 391
accessing 391
creating alert tests 391
alerts 385, 386, 389, 391
components 386
events 386
reporting 385
scheduling 391
Alerts
tracking 400
aliases
APPROOT 126
creating 127
ALL parameter 145
join structures and 145
allocating files 347
Dynamic List validation in Update Assist 347
Allocation Wizard 241, 257, 259
fields and options 257
ALPHA 311, 314
format 311, 314
application components 369, 370, 377
editing 369
viewing as text 377
APPROOT
aliases 126
variable 123

B

bar graphs 301
associating with columns 301
horizontal 301
vertical 301
Basic Expression Builder 416, 417
selection criteria 417
batch files 391
BINARY format 314
bookmarks 381
text editor 381

C

calculated values 273, 276, 290, 291
Computes tab 290
creating 290
screening values 291
sorting reports 291
case sensitivity 378, 380
cgivars.wfs file 123
characteristic classification 190
clearing virtual fields 287
cluster joins 141, 157
creating 157
columns
applying graphs to 301

Creating Reporting Applications With Developer Studio 433
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMA format</td>
<td>311, 314</td>
</tr>
<tr>
<td>comma-delimited files</td>
<td>146</td>
</tr>
<tr>
<td>restrictions</td>
<td>146</td>
</tr>
<tr>
<td>commands</td>
<td></td>
</tr>
<tr>
<td>MATCH FILE</td>
<td>180</td>
</tr>
<tr>
<td>Comment component</td>
<td>229, 372</td>
</tr>
<tr>
<td>comments</td>
<td>53, 224, 372</td>
</tr>
<tr>
<td>Comment component</td>
<td>224</td>
</tr>
<tr>
<td>creating</td>
<td>372</td>
</tr>
<tr>
<td>Component Connector toolbar</td>
<td>224, 229, 260</td>
</tr>
<tr>
<td>adding objects to procedures</td>
<td>224</td>
</tr>
<tr>
<td>components</td>
<td>35, 224</td>
</tr>
<tr>
<td>COMPUTE command</td>
<td>290</td>
</tr>
<tr>
<td>Computes tab</td>
<td>290</td>
</tr>
<tr>
<td>COMT format</td>
<td>314</td>
</tr>
<tr>
<td>concatenated join structures</td>
<td>153, 155</td>
</tr>
<tr>
<td>conditional join structures</td>
<td>141, 143, 156, 157</td>
</tr>
<tr>
<td>configuring deployment servers</td>
<td>123</td>
</tr>
<tr>
<td>creating join structures</td>
<td>166, 167</td>
</tr>
<tr>
<td>creating virtual fields</td>
<td>276, 277, 278, 279, 286</td>
</tr>
<tr>
<td>alphanumeric format</td>
<td>278</td>
</tr>
<tr>
<td>Define tool</td>
<td>278</td>
</tr>
<tr>
<td>numeric format</td>
<td>279</td>
</tr>
<tr>
<td>cross-referenced fields</td>
<td>146</td>
</tr>
<tr>
<td>cross-referenced files</td>
<td>141</td>
</tr>
<tr>
<td>cubes</td>
<td>189</td>
</tr>
<tr>
<td>customizing reports</td>
<td>139</td>
</tr>
<tr>
<td>data descriptions</td>
<td>129</td>
</tr>
<tr>
<td>data source types</td>
<td>129</td>
</tr>
<tr>
<td>data sources</td>
<td>132, 141, 146, 179, 415</td>
</tr>
<tr>
<td>joining</td>
<td>132, 141, 146, 415</td>
</tr>
<tr>
<td>merging</td>
<td>179</td>
</tr>
<tr>
<td>data types</td>
<td>292</td>
</tr>
<tr>
<td>data visualization</td>
<td>301</td>
</tr>
<tr>
<td>Date and Time Formats dialog box</td>
<td>279</td>
</tr>
<tr>
<td>date formats</td>
<td></td>
</tr>
<tr>
<td>unsupported</td>
<td>334</td>
</tr>
<tr>
<td>date separators</td>
<td>335</td>
</tr>
<tr>
<td>date-stamping fields in Update Assist applications</td>
<td>356</td>
</tr>
<tr>
<td>DB2 format</td>
<td>314</td>
</tr>
<tr>
<td>DBASE format</td>
<td>314</td>
</tr>
<tr>
<td>ddnames</td>
<td>241</td>
</tr>
<tr>
<td>default partition environments</td>
<td>108, 115</td>
</tr>
<tr>
<td>DEFINE and dates</td>
<td>288</td>
</tr>
<tr>
<td>DEFINE command</td>
<td>144, 278</td>
</tr>
<tr>
<td>join structures</td>
<td>144</td>
</tr>
<tr>
<td>Define dialog box</td>
<td>281</td>
</tr>
<tr>
<td>Define Field button in the Join component</td>
<td>168</td>
</tr>
<tr>
<td>DEFINE Function</td>
<td>292, 293, 294</td>
</tr>
<tr>
<td>command</td>
<td>292</td>
</tr>
<tr>
<td>creating temporary fields</td>
<td>292</td>
</tr>
<tr>
<td>displaying</td>
<td>294</td>
</tr>
<tr>
<td>limitations</td>
<td>293</td>
</tr>
<tr>
<td>DEFINE FUNCTION command</td>
<td>292</td>
</tr>
<tr>
<td>Define tool</td>
<td>278, 279, 281, 283, 284, 285, 286, 403, 415</td>
</tr>
<tr>
<td>assigning alphanumeric format</td>
<td>278</td>
</tr>
<tr>
<td>assigning date format</td>
<td>279</td>
</tr>
<tr>
<td>assigning numeric format</td>
<td>279</td>
</tr>
<tr>
<td>establish missing segment location</td>
<td>285</td>
</tr>
<tr>
<td>missing value attributes</td>
<td>284</td>
</tr>
<tr>
<td>Reporting Objects and</td>
<td>403</td>
</tr>
<tr>
<td>virtual fields and</td>
<td>415</td>
</tr>
<tr>
<td>defining files</td>
<td>241</td>
</tr>
<tr>
<td>Windows</td>
<td>241</td>
</tr>
<tr>
<td>deleting files from join structures</td>
<td>165</td>
</tr>
<tr>
<td>deleting join structures</td>
<td>165, 166</td>
</tr>
</tbody>
</table>
deployment
  guidelines 115
  scenario window 105
  scenarios 54, 103, 105
  servers 123
  status 119, 122
  steps 104
deployment paths 123, 126
  setting AS/400 126
Developer Studio text editor 369
DEVELOPERS variable 123
Development Tools 268
Dialogue Manager 270
DIF format 311, 314
dimension classification 190
directory paths 50
  modifying 50
display formats for reports 332
  EXCEL 332
  EXL2K 332
  EXL2K FORMULA 332
  EXL2K PIVOT 332
displaying reports 301
DOC format 311, 314
drill downs 297
drill-down reports
  using with Update Assist applications 358
dynamic join structures 141, 142
dynamically formatting virtual fields 288

E

edaserve.cfg file 124
embedded procedures 208
empty databases and Update Assist applications 354
ENGINE dialog box 269
Engine Tool 268
environments 24
  local development 24
  remote development 24
Essbase Adapter reporting rules 205
Excel 97 338, 339, 340
  Excel 2000 338
  Internet Explorer 338
  limitations 340
  reports in 339
EXCEL format 311, 314, 332
  display format 332
Excel Pivot tables 335
Execute component 259
Execute Wizard 260, 262
execution search paths 117
EXL07 format 332
EXL2K display format 332, 333, 335
  Internet Explorer and 333
  separators and 335
EXL2K format 314, 334, 335
  font support for 334
  PivotTable 335
EXL2K FORMULA display format 332
EXL2K PIVOT 314, 332
  display format 332
  format 314
EXL97 display format 338, 339
  reports in 339
expressions 273
external alert tests 391
extract files
  SAME_DB 326, 327
  temporary tables 326, 327

F

Field Creator dialog box 415
field lists 148, 152
Field Validation - File dialog box 347
Field Validation - List dialog box 346
Field Validation - Range dialog box 345
field-based reformatting 288
fields 146, 273, 342
  cross-referenced 146
  host 146
  joining 146
  selecting for Update Assist 342
fields list 190
file attributes 241
file types 32, 34, 51
  customizing 32, 34, 51
FILEDEF command 241
  Windows 241
files
  exporting 305
  generated by Update Assist 351
filters 33, 51, 190, 416
  grouping 416
  naming 416
Financial Report Painter 130
financial reports 130
flat hierarchies 194
FML Painter 130
FOCUS format 314
folders in Update Assist applications 355
footings 380
  adding with text editor 380
Format dialog box 279
formats for SAVE files 311
formatting reports as HTML 298, 299
free-form reports 130
FROLL prefix operator 201
full aggregation 201
FUSION format 314

generic procedures 219
graphical tools 382, 383
graphs 130

headings 380
  adding with text editor 380
hierarchical data 198, 199
  selecting records 199
  Show conditions 199
  sorting 198
  When conditions 199
hierarchical reporting 187
hierarchy 195, 198
  icons 195
  sort option 198
Hold dialog box 308
HOLD files 179, 306, 308, 326, 327
  creating 306, 308
  SAME_DB 326, 327
  temporary tables 326, 327
HOLD formats 314, 329, 330, 332
  ALPHA 314
  BINARY 314
  COMT 314
  DB2 314
  DBASE 314
  DIF 314
  DOC 314
  EXCEL 314
  EXL2K 314
  EXL2K format 332
  EXL2K PIVOT 314
  FOCUS 314
  FUSION 314
  HOLD formats
    COMMA 314
    SAVB 314
    HTML 314, 329, 330
    HTMLTABLE 314, 329
HOLD formats (continued)
PDF 314, 330
PS 330
SAVE 314
SQLDBC 314
SQLINF 314
SQLMSS 314
SQLDBC 314
SQLORA 314
SQLSYB 314
TABT 314
VISTISHOLD formats
WP 314
host fields 146, 153
concatenating 153
host files 141, 148, 415
HTML Composer
deploying files 115
Tree Control 208
HTML files 24
HTML Files folder 25
HTML format 311, 314, 329
HTML reports 298, 299, 330
viewing 330
HTMLTABLE format 314, 329
Join Define in File dialog box 168, 169
join field lists 148, 152
adding information 148, 152
Join Properties window 175
JOIN query command 156, 167
join structures 141, 143, 144, 145, 146, 147, 148, 150, 151, 153, 154, 155, 156, 165, 166, 167, 168, 170, 171, 172, 173, 177, 178, 415
ALL parameter and 145
clearing 172
copying 171, 173
creating 148, 150, 151, 153, 154, 155, 165, 166, 167
DEFINE command and 144
DEFINE-based 141
deleting 165, 166, 171
deleting files from 165
multiple instance 143, 144, 148
naming 156, 167
recursive 147, 170
restrictions 146
retrieving 171
running 172
saving 177, 178
single instance 143, 144, 148, 167
tag names and 170, 171
universal 146
virtual fields 167
virtual fields and 168
join syntax 171, 173
Join tool 148
Join toolbar 148, 149
Join window 148
Join Clear dialog box 171, 173
Key value navigation option for Update Assist 349

IBFS (Information Builders File System) 28
icons for multi-dimensional data sources 195
Include component 259
indexed fields 146
joining 146
Information Builders File System (IBFS) 28
inner join structures 145

J

Join Clear dialog box 171, 173
leaf nodes 194, 196
level hierarchy 187, 192
and the fields list 192
and the Object Inspector 192
listfill.fex file 347, 353
Local Deploy default value 54
Local Deploy scenario 105
logical names 241
Allocation Wizard 241
FILEDEF command and 241
Windows 241
Maintain Files folder 25
Maintain procedures 107
Managed Reporting Administrator 391
managing projects 64
mandatory variables 197
Master Files 24, 44, 45, 46, 129, 164, 306
adding to projects 44, 45
creating 306
creating synonyms 44
deleting permanently 46
new 164
removing from projects 46
Master Files folder 25, 46
upload data file 46
MATCH command 179
MATCH FILE commands 179, 180
Match Wizard 179, 180
maxFolders JavaScript variable 355
maxNodesPerFolder JavaScript variable 355
measures 190
member children cardinality 195
member level number 195
member name 195
member unique name 195
merging data sources 179, 180
missing segments 145
joining 145
mntSignon.htm file 353
multi-dimensional data sources 187, 189, 203
reporting rules 203
multi-field join structures 153, 154, 155
multiple instance join structures 143, 144, 148
multiple virtual fields 286
Add option 286
My Reports 138
naming join structures 167
Native Temporary Table 326
navbar.htm file 353, 355
customizing tree control 355
Navigation options in Update Assist 349
new databases and Update Assist applications 354
New Filter dialog box 63
New Procedure dialog box 221
New Published Content window 391
No key values navigation option for Update Assist 349, 358
using with WebFOCUS reports 358
Notification tab 400
numeric formats not supported in Excel 2000 334
Object Inspector 190, 335
Pivot tab 335
on-demand paging 297, 298, 300, 301
opening graphical tools 383
Other component 227, 369, 371, 372, 414, 420
text editor 371
Other folder 25
outer join structures 141, 145
output files 305, 307, 308, 310, 311, 314, 326, 327
creating 307, 308
formats 310, 311, 314
SAME_DB 326, 327
saving 308
temporary tables 326, 327
output files generated by Update Assist 351

P
parameters 190, 260, 262, 403
passing to procedures 260, 262
prompting for 403
parent level 195
parent unique name 195
parent/child hierarchy 188, 192, 194
and the fields list 192
and the Object Inspector 192
define fields 194
partial aggregation 201
partition project files window 121
partitioning project files 103, 112
passwords 100, 349
prompting for in Update Assist applications 349
PCHOLD file formats 321
PCHOLD formats 307, 314, 332
DOC 314
EXL2K 314, 332
HTML 314
HTMLTABLE 314
LOTUS 314
PDF 314
WP 314
PDF format 314, 330, 331, 332
compared to HTML 330, 332
compared to PostScript 330
drill downs 330, 331
software requirements 330
StyleSheet features 330, 331
PDIF 314
PivotTable 335
toolbar 335
positional column referenced calculated values 291
procedure components 224, 225, 229, 231, 232, 377, 382
checking 232
copying 232
creating 224, 225, 229
dragging and dropping 232
editing 231
running 231, 382
types 224
viewing as text 377
procedure creation tools 216, 217
Procedure Viewer
accessing the Match Wizard 179, 180
activating OLAP 270
adding to applications 222
adding to projects 222
calling 259
calling procedures 260, 262
canceling 272
components 224
copying 224
creating 215, 375
creating with text editor 220
debugging 270
decrypting 270, 271
deleting 223
editing 270, 271, 369, 370
embedding 260
encrypting 270, 271
including 260
passing parameters 260, 262
procedures (continued)
publishing 270
running 382
running locally 272
scheduling 270, 271
selecting tool for creating 217
text components 376
tools for creating 216, 217

Procedures folder 25
project directories 25
project files 103, 112, 119
deploying 103, 119
partitioning 112
project properties 49
modifying 49
Project Properties dialog box 49, 56, 112
Project Wizard 25
Project-based development environments 24
projects 25, 26, 31, 44, 49, 50, 64, 100, 112
adding Master Files 44
applying read-only security 50
creating directories 26
decrypting 100
deleting 100
encrypting 100
file types 25
managing 64
organizing 31
removing 100
securing 100
tracking code 64
viewing properties 49, 112
Projects area (continued)
saving workspace files 41
workspace files 36
workspace files properties dialog box 43
PS format 330

Q
qualified field names 171, 174
viewing 171, 174

R
Range validation option in Update Assist 345
read-only security 50
recursive join structures 147, 170
reformatting fields 288
registered file types 53
Report Assistant
creating alert tests 391
Report component 418, 419
report display formats 332
EXCEL 332
EXL2K 332
EXL2K FORMULA 332
EXL2K PIVOT 332
report output
saving 308
ReportCaster 271
scheduling procedures 271
ReportCaster Alerts
scheduling 400
tracking 400
reporting applications 23
Reporting Objects 403, 415, 418, 419, 422
folders 403
join structures 415
OLAP-enabled 403
running 422
selection criteria 403
templates 418, 419
reporting rules
  Essbase 205
  multi-dimensional data sources 203
  SAP BW 203
reporting tools 132
accessing 134
applying graphs to columns 301
calling Update Assist applications 358
columns 301
creating 129, 130, 136, 137
creating requests 132
customizing 132, 139
displaying 295, 296, 297, 298, 300, 301
displaying data 132
drilling down 297
editing as text 369, 370
export output 305
financial 130
free-form 130
output 132
requests 132, 134
running remotely 298
saving 138
selecting data 132
sorting data 132
specifying fields 132
SQL 233
templates 418, 419
types 130
viewing 330
Results page for Update Assist applications 349
running paths 117
running procedures 272
  locally 272
S
SAME DB 326
SAME_DB HOLD format 326, 327
SAP BW Adapter
  reporting rules 203
SAVB files 306
SAVB format 314
Save As/Export dialog box 309
SAVE files 306, 308, 309, 311
  creating 308, 309
  formats 311
SAVE formats 311, 314, 329
  ALPHA 311
  COMMA 311
  COMT 314
  DIF 311
  DOC 311
  EXCEL 311
  HTML 311, 329
  SAVE formats
text file 311
  WP 311
saving report output 308
Scenario Editor 105
screening data in hierarchies 199
screening values 291
script files 391
search paths 117
security 100, 349
  in Update Assist applications 349
SegmentName_validationInit.js file 353
SegmentName.mnt file 353, 356, 357
  autonumbering fields 357
date-stamping fields 356
selecting records in a hierarchy 199
selection criteria 403, 417
  creating 417
  defining for Reporting Objects 403
self-service reporting applications 23
sequential files 146
  joining cross-referenced fields 146
SESSION project 25
setting running paths 117
short paths 145
SHOW Builder 200
Show conditions 199
single instance join structures 143, 144, 148
Smart Deploy 105
Snowflake 159
source control 64, 65, 66, 69, 70, 74, 78, 87, 91, 94, 95, 97, 99
check files in and out 69
ClearCase 87
CVS 78
enabling 65
icons 70
managed reporting 66
Microsoft Visual SourceSafe 99
options 65, 97
accessing 65
requirements 64
SCC_Name Variable 95
set source control options 91
setting up 91
Subversion (SVN) 74
validating 94
view options 69
WebFOCUS Client 95
source file management 64
specifying fields 132
SQL 233
executing code 233
SQL Report Wizard 233
SQLDBC format 314
SQLINF format 314
SQLMSS format 314
SQLODBC format 314
SQLORA format 314
SQLSYB format 314
Static List validation option in Update Assist 346
Synonym Editor 157, 164, 212
cube data source 212
synonyms 44, 189
syntax coloring 378, 379
T
TABT format 314
tag names 170, 171
join structures and 170, 171
TEMPDIR parameter 308
templates 418, 419
temporary fields 132, 273, 275, 276, 277, 290, 292
calculated values 290
creating 132
creating with COMPUTE 290
DEFINE FUNCTION 292
evaluation 275
types 273
temporary tables
  HOLD files 326, 327
text
  changing colors 378, 379
  finding 378
  replacing 378
text colors 378, 379
  changing 379
text components 376
  creating 376
  adding footings 380
  adding headings 380
case sensitivity 378, 380
changing default settings 378, 379
components 369
creating procedures 375
creating text files 375
opening files 377
opening graphical tools 382, 383
text editor (continued)
  Other component 371
  text editor window 378

  text file format 311
  text files 375
    creating 375
tools for reporting 132
tree controls 208, 211
  properties 211
Tree navigation option for Update Assist 349, 355
  customizing 355

U

universal join structures 146
Update Assist 341, 342, 344, 346, 347, 349, 351,
  352, 353, 354, 357, 358
  about 353
  Automatic validation 342
  autonumbering fields 357
  calling applications from WebFOCUS reports 358
  Combo box navigation option 349
  compiling procedures 352
  Dynamic List validation option 347
  editing applications 352
  flat files and Dynamic List validation 347
  generating application 341
  List validation option 346, 347
  renaming files 351
  renaming segments or fields 344
  resorting fields 344
  running applications 353
  sorting fields 344
  User interface 349
  validation 342

uploading data files 46

V

validationInit.js files 353
variables 123, 190, 197
  setting deployment path 123
version control 64
viewing qualified field names 171, 174
viewing reports
  HTML 330
virtual fields 141, 144, 167, 168, 273, 276, 277,
  279, 283, 287, 288, 415
  calculated values 276
  clearing 287
  creating 277
date format 279
Define tool 283
deleting 415
dynamically formatting 288
editing 415
join structures 141, 144
join structures and 167, 168
virtual folders 25, 32
  adding 32
virtual subfolders 34
VISDIS format 314

W

Web deployment subfolders 112
Web server deployment path 124
WebFOCUS code 369, 370
  editing 369
WebFOCUS files
  Windows 241
WebFOCUS Reporting Server deployment path 125
WEBVIEWER parameter 298
when conditions and hierarchies 199
WHERE statements 417
Windows
  Allocation Wizard 241
defining files 241
  FILEDEF command 241
  logical names 241
workspace file 36, 37, 41, 42, 43, 44
  changing assigned projects 43
workspace file (continued)
  changing WebFOCUS environment 42
  closing files 41
  creating files 36, 37
  inserting projects 44
  loading files 41
  properties dialog box 43
  removing files 43
  renaming files 42

workspace file (continued)
  saving files 41
  WP format 311, 314

X

  XML control file 27
Customer Connections

Where users partner with Information Builders to provide feedback and create quality documentation.

Partner With Us

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also join a usability study to get an early look at new documentation products and services. Your participation helps us create documentation products that meet your needs.

To send us feedback or make a customer connection, contact Michele Brady, Service Director, Technical Content Management at Michele_Brady@ibi.com.

Request Permission to Use Copyright Material

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at Frances_Gambino@ibi.com.

Visit the Information Builders Technical Content Library

To visit the Technical Content Library, go to http://documentation.informationbuilders.com/.