

# **WebFOCUS**

Designing User Interfaces for  
Windows Reporting Applications

EDA, FIDEL, FOCUS, FOCUS Fusion, Information Builders, the Information Builders logo, TableTalk, and Web390 are registered trademarks and SiteAnalyzer, SmartMart, WebFOCUS, and WorldMART are trademarks of Information Builders, Inc.

Acrobat and Adobe are registered trademarks of Adobe Systems Incorporated.

AvantGo is a trademark of AvantGo, Inc.

WebLogic is a registered trademark of BEA Systems, Inc.

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

Informix is a registered trademark of Informix Software, Inc.

Intel is a registered trademark of Intel Corporation.

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

INTERSOLVE and Q+E are registered trademarks of INTERSOLVE.

Orbix is a registered trademark of Iona Technologies Inc.

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

Motorola is a trademark of Motorola, Inc.

Teradata is a registered trademark of NCR International, Inc.

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

Nextel is a registered trademark of Nextel Communications.

NetWare and Novell are registered trademarks of Novell, Inc.

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

Palm is a trademark and Palm OS is a registered trademark of Palm Inc.

PeopleSoft is a registered trademark of PeopleSoft, Inc.

INFOAccess is a trademark of Pioneer Systems, Inc.

BlackBerry is a trademark and RIM is a registered trademark of Research In Motion Limited.

SAP and SAP R/3 are registered trademarks and SAP Business Information Warehouse and SAP BW are trademarks of SAP AG.

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

Unicode is a trademark of Unicode, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

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

Copyright © 2001, by Information Builders, Inc. All rights reserved. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Printed in the U.S.A.

---

---

# Contents

<b>Designing User Interfaces for Windows Reporting Applications .....</b>	<b>1</b>
Controlling Applications .....	2
Controlling Application Flow in a Form .....	2
Navigating Through Forms .....	4
Creating a Windows-based Form .....	5
The Layout of the Forms Painter .....	6
Editing Multiple Forms From the Forms Painter .....	8
Adding a New Form to an Existing Form File .....	8
Changing the Form Name .....	9
Cutting and Pasting Forms .....	9
Saving Your Winforms .....	9
Designing the Form Window .....	9
Specifying Form Objects .....	10
Adding Menus to the Form Window .....	15
Changing Form Windows Properties .....	17
Formatting Text .....	19
Specifying Triggers .....	20
Creating Triggers .....	21
Editing Triggers .....	21
Using Variables .....	24
Assigning Values to Variables .....	24
Reading Variables .....	24
Sample Application .....	25
Running the Application .....	25
Designing the Application .....	26

## Contents

Changing Properties of Form Objects .....	33
Display Objects .....	33
Entry Field Tool .....	34
List Box Tool .....	37
Creating a List in the Forms Painter .....	41
Radio Button List Tool .....	43
Combo Box Tool .....	45
Check Box Tool .....	48
Text Box Tool .....	49
Push Button Tool .....	52
Picture Box Tool .....	54
Picture Button Tool .....	55
Output Display Box Tool .....	57
Hyper Region Tool .....	59
Grid Box Tool .....	60
Form Box Tool .....	62
Group Box Tool .....	64
VBX Custom Controls Tool .....	65
Displaying Forms With the WINFORM Command .....	66
Dynamically Altering a Form .....	67
Examples of Form Objects .....	70
Getting the Form Values .....	71
Creating Forms For Managed Reporting .....	72
Designing User Interfaces for WebFOCUS Tools .....	72
Using the FOCShell Command Language .....	76
Migrating Existing WINFORMs to Web Forms .....	76
Exiting the Forms Painter .....	77

---

---

# Designing User Interfaces for Windows Reporting Applications

## Topics:

- Creating a Windows-based Form
- Designing the Form Window
- Specifying Triggers
- Using Variables
- Sample Application
- Changing Properties of Form Objects
- Displaying Forms With the WINFORM Command
- Dynamically Altering a Form
- Designing User Interfaces for WebFOCUS Tools
- Migrating Existing WINFORMs to Web Forms
- Exiting the Forms Painter

User interfaces are implemented with forms. A form is a window that displays information or otherwise enables a user to interact with an application.

## Controlling Applications

The application developer can use one of two methods to control the flow of execution of the procedures within an application:

Method	Procedure
Form Driven	Design an anchor form to control the flow of execution. Additional forms are called from the main form. All of the procedures in the application are called directly from a form file.
Procedure Driven	Use Dialogue Manager control statements to control the flow of an application. Individual forms are called from separate procedures in the application. For details on Dialogue Manager, see your WebFOCUS 4.3 documentation.  This method has the advantage of giving the designer more flexibility in managing and reusing components of an application.

Forms can be used as an anchor to control an application, or as a standalone linear application design. Forms use variables to manage the flow of execution.

## Controlling Application Flow in a Form

You use variables (ampersand or amper variables) to control the application flow from within a form. The action of the user determines the value of a variable. Variables enable you to design a system that determines the selected options and acts accordingly.

When designing a form, you add objects to the form window that enable users to:

- Read information.
- Make decisions.
- Execute procedures, transfer to another form, or exit an application.
- View output or results.

There are different types of form objects for each type of action required by the user. The variable is the mechanism that enables the application to determine what decisions are made; the application then takes the appropriate action and displays the result.

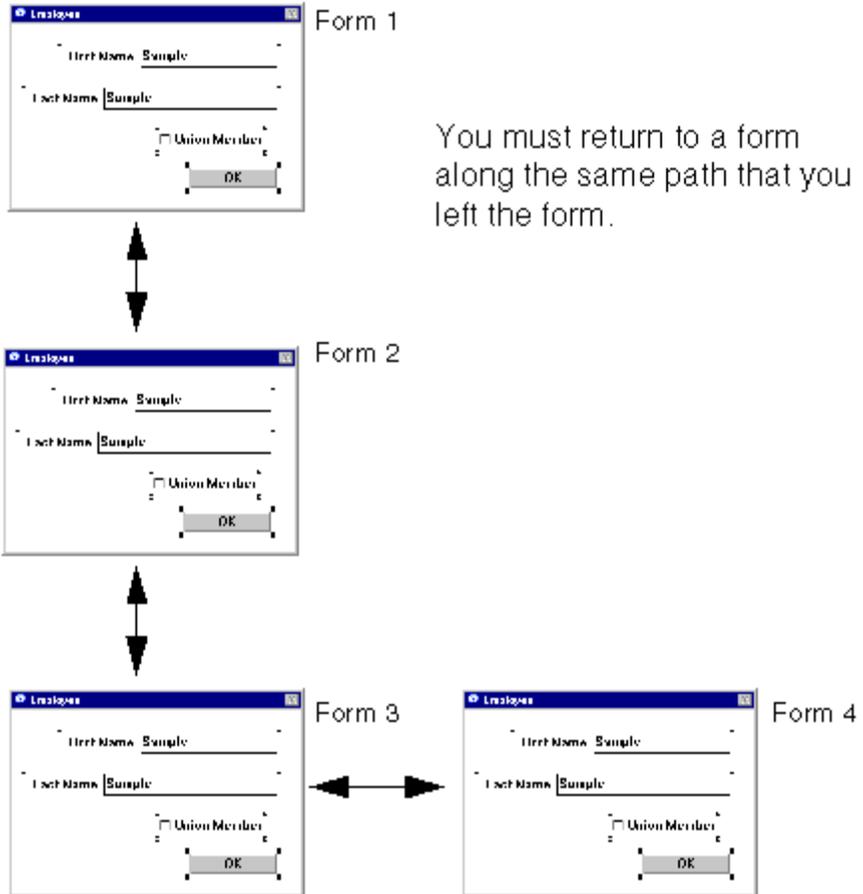
The table below describes typical uses of form objects. However, you can use most objects to perform a variety of functions.

<b>Function</b>	<b>Type of Form Object</b>
Displaying information	Text Box Picture Box
Decision making	Radio Button List List Box Combo Box Check Box
Executing procedures, exiting, and transferring control	Push Button Picture Button Hyper Region VBX Custom Controls
Displaying graph and report output	Output Display Box Grid Box Picture Box Form Box Text Box

## Navigating Through Forms

You can create many forms in an application, and call and close forms from within a form. These forms are created in a "tree" or path-type hierarchy. When creating an application, consider the following:

- You must cycle through forms following the tree hierarchy, or navigation path. The following diagram illustrates this hierarchy:



For example, suppose you want users to begin with Form 1, then navigate to Forms 2, 3, and 4. When they finish with Form 4, you want them to return to Form 2.

Returning to Form 2 directly from Form 4 opens a new instance of Form 1, leaving the original Form 1 open. Therefore, when creating applications, you should allow the user to close Forms 3 and 4, or close these forms using the `-WINFORM ... ALTER` command. This returns the user to Form 2 while still navigating within the form hierarchy.

- When you use the `-WINFORM ... ALTER` command to close a form, you should return to any preceding form through the “closed” form. For example, users can navigate from Form 1 to Form 2 to Form 3, close Form 2, then open Form 4 and close Form 3. When you close Form 4, WebFOCUS automatically navigates back through the closed forms to the first form that is not closed in the navigation path, which is Form 1 in this example.
- You can hide a form when moving to another form. However, you must still move back through the hidden form when returning to an earlier form.
- There are two ways to close a form: explicitly using the Close trigger, or through the `-WINFORM ... CLOSE` command. See *Editing Triggers* on page 21 for more information about the close trigger.

See *Displaying Forms With the WINFORM Command* on page 66 for more information about closing and hiding forms.

## Creating a Windows-based Form

---

You create and store Windows-based forms (Winforms) in form files. You can store many forms in one form file. Typically, you will use a single form file to store all of the forms for one application.

Winforms are always created as components within a procedure that usually contains other components such as Joins, Defines, Reports and Graphs. They do not exist as separate entities.

### **Procedure** How to Create a Winform

To create a form within a procedure:

1. Right-click the procedure in the Procedures folder and choose *Open* from the shortcut menu. The Procedure (FOCEXEC) window opens.
2. Click and hold a component connector (yellow diamond) at the point where you want to include the form in the procedure, then click the *Form* button on the Component Connector toolbox.

The New Form dialog box opens.

3. To create a new form file, enter a new file name in the Form File box.
4. Enter a form name in the Form Name box.

5. Click *OK*.

The Forms Painter opens. You can now create your user interface by add controls, menus, and events to the form.

### **Procedure** How to Edit a Winform

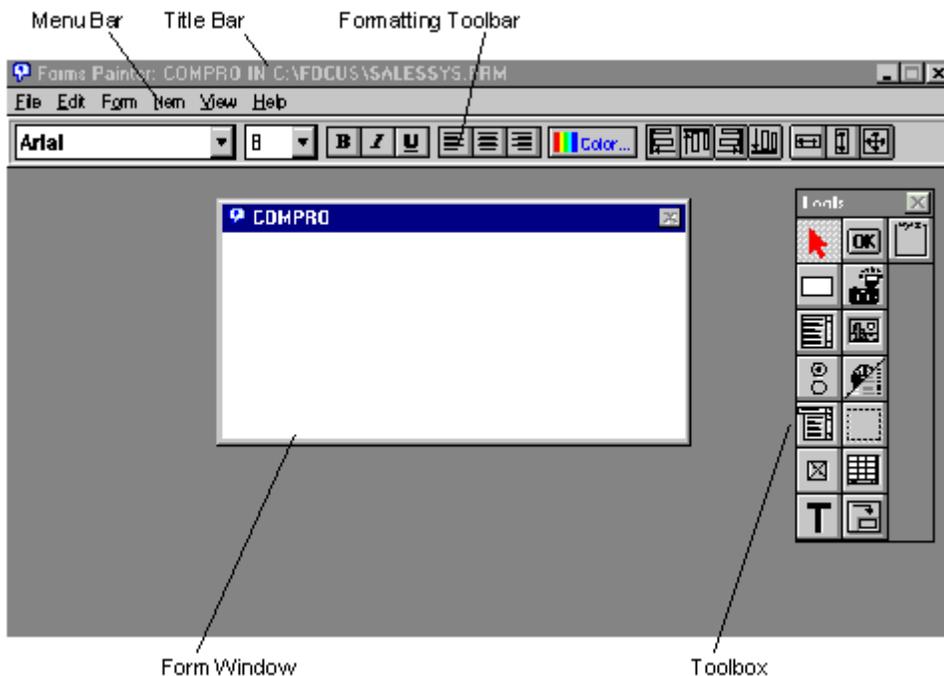
1. Right-click a procedure in the Procedures folder and choose *Open* from the shortcut menu. In the Procedure (FOCEXEC) window, right-click a Form component and choose *Edit* from the shortcut menu.

The New Form dialog box opens.

2. Select the form file from the Form File drop-down combo box.
3. Select the form name from the Form Name drop-down combo box.
4. Click *OK*.
5. The Forms Painter opens with the Form you have chosen to edit.

### **The Layout of the Forms Painter**

When the Forms Painter opens a new form, it appears as shown:



The following table describes the main components of the Forms Painter:

<b>Component</b>	<b>Description</b>
Title Bar	Identifies the Forms Painter and the current form file. You can drag the title bar to different parts of the screen when working on a full screen form to move the form.
Menu Bar	Provides pull-down menus for the Forms Painter.
Formatting Toolbar	Provides drop-down combo boxes and buttons that enable you to edit text or manipulate form objects.
Form Window	Displays a visual representation of the form window. All control objects and text are added to this window.
Toolbox	Enables you to add control objects to the form.

### **Procedure** Displaying Optional Design Aids

To display one of the design aids, choose the appropriate option from the View menu.

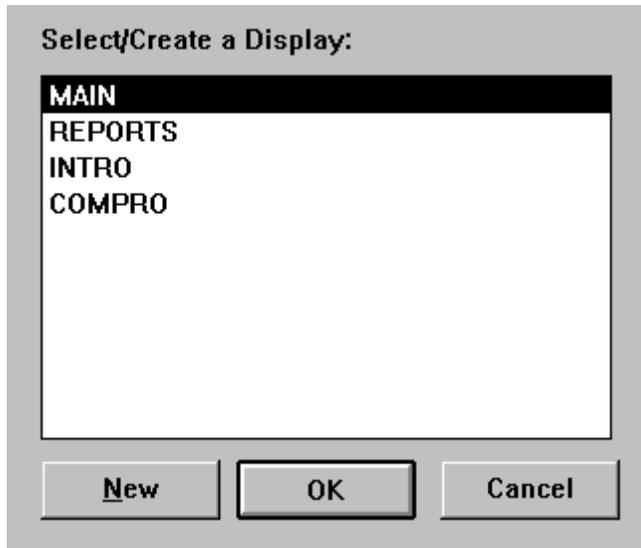
<b>Choose...</b>	<b>To display...</b>
<b>Tool Palette</b>	The toolbox.
<b>Format Bar</b>	The Formatting toolbar.
<b>Status Bar</b>	The Forms Painter status bar, which displays information about selected menu items and the names of toolbox tools.
<b>Grid</b>	A grid to position and align objects in the form window. The Grid dialog box opens, enabling you to specify the following options: <i>Grid Spacing</i> Specify grid spacing in pixels. <i>Show Grid</i> Display grid coordinates. <i>Snap to Grid</i> Automatically position form objects at grid coordinates.
<b>Rulers</b>	Rulers on top and left of the form window.

## Editing Multiple Forms From the Forms Painter

From the Forms Painter, you can open and edit several forms in the same form file. To edit an existing form that belongs to the same form file:

1. Choose *Open* from the Form menu.

The Select/Create a Display dialog box opens:



2. Select a form name from the list.
3. Click *OK*.

The selected form opens.

**Note:** You can only open one form file at a time, but you can open several form files in one session.

## Adding a New Form to an Existing Form File

To add a new form to an existing form file:

1. Choose *Open* from the Form menu.

The Select/Create a Display dialog box opens.

2. Click *New*.

The Form Properties dialog box opens.

3. Enter the Form Name and Title.

4. Click *OK*.

The form window for the new form opens.

## Changing the Form Name

To change the form name:

1. Choose *Properties* from the Form menu.

The Form Properties dialog box opens.

2. Enter a new name in the Form Name box.
3. Click *OK*.

## Cutting and Pasting Forms

The Edit menu options enable you to cut and paste entire forms.

Task	Procedure
Copy the entire form to the Clipboard	Choose <i>Copy</i> from the Edit menu.
Paste an entire form from the Clipboard	Choose <i>Paste</i> from the Edit menu.

## Saving Your Winforms

To save the current changes to the form file, select *Save* from the File menu.

If your form is covering the menu bar, press Alt+F4, then select *Save* from the menu.

If you wish to convert the presentation layer of your Winform to a Webform, you can do so by choosing *Save As* from the File menu and saving the form in .HTM format. However, we recommend that you use the more robust procedures provided for converting both the presentation and process layers of your Winforms. For more information see *Migrating Existing WINFORMs to Web Forms* on page 76.

## Designing the Form Window

---

You can add form objects and menus to the form window in order to allow the user to interact with the application.

## Specifying Form Objects

You can add the following form objects using the toolbox.

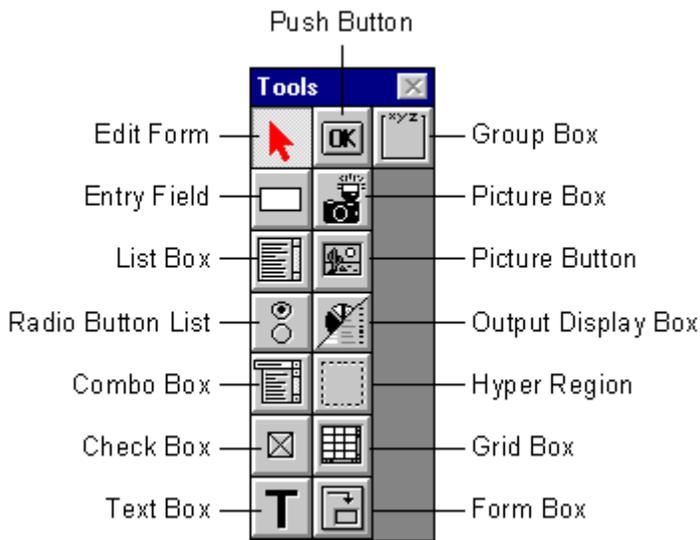
Button	Object Type	Function
	Entry Field	Enter text.
	List Box	Select one item from a scrollable list of items.
	Radio Button List	Select one item from a group of five or less.
	Combo Box	Combination entry field and List Box. Select from a list or enter a value.
	Check Box	Turn option on or off.
	Text Box	Display text.
	Push Button	Execute an action.
	Picture Box	Display a bitmap image.
	Picture Button	Execute an action
	Output Display Box	Display report or graph output.
	Hyper Region	Execute an action. (This is a transparent button that is placed over an area in a picture box object.)

Button	Object Type	Function
	Grid Box	Display a HOLD file in a grid.
	Form Box	View an embedded form.
	Group Box	Visually group form objects.

See *VBX Custom Controls Tool* on page 65 for information about an additional tool, the VBX custom controls tool.

**Procedure Adding Form Objects**

The toolbox enables you to add form objects to a form.



To add an object to a form:

1. Click an object button on the toolbox.
2. Position the mouse pointer in the form window at the position where you want the upper left corner of the object to appear.

The mouse pointer changes to a cross-hair.

3. Use the cross-hair to paint the new form object. Proceed as follows:
  - a. Click and hold the left mouse button.
  - b. Drag the cross-hair down and to the right until the box is correctly sized, then release the mouse button.

The object is added to the screen, and the Properties dialog box for the new object opens.
4. Specify the properties for this object in the Properties dialog box. See *Changing Properties of Form Objects* on page 33 for more information about object properties.
5. After specifying the object properties, exit the Properties dialog box by performing one of the following actions:
  - Double-click the control menu.
  - Click the control menu, then choose *Close*.
  - Press Alt+F4.

The specified properties are assigned to the object.

### **Procedure** Cutting and Pasting Form Objects

The Edit menu options enable you to cut and paste form objects.

<b>Task</b>	<b>Procedure</b>
Remove the selected object and place a copy on the Clipboard	Select <i>Cut Object</i> from the Edit menu.
Copy the selected object to the Clipboard	Select <i>Copy Object</i> from the Edit menu.
Paste a form object from the Clipboard to the currently active form window	Select <i>Paste Object</i> from the Edit menu.

### **Procedure** Deleting Objects

To delete a form object, select it, then choose one of the following options:

- Press the *Delete* key.
- Select *Delete* from the Item menu.

**Procedure Moving Form Objects**

The following table describes how to move objects in a form window.

Task	Procedure
Move an individual object	Select the object and drag it to a new location.
Move a group of objects	Multi-select the objects and drag them to a new location.  <b>Note:</b> To multi-select objects, hold the shift key while selecting each object.

**Procedure Aligning Form Objects**

To align a group of objects with each other:

1. Select all objects to be aligned, selecting the anchor object (the object to which you want to align the objects) last.
2. Click the appropriate alignment button on the Formatting toolbar.

The selected objects align to the border of the anchor object.

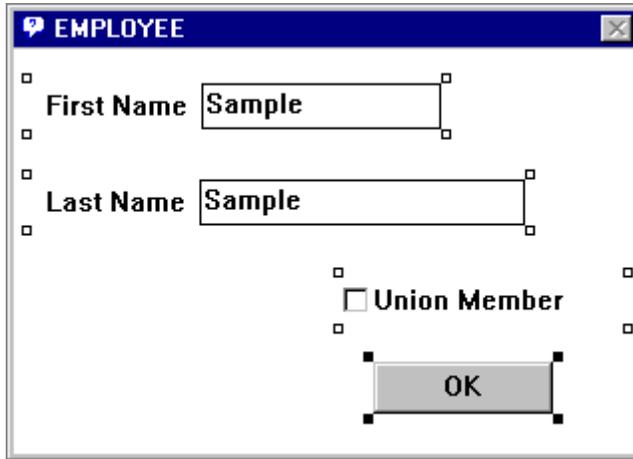
The following alignment buttons are available on the Formatting toolbar:

Click...	To...
	Left-align selected objects.
	Top-align selected objects.
	Right-align selected objects.
	Bottom-align selected objects.

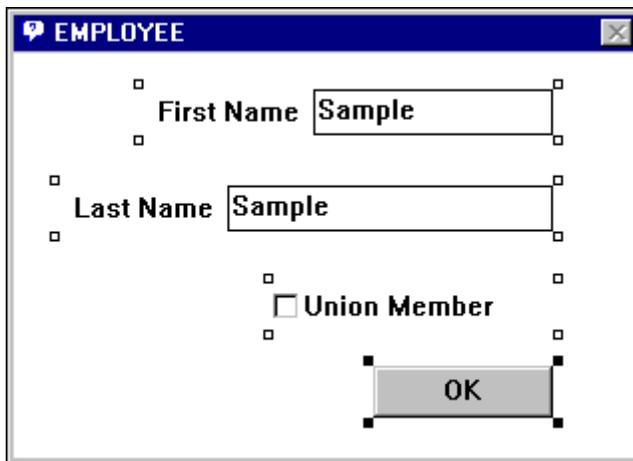
**Example**    **Aligning Form Objects**

You can align the objects in a form. To align the objects with the right border of the *OK* push button:

1. Press and hold the *Shift* key and select each object; select the *OK* push button last.



2. Click the *right-align* button on the Formatting toolbar.  
The objects realign.



**Procedure Resizing Objects**

To resize an individual object, select the object and drag its corner handle (black marker) to a new location.

To resize a group of objects to the same size:

1. Select all objects to be resized, selecting the anchor object last.
2. Click the appropriate resize button on the Formatting toolbar.

The selected objects are resized to match the size of the anchor object.

The following resize buttons are available on the Formatting toolbar.

Click...	To resize...
	The width of selected objects.
	The height of selected objects.
	The width and height of selected objects.

**Adding Menus to the Form Window**

You can include a menu bar in your form window that displays drop-down menus.

To create a menu bar and specify menus and menu items:

- Select *Menus* from the Form menu. The Menus dialog box opens.

You can perform the following using the Menus dialog box.

Task	Procedure
Add a menu to the menu bar	<ol style="list-style-type: none"> <li>1. Click the <i>Menu</i> button at the lower left of the dialog box.</li> <li>2. Enter the menu name in the Label box.</li> </ol> <p>The new menu is added to the Menu list.</p>

Task	Procedure
Add a menu item to a menu	<p>In the Menus dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the menu from the Menu list (if it is not already selected) to which this item will be added.</li> </ol> <p>The menu items for the selected menu display in the Menu Items list.</p> <ol style="list-style-type: none"> <li>2. Click the <i>Item</i> button.</li> <li>3. Enter the item name in the Label box.</li> </ol> <p>The new item is added to the Menu Items list.</p>
Add a separator bar (to separate items in a menu)	<p>To add a separator bar to separate items in a menu, click the <i>Separator</i> button in the Menus dialog box.</p>
Edit a menu name	<p>In the Menus dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the name in the Menu list.</li> <li>2. Edit the name in the Label box.</li> </ol>
Edit a menu item name	<p>In the Menus dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the name in the Menu Items list.</li> <li>2. Edit the name in the Label box.</li> </ol>
Edit a list of menus or menu items	<p>In the Menus dialog box:</p> <ol style="list-style-type: none"> <li>1. Select a menu from the Menu list (or a menu item from the Menu Item list).</li> <li>2. Perform one of the following actions: <ul style="list-style-type: none"> <li>• Click the <i>Up</i> button to move the menu (or item) up the list.</li> <li>• Click the <i>Down</i> button to move the menu (or item) down the list.</li> <li>• Click the <i>Delete button</i> to delete the menu (or item).</li> </ul> </li> </ol>
Specify the trigger event for the selected menu item	<p>Click <i>Triggers in the</i> Menus dialog box.</p> <p>For more information, see on page .</p>

**Note:** The Label box shows the name of the most recently selected menu or menu item name.

## **Reference** Specifying Keyboard Access Characters

Keyboard access characters are automatically assigned to menus and menu items. You can use these characters to display drop-down menus or to choose menu items. To use a keyboard access character, press Alt and the character simultaneously.

The first letter in the menu or menu item name is used as the keyboard access character. To specify a letter other than the first to be the hot key:

- Insert an ampersand (&) just before that letter in the menu or menu choice's name. The letter is underlined at runtime to indicate that it is the hot key.

## **Changing Form Windows Properties**

Double-click the form. The Form Properties dialog box opens.

The following topics explain how to:

- Change the title bar text.
- Make the form window movable.
- Make the form window resizable.
- Display/hide the control menu.
- Order cursor movement between objects in a form window.
- Execute a procedure before or after the form opens.
- Allow an application to be minimized.

## **Procedure** Changing the Form Window Title

The form window title is a unique label that is displayed in the title bar. The form name is used as the default title.

To specify the form window title:

1. Choose *Properties* from the Form menu.  
The Form Properties dialog box opens.
2. Enter the title in the Title box.
3. Click *OK*.

**Procedure Making the Window Movable**

The user can move a movable form window around the screen at runtime.

To make the form window movable:

1. Choose *Properties* from the Form menu.  
The Form Properties dialog box opens.
2. Select the *Movable* check box.
3. Click *OK*.

**Procedure Making the Window Resizable**

A resizable form window can be resized by the user at runtime.

To make the form window resizable:

1. Choose *Properties* from the Form menu.  
The Form Properties dialog box opens.
2. Select the *Sizable* check box.
3. Click *OK*.

**Procedure Allowing an Application to be Minimized**

You can allow a form application to be minimized at runtime. If the user minimizes the active form, all open forms in the form application will be minimized. When the window is restored to its previous size, all other open windows for the application will also be restored.

To allow the form to be minimized:

1. Choose *Properties* from the Form Menu.  
The Form Properties dialog box opens.
2. Select the *Minimize* check box.
3. Click *OK*.

**Procedure Displaying/Hiding the Control Menu**

To display (or hide) the control menu:

1. Choose *Properties* from the Form menu.  
The Form Properties dialog box opens.
2. Select (or deselect) the *Control* menu check box.
3. Click *OK*.

### **Procedure Controlling Cursor Movement**

You can specify the order in which to move between control objects (get focus) when a user presses the tab key:

1. Choose *Tab Order* from the Form menu.  
The Tab Order dialog box opens, listing the objects in the form.
2. Arrange the order of the listed items.
3. Click *OK*.

### **Procedure Displaying a Form Before or After the On Open Trigger Executes**

You can display a form before or after an On Open trigger runs. This is useful if the On Open trigger dynamically alters the appearance of the form using the command `-WINFORM ALTER...` In some cases, you may wish to display the form after the On Open trigger has completed and any dynamic changes to the form have been made.

To display a form *before* a trigger runs:

1. Double-click the form. The Form Properties dialog box opens.
2. Select the *Run on Open* trigger before displaying check box.

The check box is unchecked by default.

To display a form *after* the trigger runs:

- Deselect the *Run on Open* trigger before displaying check box.

See *Display Objects* on page 33 for more information about the form.

### **Formatting Text**

You can change the formatting of text that appears in form objects using the Formatting toolbar. For objects that have only one type of text, such as a text box or push button, you can select the object and then choose a new font property from the Formatting toolbar.

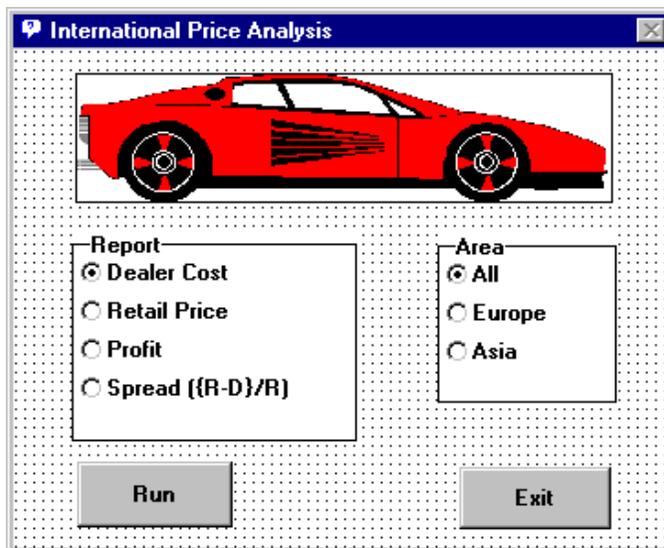
Some objects, such as a radio button list, have label text and item text, which can be formatted using different font types. In this case, open the Properties dialog box for the object in order to specify the type of text (label or item) that you wish to format.

The Formatting toolbar provides the following formatting options:

<b>Typeface</b>	Lists the typefaces (fonts) you have installed on your system.
<b>Font Size</b>	Lists the point sizes available with the typeface currently selected in the Font drop-down combo box.
	Changes font style to and from boldface.
	Changes font style to and from italics.
	Changes font style to and from underline.
	Displays the Windows Color Palette, which enables you to specify the color of the currently selected object. If no object is selected, then the form background color is changed.  <b>Note:</b> The colors displayed in the Color Palette is determined by the number of colors supported by your Windows video driver.

## Specifying Triggers

A trigger is an *event* or *action* that enables a user to interact with a form. For example, consider the application TEN\_MIN. This application is described in detail in *Sample Application* on page 25.



In the main form of this application, the *Run* push button enables a user to run a report. In this example:

- Clicking *Run* is the *event*.
- Running the report is the *action*.

The application developer specified the following triggers for this push button:

<b>Event</b>	Clicked.
<b>Action</b>	<p>Inline Only.</p> <p><i>This is a procedure (FOCEXEC) that you type directly into the Triggers dialog box. In this case, the procedure displays a report.</i></p>

## Creating Triggers

A default trigger is automatically created for each form object when the object is created. The default trigger action is Inline Only. (The default Inline procedure is empty and no action is taken.) You must enter a FOCEXEC for an action to be taken.

## Editing Triggers

To edit the triggers for a form object, perform one of the following actions:

- Select the form object; then choose *Triggers* from the Form menu. The Triggers dialog box opens.
- Choose *Triggers* from the Form menu. The Triggers dialog box opens. Choose the object name from the Object Name drop-down list box.
- From the Properties dialog box for a given object, click *Triggers*. The Triggers dialog box opens.

You can make the following changes to the object triggers using the Triggers dialog box:

<b>Task</b>	<b>Procedure</b>
Select an object other than the default for editing	Select the object from the Object Name drop-down list box in the Triggers dialog box.
Specify an event	Select the event from the Event drop-down list box in the Triggers dialog box.
Specify an action	Select the action from the Action drop-down list box in the Triggers dialog box.

Task	Procedure
Enter an Inline FOCEXEC	Type or copy text to the Inline FOCEXEC box (the text box at the bottom of the dialog box).  <b>Note:</b> You can use hot keys to cut and paste text: Ctrl+X to cut, Ctrl+C to copy, and Ctrl+V to paste. The Edit menu cannot be used to edit text.
Specify action to run immediately after event	Select the <i>Run Immediately</i> check box in the Triggers dialog box.  From the Dialogue Manager perspective, this is done by including the -RUN command right after the trigger. For details on Dialogue Manager, see your WebFOCUS 4.3 documentation.  Using a trigger that does not run immediately is an advanced technique used by developers who want to put off the trigger action so that they can execute additional actions.

### Reference Specifying Events

The event is the action performed by the *user* on the object. When the trigger event occurs, the form invokes the trigger action.

The type of events available for a given form object depends upon the type of object. To specify an event, choose one of the following options from the Event drop-down combo box:

Event	Object Type
<b>Clicked</b>	All objects except Entry Field.
<b>Db1 Clicked</b>	List Box and Hyper Region.
<b>Entry</b>	All objects except Text Box and Picture Box.
<b>Exit</b>	All objects except Text Box and Picture Box.
<b>Changed</b>	Entry Field only.
<b>Drop Down</b>	Combo Box only.
<b>On Open</b>	Display Object only.
<b>On Close</b>	Display Object only.
<b>On Esc Key</b>	Display Object only.
<b>On Enter Key</b>	Display Object only.

You can assign separate actions to each type of event for a given object. For example, you can specify that one message is displayed each time the user enters a value in an entry field, and a second message is displayed each time the value in the entry field changes.

### Reference **Specifying Actions**

The trigger action occurs as a result of the specified event. You can assign separate actions to each type of event for a given object. To specify an action, choose one of the following options from the Action drop-down combo box:

Action	Description
<b>Inline Only</b>	Executes the inline procedure (FOCEXEC). Enter the inline procedure directly into the text box.
<b>External FOCEXEC</b>	Executes an external procedure (FOCEXEC). When you choose this action, the Select a FOCEXEC drop-down combo box displays. You can: <ul style="list-style-type: none"> <li>• Choose a procedure file name from the Action drop-down combo box.</li> <li>• Enter a procedure file name in the Action box.</li> <li>• Enter a variable file name in the Action box.</li> </ul>
<b>Perform Form</b>	Displays a form. When you choose this action, the Select a Form drop-down combo box displays. You can: <ul style="list-style-type: none"> <li>• Choose a procedure file name from the Action drop-down combo box.</li> <li>• Enter a procedure file name in the Action box.</li> <li>• Enter a variable file name in the Action box.</li> </ul> <p><b>Note:</b> If you use a variable file name, you can initialize the variable to the value &lt;NOFORM&gt;. This will prevent an error if no other value has been assigned to the variable.</p>
<b>Close Current Form</b>	Closes the form and returns to the calling form window or FOCEXEC.
<b>Quit Focexec</b>	Ends the application and returns control to the Application Manager.

## Using Variables

---

You can use (amper) variables to manage the application flow from within a form. You can assign variables from within the form objects, enabling you to design a system that determines the selected options and acts accordingly. When designing the form window, you use certain types of objects that automatically assign values to variables. Other types of objects and menu items read the variable and perform an action.

See *Sample Application* on page 25, for an example of using variables in a form.

### Assigning Values to Variables

The following form objects automatically assign a value to a variable when they are selected:

Form Objects	Enables the user to...
Radio Button List	Select one option from a group of five or less.
List Box	Select one item from a list.
Combo Box	Select one item from a list (or enter an item name).
Check Box	Turn the option on/off.
Picture Button	Turn the option on/off.
VBX Custom Controls Button	Select from a list of properties, determined by the VBX that is loaded.

In each case, the user makes a decision that determines the processing of subsequent events. The results of this decision are passed on to subsequent procedures via the variable.

### Reading Variables

The following form objects can be used to perform an action and assign a value to a variable. Typically, the application developer specifies a trigger action of Inline FOCEXEC, and then designs the inline procedure that evaluates the variable and executes the appropriate action.

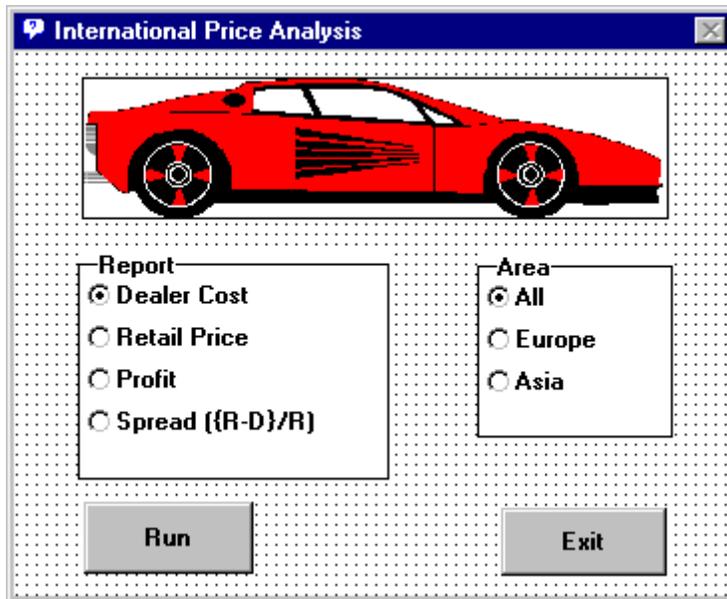
- Push Button
- Picture Button
- Hyper Region
- Menu Items
- VBX Custom Control Button

## Sample Application

The following topics demonstrate the design of a hypothetical application. Note that this is *not* intended as an example to be run. Rather, it is intended to demonstrate the links between a sample form and the behaviors triggered from that form.

### Running the Application

Suppose that the following form opens when you run a hypothetical application called TEN\_MIN, which provides reports analyzing international car sales. This form allows you to select the type of report and the geographic area.

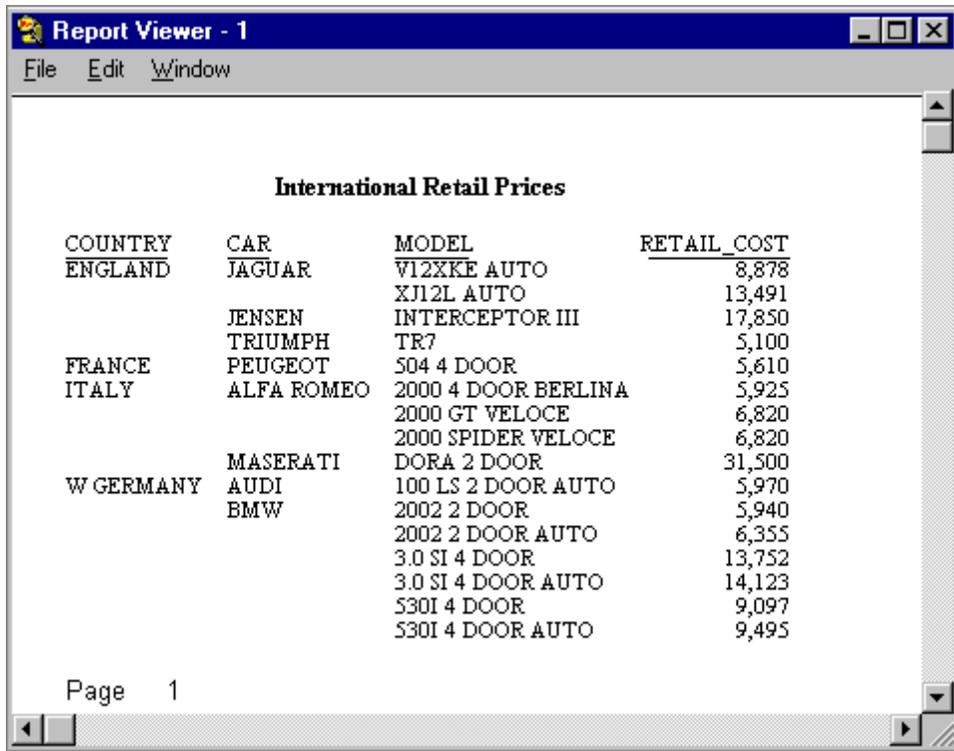


In this application, two radio button lists (Report and Area) enable the user to configure a report before it is displayed. The user then clicks the *Run* push button to display the report.

For example, suppose the user performs the following actions:

1. Selects *Retail Price* from the Report list.
2. Selects *Europe* from the Area list.
3. Clicks *Run*.

The Report Viewer opens and displays a report that shows the retail car prices for the countries in Europe.



The screenshot shows a window titled "Report Viewer - 1" with a menu bar containing "File", "Edit", and "Window". The main content area displays a report titled "International Retail Prices" which is a table with four columns: "COUNTRY", "CAR", "MODEL", and "RETAIL\_COST". The table lists various car models and their retail costs for different countries. At the bottom left of the window, it says "Page 1".

COUNTRY	CAR	MODEL	RETAIL_COST
ENGLAND	JAGUAR	V12XKE AUTO	8,878
		XJ12L AUTO	13,491
FRANCE	JENSEN	INTERCEPTOR III	17,850
	TRIUMPH	TR7	5,100
	PEUGEOT	504 4 DOOR	5,610
ITALY	ALFA ROMEO	2000 4 DOOR BERLINA	5,925
		2000 GT VELOCE	6,820
		2000 SPIDER VELOCE	6,820
W GERMANY	MASERATI	DORA 2 DOOR	31,500
	AUDI	100 LS 2 DOOR AUTO	5,970
	BMW	2002 2 DOOR	5,940
		2002 2 DOOR AUTO	6,355
	3.0 SI 4 DOOR	13,752	
	3.0 SI 4 DOOR AUTO	14,123	
	530I 4 DOOR	9,097	
530I 4 DOOR AUTO	9,495		

## Designing the Application

The application TEN\_MIN is shown in the following application window:



This application consists of six procedures:

Procedure	Function	Creation Method
TEN_MIN	Invokes the main form.	Forms Painter; includes one Inline procedure that contains Dialogue Manager commands.
DEALERC	Displays the dealer costs.	Report Painter
RETAILP	Displays the retail prices.	Report Painter
PROFIT	Displays the profit.	Report Painter
SPREAD	Displays the dealer costs.	Report Painter
AREA	Specifies which countries are included in the report based upon the geographic area selected in the Area radio button list.	WebFOCUS Text Editor: this procedure uses Dialogue Manager commands to evaluate and assign values to variables.

The procedure TEN\_MIN displays the form window shown in *Running the Application* on page 25. This form enables the user to choose the parameters of the displayed report. The syntax for this procedure is

```
-* File TEN_MIN. FEX
-WINFORM MAIN IN PAR
```

where:

`-WINFORM`

Is the Dialogue Manager control statement that invokes a form. For details on Dialogue Manager, see your WebFOCUS 4.3 documentation.

`MAIN`

Is the form name.

`PAR`

Is the name of the form file in which the form is stored.

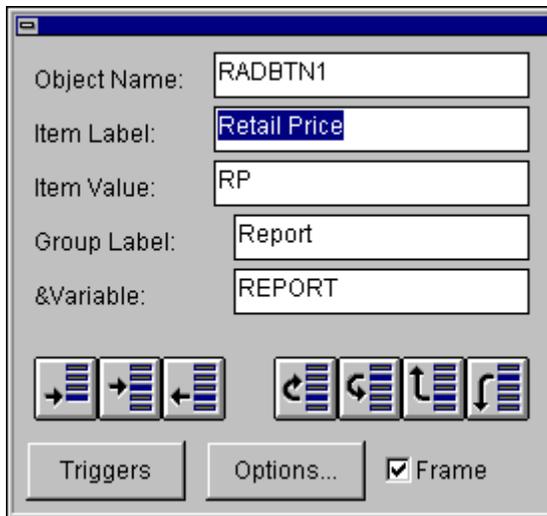
The following topics explain how this form is used to control the application and to display the parameterized reports:

- The Report Radio Button List
- The Area Radio Button List
- The Run Push Button
- The Procedure AREA
- The Procedure RETAILP

## Reference The Report Radio Button List

The *Report* radio button list enables the user to select the type of report to be displayed. In the example below, the *Retail Price* button for this radio button list is being modified in the Properties dialog box. (For more information on radio button lists, see *Changing Properties of Form Objects* on page 33.)

Click the screen items for their descriptions.



The variable for this radio button list is &REPORT. It is assigned one of the following values, depending on the item selected in the radio button list:

Item Label	Item Value
Dealer Cost	DC
Retail Price	RP
Profit	P
Spread	S

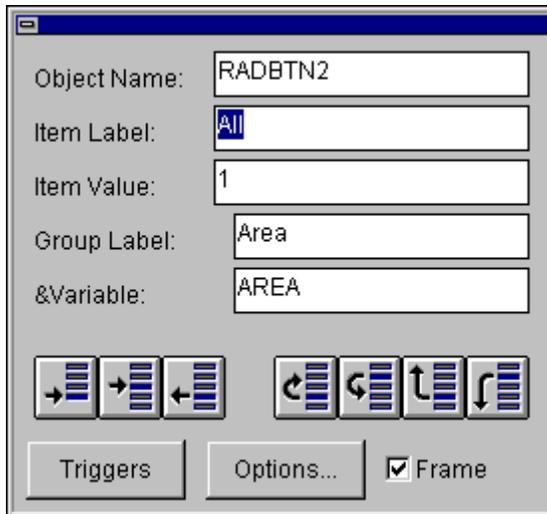
Thus, if the user selects the *Retail Price* button, then &REPORT = 'RP'. This variable is referred to in the inline procedure that is executed when you click the *Run* push button. The variable determines which of the following procedures are run: DEALERC, RETAILP, PROFIT, or SPREAD.

**Note:** If the Item value for this item was left blank, then &REPORT would be assigned the item label—that is &REPORT = 'Retail Report'.

## Reference The Area Radio Button List

The *Area* radio button list enables the user to select the geographic area for the report. In the example below, the *All* button for this radio button list is being edited in the Properties dialog box.

Click the screen items for their descriptions.



The variable for this radio button list is &AREA. It is assigned one of the following values, depending on the item that is currently selected in the radio button list:

Item Label	Item Value
All	1
Europe	2
Asia	3

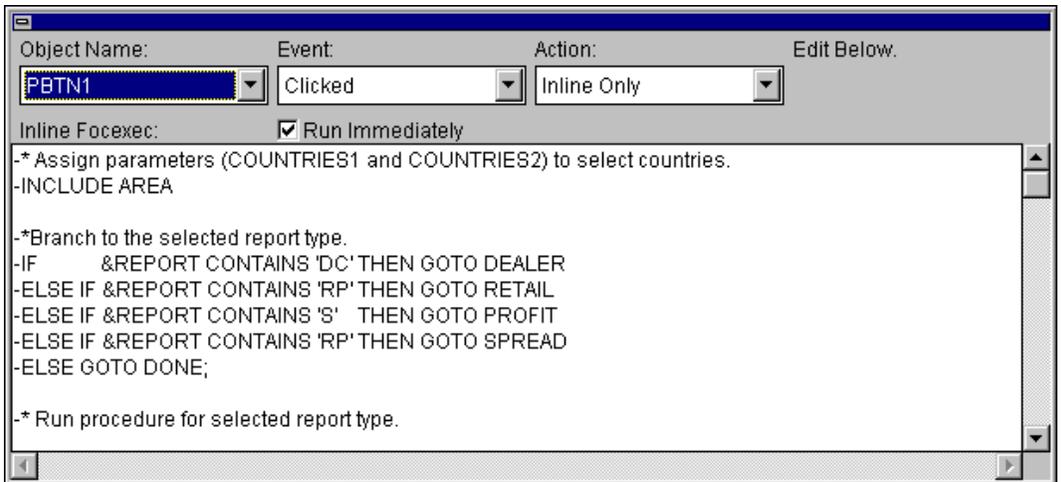
Thus, if the user selects the *All* button, then &AREA='1'. This variable is used in the procedure AREA to determine which countries should be included in the report.

## Reference The Run Push Button

The *Run* push button enables the user to run the parameterized report. When you place the *Run* push button on the form, the Properties dialog box for the *Run* push button displays:



Clicking *Triggers* opens the Trigger dialog box for this push button:



In this dialog box, the following triggers are specified:

<b>Event</b>	Clicked.
<b>Action</b>	Inline Only.  This is a procedure (FOCEXEC) that you type directly into the Triggers dialog box. In this case, the procedure displays a report.

When the user clicks the *Run* push button, the inline procedure (FOCEXEC) executes.

The complete inline procedure for the *Run* push button follows:

```
-* Assign parameters (COUNTRIES1 and COUNTRIES2) to select countries.
-INCLUDE AREA

-* Branch to the selected report type.
-IF          &REPORT CONTAINS 'DC' THEN GOTO DEALER
-ELSE IF &REPORT CONTAINS 'RP' THEN GOTO RETAIL
-ELSE IF &REPORT CONTAINS 'S'   THEN GOTO PROFIT
-ELSE IF &REPORT CONTAINS 'RP' THEN GOTO SPREAD
-ELSE GOTO DONE;

-* Run procedure for selected report type.
-DEALER
-INCLUDE DEALERC
-GOTO DONE
-RETAIL
-INCLUDE RETAILP
-GOTO DONE
-PROFIT
-INCLUDE PROFIT
-GOTO DONE
-SPREAD
-INCLUDE SPREAD
-DONE
```

This procedure performs the following actions:

1. Invokes the procedure AREA to evaluate the variable &AREA and assign values to the variables &COUNTRIES1 and &COUNTRIES2. These variables are referenced by the procedure that generates the report.
2. Evaluates the variable &REPORT and branches to the appropriate procedure to generate a report.
3. Invokes one of the following procedures: DEALERC, RETAILP, PROFIT, or SPREAD. In the case of the report in this example, the procedure RETAILP is invoked.

## **Syntax**      **The Procedure AREA**

The inline procedure for the *Run* push button invokes the procedure AREA. This procedure evaluates the variable &AREA and assigns values to the variables &COUNTRIES1 and &COUNTRIES2. Thus, the countries that are listed in the report are determined by the geographic area selected by the user.

```
-x File Area.fex
-IF      &AREA EQ '1' THEN GOTO ALL
-ELSE IF &AREA EQ '2' THEN GOTO EUROPE
-ELSE IF &AREA EQ '3' THEN GOTO ASIA
-ELSE GOTO ALL;
-ALL
-SET &COUNTRIES1=' "ENGLAND" OR "FRANCE" OR "ITALY" OR "W GERMANY" ';
-SET &COUNTRIES2=' "JAPAN" ';
-GOTO DONE
-EUROPE
-SET &COUNTRIES1=' "ENGLAND" OR "FRANCE" OR "ITALY" OR "W GERMANY" ';
-SET &COUNTRIES2=' " " ';
-GOTO DONE
-ASIA
-SET &COUNTRIES1=' "JAPAN" ';
-SET &COUNTRIES2=' " " ';
-DONE
```

## **Syntax**      **The Procedure RETAILP**

The procedure RETAILP is shown below. This procedure displays the Retail Price report that is shown in *Running the Application* on page 25. The variables &COUNTRIES1 and &COUNTRIES2 specify the countries that are listed in the report.

```
-x File RETAILP.FEX
TABLE FILE CAR
ON TABLE SET PAGE-NUM OFF
PRINT MODEL AND RETAIL_COST BY COUNTRY BY CAR
HEADING CENTER
"International Retail Prices"
" "
FOOTING
"Page <TABPAGENO "
ON TABLE NOTOTAL
WHERE COUNTRY EQ &COUNTRIES1 OR
                &COUNTRIES2;
END
```

## Changing Properties of Form Objects

---

When you add a new form object to a form window, the Properties dialog box for that object automatically opens. Each form object has different properties. This section explains how to change the properties of each type of form object.

See *How to Edit a Winform* on page 6 to learn how to add an object to a form.

To change the properties of an existing form object, either:

- Select the object; then choose *Properties* from the Item menu.
- Double-click the object in the form window.

The Properties dialog box for the selected object opens.

## Display Objects

The form itself is an object. You can specify parameters and triggers that execute when the form opens, or when other events on the form occur. To specify parameters and triggers for the form:

1. Double-click the form.

The Form Properties dialog box opens.

2. Select *Triggers*.

The Triggers dialog box opens.

The following table describes the trigger actions that can occur on the active form:

Task	Procedure
Specify a trigger action when the Enter key is pressed	<ol style="list-style-type: none"> <li>1. Choose <i>On Enter Key</i> from the Event drop-down list box.  The default performs an action when the Enter button is pressed, such as clicking an <i>OK</i> button.</li> <li>2. Specify an action.</li> </ol>
Specify a trigger action when the Escape key is pressed	<ol style="list-style-type: none"> <li>1. Choose <i>On Esc Key</i> from the Event drop-down list box.  The default cancels an action when the <i>Esc</i> button is pressed, such as clicking the <i>Cancel</i> button.</li> <li>2. Specify an action.</li> </ol>

Task	Procedure
Specify a procedure to execute when the form opens	<ol style="list-style-type: none"> <li>1. Choose <i>On Open</i> from the Event drop-down list box.</li> <li>2. Specify an action.</li> </ol>
Specify a procedure to execute when the form closes	<ol style="list-style-type: none"> <li>1. Choose <i>On Close</i> from the Event drop-down list box.</li> <li>2. Specify an action.</li> </ol>

## Entry Field Tool

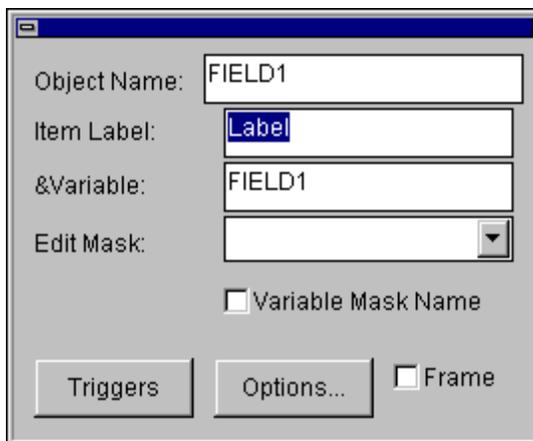


The entry field enables the user to enter a data value.

To add an entry field, click the *Entry Field* button on the toolbox; then paint the entry field on the form. See *Specifying Form Objects* on page 10, for more information about painting objects.

The Properties dialog box for this form object is shown in the following example:

Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.

Task	Procedure
Change the label	In the Properties dialog box, enter the new label in the Item Label box.
Change the variable name	<p>Enter a new variable name in the &amp;Variable box in the Properties dialog box.</p> <p>Variable names can have 12 characters or less with no spaces. Do not include the &amp; symbol in the variable name, unless you wish to create a global variable.</p>
Add an edit mask	<p>In the Properties dialog box, either:</p> <p>Enter the edit mask in the Edit Mask box. An edit mask is an input filter, which, at runtime, restricts the values a user can enter into an entry field.</p> <p>Select a pre-formatted mask from the Edit Mask drop-down combo box. (Pre-formatted masks include phone number or social security number.)</p> <p>The following mask symbols are valid:</p> <ul style="list-style-type: none"> <li># Numeric character.</li> <li>@ Alphabet letter.</li> <li>! Alphabet letter in uppercase.</li> <li>\$ Alphabet letter in lowercase.</li> <li>? Alphanumeric character in uppercase.</li> <li>  Alphanumeric character in lowercase.</li> <li>% Alphanumeric character in upper or lowercase.</li> <li>^ Any character following this value is treated as a constant character.</li> <li>&amp; Any printable character.</li> </ul> <p>For example, eight question marks (?) restricts entry to eight uppercase characters.</p> <p><b>Note:</b> Input is mixed case by default. Input can also be set to uppercase by selecting the <i>Options...</i> button on the Properties dialog box.</p>

Task	Procedure
Convert all input to uppercase	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Click <i>Options...</i> The Options dialog box opens.</li> <li>2. Click the <i>Uppcase User Input</i> check box.</li> </ol> <p><b>Note:</b> You can also use an edit mask to accept only uppercase input.</p>
Include a variable mask name	<p>In the Properties dialog box, enter the variable name in the Edit Mask box and select the <i>Variable Mask Name</i> check box.</p> <p>Use data from a variable to specify your Edit Mask.</p>
Change actions to perform	<p>In the Properties dialog box, click <i>Triggers</i>.</p> <p>The Triggers dialog box opens. For more information, see on page .</p>
Change the object colors	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Click <i>Options...</i> The Options dialog box opens.</li> <li>2. Select one of the following options from the Color Applies To drop-down combo box: <i>All background, Label text, Label background, Input text, or Input background</i>.</li> <li>3. Click the <i>Color</i> button from the Formatting toolbar and select a new color.</li> </ol>
Change the text font	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Click <i>Options...</i> The Options dialog box opens.</li> <li>2. Select one of the following options from the Font Applies To drop-down combo box: <i>Input, or Label and Input</i>.</li> <li>3. Select the desired font properties from the Formatting toolbar.</li> </ol> <p>See on page for more information about formatting text.</p>

Task	Procedure
Perform action only if value changes	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Click <i>Triggers</i>.</li> </ol> <p>The Triggers dialog box opens.</p> <ol style="list-style-type: none"> <li>2. Select <i>On Changed</i> from the Event drop-down combo box.</li> </ol> <p>The action specified for the On Changed trigger event will only be executed if the value of the entry field changes.</p>
Display a frame around the field	<p>In the Properties dialog box, select the <i>Frame</i> check box.</p>
Hide user input	<p>You can hide passwords and other sensitive information as users enter them in the Entry Field. To hide user input:</p> <ol style="list-style-type: none"> <li>1. Click <i>Options...</i></li> </ol> <p>The Options dialog box opens.</p> <ol style="list-style-type: none"> <li>2. Click the <i>Password</i> protect user input check box.</li> </ol> <p><b>Note:</b> Password-protect user input cannot be used in conjunction with edit masks.</p>

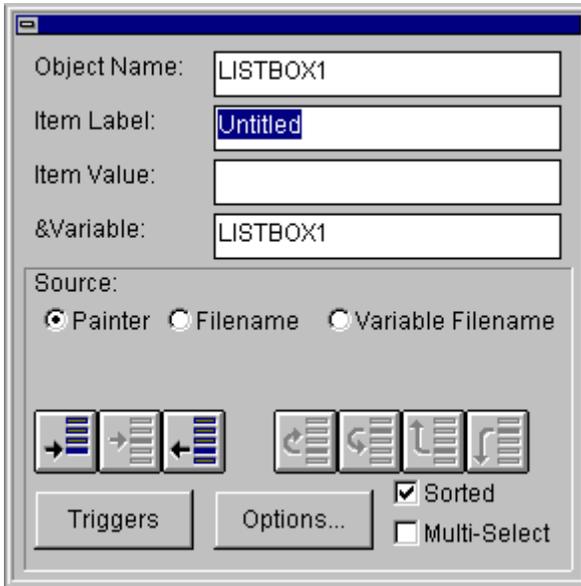
## List Box Tool



A List Box enables the user to select one or multiple items from a scrollable list.

To add a List Box, click the *List Box* button on the toolbox; then paint the list box on the form. See *Specifying Form Objects* on page 10 for more information about adding objects to a form.

The Properties dialog box for this form object is shown in the following example:  
Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Change an item label	In the Properties dialog box, enter the new label in the Item Label box.
Change the item value that is returned to the variable	In the Properties dialog box, enter the new value in the Item Value box.
Change the variable name	Enter a new variable name in the &Variable box in the Properties dialog box.  Variable names can have 12 or fewer characters with no spaces. Do not include the & symbol in the variable name unless you wish to create a global variable.

Task	Procedure
<p>Create a list of items in the Forms Painter</p>	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Painter</i> radio button.</li> <li>2. Deselect the <i>Sorted</i> check box (if it is already selected). It is easier to sort the items after completing the list.</li> <li>3. Enter an item name in the Item Label box.</li> <li>4. If you wish the value of the item to be different from the label name, enter a value in the Item Value box.</li> </ol> <div style="text-align: center;">  </div> <ol style="list-style-type: none"> <li>5. Click  (the <i>Insert</i> button) to insert the next item in the list. Enter the item name and item value for this item. The inserted item is added to the top of the list.</li> <li>6. Repeat Step 5 for each item that you add to the list.</li> <li>7. Use the List buttons to move, add, and delete list items. See on page for more information.</li> <li>8. To sort the items in the list, select the <i>Sort</i> check box.</li> </ol>
<p>Display a list from the contents of a text file</p>	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Filename</i> radio button. The File... button and the Filename box display.</li> <li>2. Either: <ul style="list-style-type: none"> <li>• Click the <i>File...</i> button and select a file from the Open dialog box.</li> <li>• Enter a file name in the Filename box.</li> </ul> </li> </ol> <p>Note that the selected file must have a .TXT extension. The contents of the text file are displayed as a list of items.</p>
<p>Allow multiple items to be selected</p>	<p>In the Properties dialog box, select the <i>Multi-Select</i> check box. See on page for more information.</p>

Task	Procedure
Display a list from the contents of a variable text file	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Variable Filename</i> radio button. The Variable Filename box displays.</li> <li>2. Enter the variable file name in the Variable Filename box.</li> </ol>
Display a list of file names	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Filename</i> radio button. The Filename box displays.</li> <li>2. Enter a file name wildcard (such as *.MAS) in the Filename box. The asterisk (*) and question mark (?) are valid wildcards.</li> </ol> <p><b>Note:</b> WebFOCUS searches the appropriate application paths when displaying Master Files and FOCEXEC file names.</p> <p>By default, the list displays the extension of each file in the list box. To hide the file extension, deselect the <i>Show filename</i> extension check box. The three-character file extension will not be displayed.</p>
Change triggers	<p>In the Properties dialog box, click <i>Triggers</i>.</p> <p>The Triggers dialog box opens. For more information, see on page .</p>
Apply color	<p>In the Properties dialog box, click <i>Options...</i></p> <p>The Options dialog box opens, enabling you to apply color to background or text.</p>
Display list items in sorted order	<p>In the Properties dialog box, select the <i>Sort</i> check box.</p> <p>List items are automatically sorted in alphabetical order.</p>

**Example Processing a Multi-Select List**

The List Box tool enables users to select multiple values from a list. This is useful when users wish to see information about several fields in a report.

When a user selects multiple items from a list box, the variable name for the object is assigned a value. This value depends on the number of items selected from the list. Each value is returned to a new variable with a sequence number added to the variable name.

For example, if a user selects JAPAN and GERMANY from a list box containing the ampersand variable &COUNTRY, the variable is assigned a value of 2, &COUNTRY1 is assigned the value JAPAN, and &COUNTRY2 is assigned the value GERMANY.

To enable Multi-Select:

- Select the *Multi-Select* check box in the Properties dialog box.

The following is an example of an inline FOCEXEC that processes a Multi-Select list. This FOCEXEC processes the values selected by a user in the Countries list box, and uses the newly created variable to select records from the CAR database:

```
-SET &COUNTER=0;  
TABLE FILE CAR  
PRINT COUNTRY CAR MODEL  
-IF &COUNTRY EQ 0 THEN GOTO LABEL1;  
IF COUNTRY IS &COUNTRY1  
-REPEAT LABEL1 FOR &COUNTER FROM 2 TO &COUNTRY  
OR &COUNTRY.&COUNTER  
-LABEL1  
END
```

**Note:** We recommend omitting numerals from variable names when using Multi-Select. WebFOCUS automatically assigns sequence numbers to the variable depending on the number of selected values.

## Creating a List in the Forms Painter

The Properties dialog boxes for the following form objects provide List buttons that enable you to create a list of items:

- Radio Button List
- List Box
- Combo Box

You can use the List buttons to insert, delete, and move the items in a list. The are described in the following table:

<b>Button</b>	<b>Function</b>
	Adds a new list item to the end of the list.
	Inserts a new list item before the one currently being edited.
	Deletes the current list item.
	Moves the current list item up the list one position.
	Moves the current list item down the list one position.
	Moves the current list item to the top of the list.
	Moves the current list item to the end of the list.

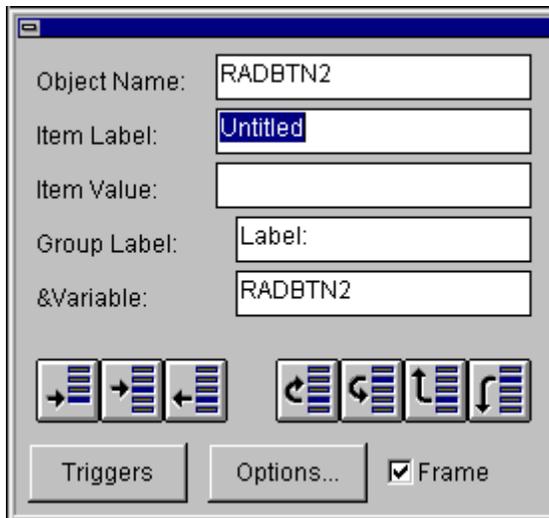
## Radio Button List Tool



A radio button list enables the user to select one item from a group of five or less.

To add a radio button list, click the *Radio* button list tool button on the toolbox; then paint the list on the form. The Properties dialog box for this form object is shown in the following example:

Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Change the label	In the Properties dialog box, enter the new label in the Item Label box.
Change the radio button value that is assigned to the variable	In the Properties dialog box, enter the new value in the Item Value box.

Task	Procedure
Change the variable name	<p>Enter a new variable name in the &amp;Variable box in the Properties dialog box.</p> <p>Variables names can have 12 or fewer characters with no spaces. Do not include the &amp; symbol in the variable name, unless you wish to create a global variable.</p>
Create a list of radio buttons	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Enter the name of the first button in the Item Label box.</li> <li>2. If you wish the value of the button to be different from the label name, enter a value in the Item Value box.</li> <li>3. Click  (the <i>Add</i> button) to add the next item to the list. Enter the item name and item value (if desired) for this item. The added item is added to the bottom of the list.</li> <li>4. Repeat Step 3 for each item that you add to the list.</li> <li>5. Use the List buttons to move, insert, and delete list items. See on page for more information.</li> </ol>
Change triggers	<p>In the Properties dialog box, click <i>Triggers</i>.</p> <p>The Triggers dialog box opens. For more information, see on page .</p>
Apply color	<p>In the Properties dialog box, click <i>Options...</i></p> <p>The Options dialog box opens, enabling you to apply color to background or text.</p>
Display a frame around the list	<p>In the Properties dialog box, select the <i>Frame</i> check box.</p>

See *Sample Application* on page 25 for examples of radio button lists.

## Combo Box Tool

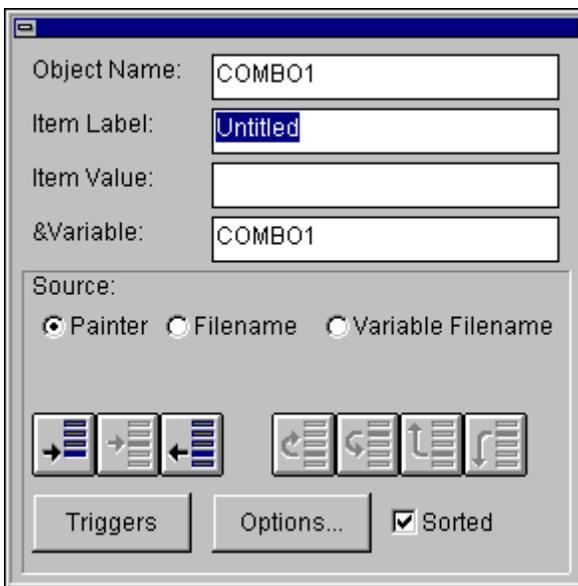


A combo box is a combination of an entry field and a list box. It enables the user to enter data or select an item from a list.

To add a combo box, click the *Combo Box* button on the toolbox; then paint the combo box on the form. See *Specifying Form Objects* on page 10 for more information about painting form objects.

The Properties dialog box for this form object is shown in the following example:

Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Change the label	In the Properties dialog box, enter the new label in the Item Label box.

Task	Procedure
Change the item value that is assigned to the variable	In the Properties dialog box, enter the new value in the Item Value box.
Change the variable name	<p>Enter a new variable name in the &amp;Variable box in the Properties dialog box.</p> <p>Variables names can have 12 or fewer characters with no spaces. Do not include the &amp; symbol in the variable name unless you wish to create a global variable.</p>
Create a list of items	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Painter</i> radio button.</li> <li>2. Enter the name of the first item in the Item Label box.</li> <li>3. If you wish the value of the item to be different from the label name, enter a value in the Item Value box.</li> <li>4. Click  (the <i>Add</i> button) to add the next item to the list. Enter the item name and item value (if desired) for this item.  The item is added to the bottom of the list.</li> <li>5. Repeat Step 4 for each item that you add to the list.</li> <li>6. Use the List buttons to move, insert, and delete list items. See on page for more information.</li> <li>7. To sort the items in the list, select the <i>Sort</i> check box.</li> </ol>

Task	Procedure
<p>Display a list from the contents of a text file</p>	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Filename</i> radio button. The <i>File...</i> button and the Filename box display.</li> <li>2. Either: <ul style="list-style-type: none"> <li>• Click the <i>File...</i> button and select a file from the Open dialog box.</li> <li>• Enter a file name in the Filename box.</li> </ul> </li> </ol> <p>Note that the selected file must have an .FTM extension. The contents of the text file are displayed as a list of items.</p>
<p>Display a list from the contents of a variable text file</p>	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Variable Filename</i> radio button. The Variable Filename box displays.</li> <li>2. Enter the variable file name in the Variable Filename box.</li> </ol>
<p>Display a list of file names</p>	<p>In the Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Filename</i> radio button. The Filename box displays.</li> <li>2. Enter a file name wildcard (such as *.MAS) in the Filename box. The asterisk (*) and question mark (?) are valid wildcards.</li> </ol> <p><b>Note: WebFOCUS</b> searches the appropriate application paths when displaying Master Files and FOCEXEC file names.</p> <p>By default, the list displays the extension of each file in the list box. To hide the file extension, deselect the <i>Show filename</i> extension check box. The three-character file extension will not be displayed.</p>
<p>Change triggers</p>	<p>In the Properties dialog box, click <i>Triggers</i>.</p> <p>The Triggers dialog box opens. For more information, see on page .</p>

Task	Procedure
Apply colors	In the Properties dialog box, click <i>Options...</i> The Options dialog box opens, enabling you to apply color to background or text.
Display a frame around the combo box	In the Properties dialog box, select the <i>Frame</i> check box.

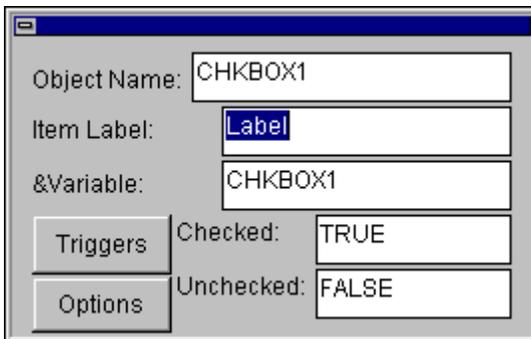
## Check Box Tool



A check box enables the user to turn an option on or off.

To add a check box, click the *Check Box* tool button on the toolbox; then paint the check box on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

The Properties dialog box for this form object is shown in the following example:



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Change the variable name	Enter a new variable name in the &Variable box in the Properties dialog box.  Variables names can have 12 or fewer characters with no blanks. Do not include the & symbol in the variable name unless you wish to create a global variable.

Task	Procedure
Change value of variable	<p>In the Check Box Properties dialog box:</p> <p>To change the value of a variable when the box is checked, enter a new value in the <i>Checked</i> box.</p> <p>To change the value of a variable when the when box is unchecked, enter a new value in the <i>Unchecked</i> box.</p>
Change triggers	<p>In the Properties dialog box, click <i>Triggers</i>.</p> <p>The Triggers dialog box opens. For more information, see on page .</p>
Apply colors	<p>In the Properties dialog box, click <i>Options...</i></p> <p>The Options dialog box opens, enabling you to apply color to background or text.</p>

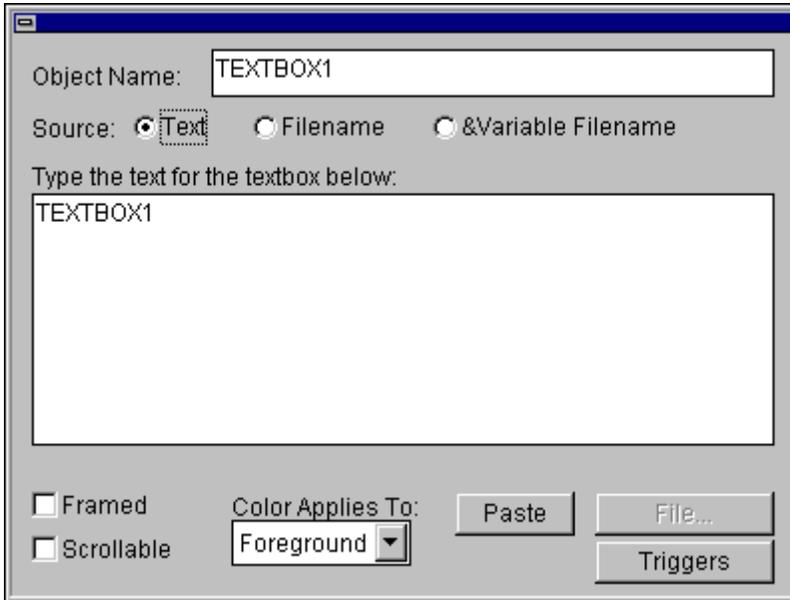
## Text Box Tool



A Text Box displays information on a form.

To add a Text Box, click the *Text Box* tool button on the toolbox; then paint the text box on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

The Properties dialog box for this form object is shown in the following example:  
Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Enter text	In the Text Box Properties dialog box: <ol style="list-style-type: none"><li>1. Select the <i>Text</i> radio button.</li><li>2. Type the text in the Text box.</li></ol> <p><b>Note:</b> You can include a variable in the text. Type the variable name with the text you wish to display.</p>

Task	Procedure
Copy and paste text to/from the Clipboard	<p>In the Text Box Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Text</i> radio button. The Paste button is enabled.</li> <li>2. Perform one of the following actions: <ul style="list-style-type: none"> <li>• Press Ctrl+C to copy text to the Clipboard.</li> <li>• Press Ctrl+X to cut text and copy to the Clipboard.</li> <li>• Click <i>Paste</i> to paste text from the Clipboard.</li> <li>• Press Ctrl+V to paste text from the Clipboard.</li> </ul> </li> </ol>
Display text from the contents of a text file	<p>In the Text Box Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Filename</i> radio button. The Filename box displays and the <i>File...</i> button is enabled.</li> <li>2. Either: <ul style="list-style-type: none"> <li>• Click the <i>File...</i> button and select a file from the Open dialog box.</li> <li>• Enter a file name in the File box.</li> </ul> </li> </ol>
Display a list from the contents of a variable text file	<p>In the Text Box Properties dialog box:</p> <ol style="list-style-type: none"> <li>1. Select the <i>Variable Filename</i> radio button. The Variable Filename box displays.</li> <li>2. Enter the variable file name in the Variable Filename box.</li> </ol>
Change triggers	<p>In the Text Box Properties dialog box, click <i>Triggers</i>. The Triggers dialog box opens. For more information, see on page .</p>

Task	Procedure
Change colors	In the Text Box Properties dialog box: <ol style="list-style-type: none"> <li>1. Select one of the following options from the Color Applies To drop-down combo box:                             <ul style="list-style-type: none"> <li><i>Foreground</i>, to change the text color.</li> <li><i>Background</i>, to change the text box background color.</li> </ul> </li> <li>2. Click <i>Color</i> on the Formatting toolbar and select a new color.</li> </ol>
Format text	In the Text Box Properties dialog box, select a new font parameter from the Formatting toolbar.
Align text	To align the text in a text box, choose one of the following options from the Format bar:
	 Left-align text.
	 Center text.
	 Right-align text.
Display a frame around the text	In the Properties dialog box, select the <i>Framed</i> check box.

## Push Button Tool

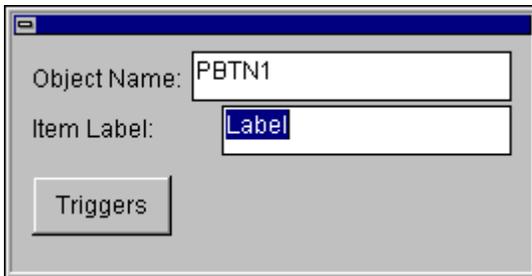


A push (or command) button enables the user to execute an action.

To add a push button, click the *Push Button* tool button on the toolbox; then paint the push button on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

The Properties dialog box for this form object is shown in the following example:

Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Change the button label	In the Properties dialog box, enter the new label in the Item Label box.
Change triggers	In the Properties dialog box, click <i>Triggers</i> . The Triggers dialog box opens. For more information, see on page .

### Reference **Specifying Keyboard Access Characters**

Keyboard access characters are automatically assigned to menus and menu items. In addition, you can assign these characters to activate push buttons. To use a keyboard access character, press the Alt key and the *underlined* character simultaneously.

By default, the first letter in the push button name is used as the keyboard access character. To specify a letter other than the first to be the hot key:

- Insert an ampersand (&) just before that letter in the push button's name. The letter is underlined at runtime to indicate that it is the hot key.

## Picture Box Tool



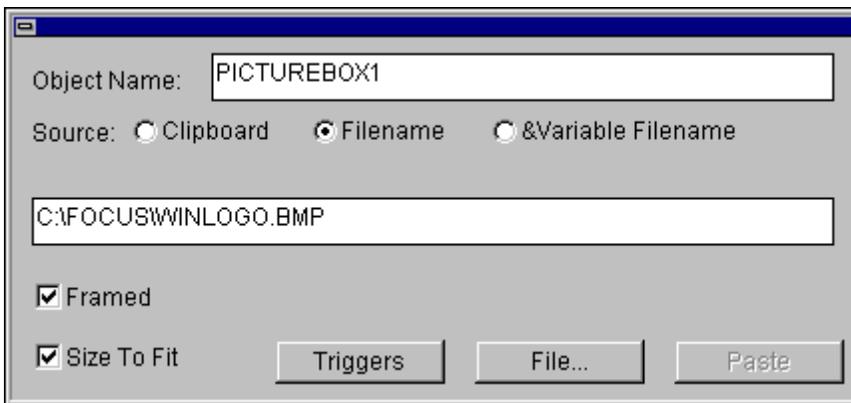
A picture box enables the user to view a graphic image, and execute an action by clicking on it.

You can display an image from a bitmap (BMP) file or from the Clipboard.

To add a picture box, click the *Picture Box* tool button on the toolbox; then paint the picture box on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

The Properties dialog box for this form object is shown in the following example:

Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Paste the image from the Clipboard	In the Picture Box Properties dialog box: <ol style="list-style-type: none"><li>1. Select the <i>Clipboard</i> radio button. The Paste button is enabled.</li><li>2. Click <i>Paste</i>.</li></ol>

Task	Procedure
Display the image from a file	In the Picture Box Properties dialog box: <ol style="list-style-type: none"> <li>1. Select the <i>Filename</i> radio button. The Filename box displays and the <i>File...</i> button is enabled.</li> <li>2. Either:                             <ul style="list-style-type: none"> <li>• Click the <i>File...</i> button and select a file from the Open dialog box.</li> <li>• Enter a file name in the File box.</li> </ul> </li> </ol>
Display a list from the variable image file	In the Picture Box Properties dialog box: <ol style="list-style-type: none"> <li>1. Select the <i>Variable Filename</i> radio button. The Variable Filename box displays.</li> <li>2. Enter the variable file name in the Variable Filename box.</li> </ol>
Change triggers	In the Properties dialog box, click <i>Triggers</i> . The Triggers dialog box opens. For more information, see on page .
Display a frame around the picture box	In the Properties dialog box, select the <i>Framed</i> check box.
Automatically resize image	In the Properties dialog box, select the <i>Size To Fit</i> check box.

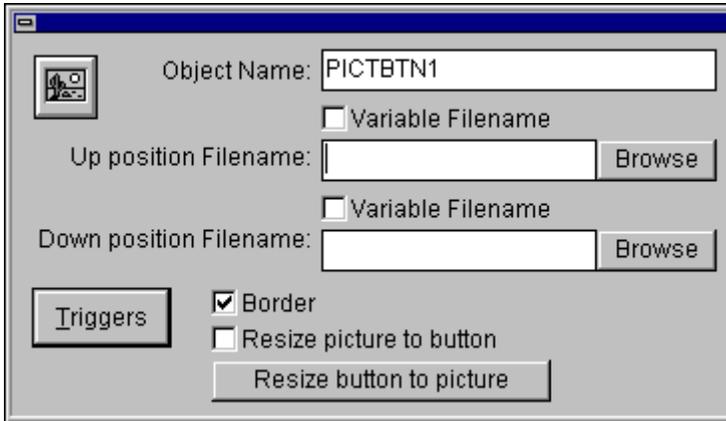
## Picture Button Tool



A picture button is a multi-state button, and like other button controls, executes an action when clicked. By changing its display when clicked, the picture button indicates to the user if the button is *up* or *down*.

To add a picture button, click the *Picture Button* button on the toolbox; then paint the picture button on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

The Properties dialog box for this form object is shown in the following example:  
Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Specify the bitmap file for the up (normal) position	In the Picture Button Properties dialog box, either: <ul style="list-style-type: none"> <li>• Enter the file name in the Up Position Filename box.</li> <li>• Click the <i>Browse</i> button next to the Up Position Filename box; then select a file.</li> <li>• Select the <i>Variable Filename</i> check box; then enter a variable file name in the Up Position Filename box.</li> </ul>
Specify the bitmap file for the down (depressed) position	In the Picture Button Properties dialog box, either: <ul style="list-style-type: none"> <li>• Enter the file name in the Down Position Filename box.</li> <li>• Click the <i>Browse</i> button next to the Down Position Filename box; then select a file.</li> <li>• Select the <i>Variable Filename</i> check box; then enter a variable file name in the Down Position Filename box.</li> </ul>

Task	Procedure
Change triggers	In the Properties dialog box, click <i>Triggers</i> . The Triggers dialog box opens. For more information, see on page .
Resize picture to fit in button	In the Picture Button Properties dialog box, select the check box <i>Resize picture to button</i> .
Resize the button to the size of the picture	In the Picture Button Properties dialog box, click the button <i>Resize button to picture</i> .
Display a border around the button	In the Picture Button Properties dialog box, select the <i>Border</i> check box.

## Output Display Box Tool



An Output Display Box shows the output from a report or graph.

To add an output display box to a form:

1. Click the *Output Display Box* tool button on the toolbox.
2. Paint the output display box on the form.

To add a graph to the output display box:

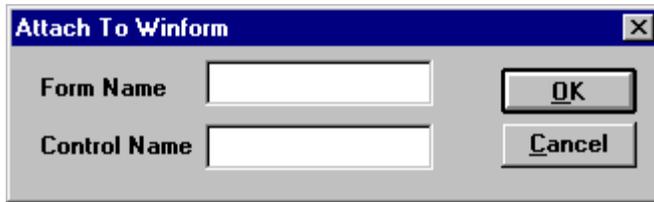
1. Open *Graph Assistant*.
2. Choose the *Properties* tab.
3. Select the *Output to Winform* radio button.
4. Select the form from the Winform box and click *OK*.
5. Close *Graph Assistant* and return to the Forms Painter.
6. Type the object's name in the Object Name field of the Properties dialog box.
7. Close the *Properties* dialog box: double-click the control menu. Click *Yes* to save the changes.

To direct report output to the display box:

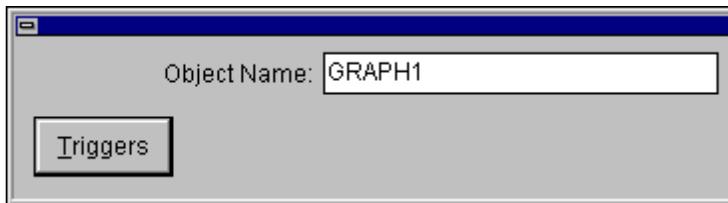
1. Open *Report Painter*.
2. Choose *Output to Winform* from the Report menu.

The Attach to Winform dialog box opens.

Click the screen items for their descriptions.



3. Type the form name and control name in the boxes and click *OK*.
4. Close the *Report Painter* and return to the *Forms Painter*.
5. Type the object's name in the Object Name field of the *Properties* dialog box as shown in the following example:



6. Close the *Properties* dialog box: double-click the control menu. Click *Yes* to save the changes.

You can change the following properties of this object using the *Properties* dialog box.

Task	Procedure
Change the object name	In the <i>Properties</i> dialog box, enter a new object name in the Object Name box.
Change triggers	In the <i>Properties</i> dialog box, click <i>Triggers</i> . The <i>Triggers</i> dialog box opens. For more information, see on page .

**Note:** If the report uses the drill down feature, the subsequent (drill down) report can only call additional drill down reports if they are also displayed on a form. As soon as the *Report Viewer* is used to display a report, there can be no further drill down from that report. This behavior does not apply to reports that are run from the *Report Painter*.

### Example Using Drill Down With an Output Display Box

When you display a drill down report on a form using the Output Display box, you can only call subsequent drill down reports if you also display them on a form.

**Note:** This behavior does not apply for reports that are run from the Report Painter. As soon as the Report Viewer is used to display a report, there can be no further drill down from that report.

To illustrate drill down in a form, the following example uses three reports and the EMPLOYEE database. Each report provides an additional level of detail.

- The report TOT\_SAL displays total salary figures for all the records in the database and allows you to drill down to report DEPT\_SAL for more detail.
- The report DEPT\_SAL displays total salary figures for each department and allows you to drill down to report EMP\_SAL for details within the department.
- EMP\_SAL lists all the employees and their salaries within a department.

You can then design and build a form-based application that displays the output from the TOT\_SAL report on a form, and allows the user to drill down for more information. If the user chooses to drill down, the DEPT\_SAL report executes and displays the results on a form. The user can then perform an additional drill down to the final level of detail, the EMP\_SAL report.

However, if you choose to display the results from DEPT\_SAL in Report Viewer, rather than on a form, you will not be able to drill down from this report to the EMP\_SAL report.

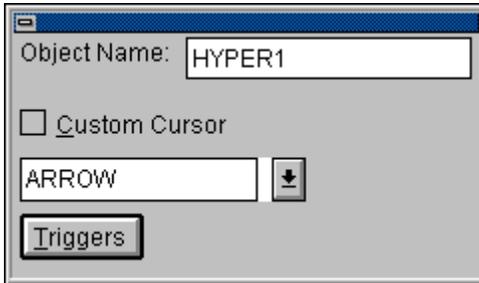
### Hyper Region Tool



The Hyper Region is a transparent button. It can be positioned over any other object, but it is most commonly placed over a picture box containing a map or other geographic image. It enables the user to click an area of a map (or other image) and cause an action to be executed.

To add a hyper region, click the *Hyper Region* button on the toolbox; then paint the hyper region on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

The Properties dialog box for this form object is shown in the following example:  
Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Change triggers	In the Properties dialog box, click <i>Triggers</i> . The Triggers dialog box opens. For more information, see on page .
Display a different cursor when the mouse pointer is over the hyper region	In the Hyper Region Properties dialog box: <ol style="list-style-type: none"> <li>1. Choose a cursor from the drop-down list box. or Select the <i>Custom Cursor</i> check box.</li> <li>2. Click the <i>File...</i> button.</li> <li>3. Specify a custom cursor. This is the cursor that displays when users move the mouse over the hyper region.</li> </ol> <p><b>Note:</b> Custom cursors must have a .CUR extension.</p>

## Grid Box Tool

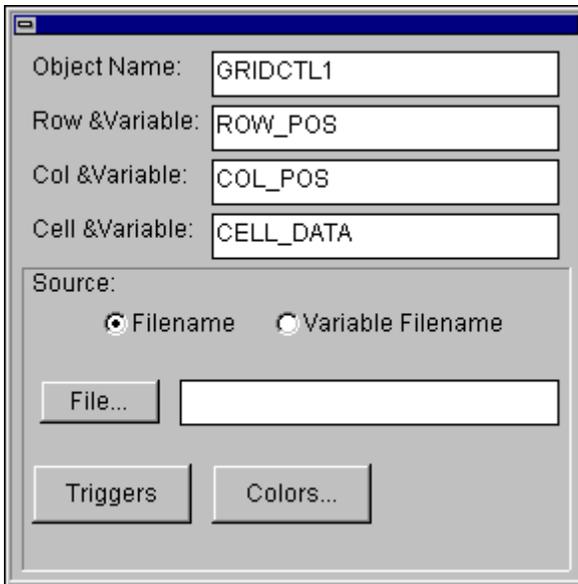


The grid box enables the user to select an item or coordinate from a grid that displays the values of a HOLD file.

To add a grid box, click the *Grid Box* button on the toolbox; then paint the grid box on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

The Properties dialog box for this form object is shown in the following example:

Click the screen items for their descriptions.



You can change the following properties of this object using the Properties dialog box.

Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Change the row variable	In the Grid Box Properties dialog box, enter the new variable name in the Row &Variable box.
Change the column variable	In the Grid Box Properties dialog box, enter the new variable name in the Col &Variable box.
Change triggers	In the Properties dialog box, click <i>Triggers</i> . The Triggers dialog box opens. For more information, see on page .

Task	Procedure
Specify a HOLD Master File	In the Grid Box Properties dialog box, <b>1.</b> Select the <i>Filename</i> radio button. <b>2.</b> Either: <ul style="list-style-type: none"> <li>• Enter the file name in the Filename box.</li> <li>• Click <i>File</i>; then select a file from the Open dialog box.</li> </ul>
Specify a variable HOLD Master File	In the Grid Box Properties dialog box, <b>1.</b> Select the <i>Variable Filename</i> radio button. <b>2.</b> Enter the variable file name in the Filename box.
Change grid colors	In the Grid Box Properties dialog box, click <i>Colors...</i> and select a new color.

## Form Box Tool

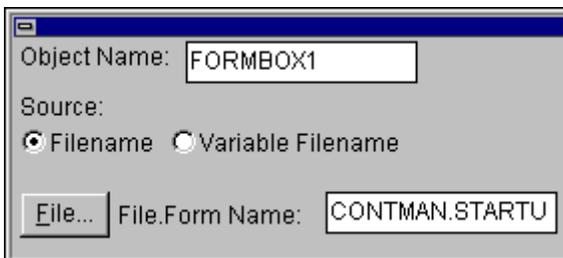


The form box tool enables you to embed a form within another form file in the current form. You can reuse a form you have already created without having to paint another form or recode triggers for objects.

To add a form box, click the *Form Box* button on the toolbox; then paint the form box on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

You can change object properties, including the name of the object and the source of the embedded form, using the Properties dialog box.

The Properties dialog box for form objects is shown in the following example:



If you wish to change the object name:

- Enter a new name in the Object Name box.

There are two ways to specify the source of an embedded form:

- By specifying an amper (&) variable.
- By selecting the form name using a dialog box.

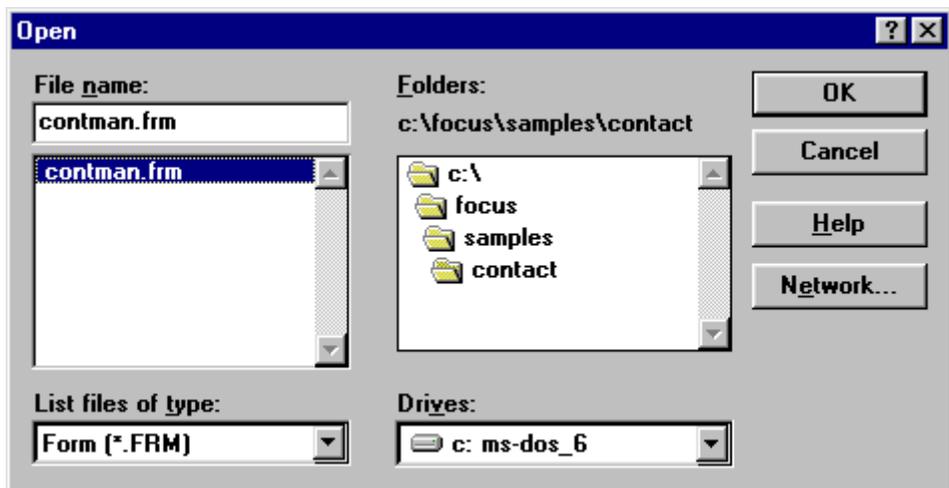
To select the form using an amper variable:

1. Open the Properties dialog box by selecting *Properties* from the Form menu, or double-clicking the form box on the open form.
2. Select the *Variable Filename* radio button and enter the variable name in the Amper Name box.

**Note:** The value for the variable you specify should contain the name of the embedded form file, followed by a period and the form name.

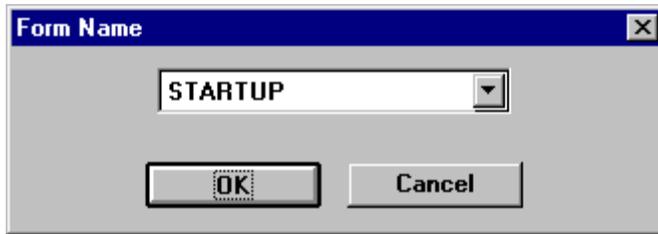
To select the form using the Filename dialog box:

1. Open the *Properties* dialog box by selecting *Properties* from the Form menu, or by double-clicking the form box on the open form.
2. Choose the *Filename* radio button.
3. Click the *File...* button. The Open dialog box opens:



Select the form you wish to embed, and click *OK*.

4. The Form name dialog box opens:



Enter the form name, or select it from the drop-down list, and click *OK*.

### **Procedure Working With Embedded Forms**

When you embed a form, you must include all of the forms that the embedded form calls, or references. If you do not, WebFOCUS returns an error at runtime. To include these forms, you must copy and paste the forms that the embedded form calls into the target form file:

1. Open the form file that contains the form you wish to embed.
2. Open the form that the form calls, and copy it to the Windows Clipboard:
  - Choose *Copy Form* from the Form menu.
3. Close the form file, and open the target form file (the form file in which you wish to embed the form).
4. Paste the form and rename it, if necessary.

The embedded form appears as a box on the form.

### **Group Box Tool**



A group box enables you to visually group objects. While the group box tool does not actually group items together, it separates them from the rest of the objects on your form, using a text label, a border, and optionally, color.

For example, you may wish to separate a group of objects from the rest of the form you are creating. The group box tool can make these objects stand out by grouping them together logically on the form.

To add a group box to the form:

1. Click the *Group Box* button on the Forms Painter toolbox.
2. Paint the group box on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

You can change the following properties of the object and the text label, using the Properties dialog box.



Task	Procedure
Change the object name	In the Properties dialog box, enter a new object name in the Object Name box.
Specify a title for the group box	In the Properties dialog box, enter a title in the Group Label box.

In addition, if you wish to change the color of a group box or the font of a label, click the *Options...* button in the Properties dialog box. See *Formatting Text* on page 19 for more information about specifying these attributes.

## VBX Custom Controls Tool

A VBX custom control allows you to add Visual Basic® Extension (VBX) custom controls to your form. Custom controls are often provided by third-party vendors, and allow you to add additional functionality to your forms. For example, the GRID.VBX control adds a spreadsheet-like grid to a form.

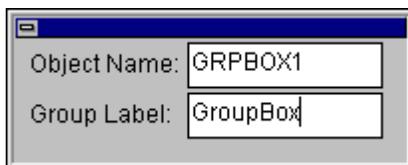
### Note:

- You must have a licensed VBX custom control to use the custom control with the Forms Painter.
- OCX files are not supported.

To use a VBX custom control with the Forms Painter, you must first add the custom control to the Forms Painter toolbox.

1. Choose *Manage VBX Libraries* from the File menu.

The Manage VBX Controls dialog box opens:



2. Select the file you wish to include, and click *Add*. You can add other VBX files by repeating these steps. Click *OK* when you are finished.

**Note:** The Forms Painter toolbox expands to display all of the loaded custom controls.

To add a custom control to the form:

1. Click the *Custom Control* button on the Forms Painter toolbox.

**Note:** The appearance of the VBX button on the toolbox varies according to the VBX custom control loaded.

2. Paint the custom control on the form. See *Specifying Form Objects* on page 10 for more information about painting objects.

## Displaying Forms With the WINFORM Command

---

You can display a form in WebFOCUS procedures (FOCEXECs) with the Dialogue Manager WINFORM control statement.

The syntax of the WINFORM command is

```
-WINFORM formname IN formfile
```

where:

**-WINFORM**

Is the Dialogue Manager control statement that displays a form. For details on Dialogue Manager, see your WebFOCUS 4.3 documentation.

*formname*

Is the name of the form to display. This is the form name as entered in the Form Name box in the New Form dialog box, and in the Name box in the Form Properties dialog box.

IN *formfile*

Specifies the form file containing the form being called by the WINFORM command. Form file names have the extension .FRM.

**Note:** You cannot embed the -WINFORM ... IN command in an inline FOCEXEC or other FOCEXEC called from a trigger event. You can only display a form within another form file with the Form Box tool. For more information on embedding forms, see *Form Box Tool* on page 62.

## Dynamically Altering a Form

---

You can dynamically alter your forms as they run. You can alter:

- Properties of the form window. For example, you can close, hide, or show the form, or change the title bar text.
- Properties of objects within the form.

To dynamically alter a form or form object, issue the command

```
-WINFORM form_name ALTER object_name property value
```

where:

*-WINFORM*

Is the Dialogue Manager control statement for referencing forms. For details on Dialogue Manager, see your WebFOCUS 4.3 documentation.

*form\_name*

Is the name of the form that is contained within the form file.

*ALTER*

Is the keyword specifying that the form is to be altered.

**Note:** This parameter cannot be used with objects that contain embedded spaces in the name.

*object\_name*

Is the name of the form object to be dynamically altered (such as PBTN1 for a push button). When altering the form itself, use the object name WINDOW. When altering menu items, use the menu item text as the object name.

**Note:** An embedded form's objects can be altered. In this case, specify the form name for the active (displayed) form window rather than the name of the embedded form.

*property*

Is the type of object property that will be altered. See the following list of valid properties.

*value*

Is the value to be assigned to the specified property.

The valid values for the property and value parameters are described in the following table. Note that position and size coordinates are specified in pixels.

Valid Object Types	Property	Value	Explanation
All except Form Box	SHOW	TRUE FALSE	Shows the form object if the value is TRUE; hides the form object if the value is FALSE.  <b>Note:</b> <ul style="list-style-type: none"> <li>You cannot hide Form Box objects, but you can hide the objects within them.</li> <li>You cannot hide an active form.</li> <li>Forms become active either explicitly (through the Perform Form trigger action) or implicitly (when the following form closes and control returns to the hidden form).</li> </ul>
All except Text Box and Picture Box	ENABLE	TRUE FALSE	Enables a push button or other object if the value is TRUE; disables (dims) a button or other object if value is FALSE.  <b>Note:</b> Disabled objects cannot get focus (accept mouse or keyboard input).
WINDOW, Menu Items, Check Box, Push Button	TEXT	<i>text</i>	Specifies text to display as an object label. The text can be modified in the window title bar and other objects.  <b>Note:</b> The WINDOW object type is used for altering the window title bar of the form.

Valid Object Types	Property	Value	Explanation
WINDOW	POPUP	TRUE FALSE	<p>Displays the form as the top window after an inline FOCEXEC trigger executes. The form becomes the active window. Note that WINDOW is the only valid object type for the POPUP property.</p> <p>TRUE is the default. When FALSE, the form does not pop-up on top of the screen after running the inline FOCEXEC or trigger.</p> <p><b>Note:</b> The POPUP property applies to the active form only. At the end of trigger execution, POPUP resets to TRUE when returning to the active form.</p>
All	LEFT	<i>x</i>	Positions the left edge of the object at <i>x</i> pixels from the left of the display for WINDOW, or <i>x</i> pixels from the left of the form for all other objects.
All	TOP	<i>y</i>	Positions the top edge of the object at <i>y</i> pixels from the top of the display for WINDOW, or <i>y</i> pixels from the top of the form for all other objects.
All	WIDTH	<i>w</i>	Resizes the object to the specified width in pixels.
All	HEIGHT	<i>h</i>	Resizes the object to the specified height in pixels.
All including WINDOW	MOVE	<i>x y</i>	Moves the top left corner of the object, in pixels, to the specified coordinates.
All including WINDOW	POSITION	<i>x y w h</i>	Repositions and resizes the object.

Valid Object Types	Property	Value	Explanation
WINDOW	CLOSE		Closes the form window.  <b>Note:</b> If a form is not at the end of the navigation path (see on page ), it is marked for closure and hidden. The form closes only when control is returned to a form earlier in the navigation path.

## Examples of Form Objects

The following table lists some examples of dynamically altered form objects.

Example	Function
-WINFORM MAIN ALTER PBTN1 SHOW FALSE	Hides the push button PBTN1.
-WINFORM MAIN ALTER PBTN1 ENABLE FALSE	Disables (dims) the push button PBTN1.
-WINFORM MAIN ALTER PBTN1 LEFT 200	Moves the push button PBTN1 to the left coordinate 200 pixels.
-WINFORM MAIN ALTER PICTUREBOX1 MOVE 225,500	Moves the picture box to the specified coordinates (225,500).
-WINFORM MAIN ALTER PICTUREBOX1 POSITION 225,300,375,500	Moves the picture box to the specified coordinates (225,300) and resizes the width (375) and height (500).
-WINFORM MAIN ALTER WINDOW CLOSE	Closes the form MAIN.
-WINFORM MAIN ALTER WINDOW TEXT 'Personnel Management System'	Changes the form title bar text to Personnel Management System.
-WINFORM MAIN ALTER 'File' TEXT 'Files'	Changes menu item File to Files.
-WINFORM MAIN ALTER WINDOW SHOW FALSE	Hides the form window.
-WINFORM MAIN ALTER WINDOW POPUP FALSE	Does not pop-up the form MAIN on top after an inline FOCEXEC or trigger runs.

## Getting the Form Values

You can obtain the value of a form object by issuing the command

```
-WINFORM form_name GET object_name property variable_name
```

where:

*-WINFORM*

Is the Dialogue Manager control statement for referencing forms.

*form\_name*

Is the name of the form contained within the form file.

*GET*

Is the keyword used to get the form value.

*object\_name*

Is the name of the form object that obtains a property value (such as PBTN1 for a push button). When obtaining a property value of the form itself, use the object name WINDOW. When obtaining a property value of menu items, use the menu item text as the object name.

*property*

Is the type of property that obtains a value. See the table earlier in this chapter for the list of valid properties for each object type.

*variable\_name*

Is the name of a variable that is assigned the value of the specified property. Do not include the & symbol in the variable name, unless you wish to assign a global variable. For details on Dialogue Manager variables, see your WebFOCUS 4.3 documentation.

## Creating Forms For Managed Reporting

If you are creating forms for use with WebFOCUS Managed Reporting applications, you will need to make some edits in the HTML file.

Below is the beginning of a form file saved in HTML format by the Forms Painter when you save a form as HTML:

```
<html>
<head>
<SCRIPT LANGUAGE="JavaScript">

// Manage reporting flag:  1 - Manage reporting is on.
//                          0 - Manage reporting is off.
var MR = 0;

// Please enter the domain name for IBIMR_domain.
// Example: domain = "domain2/domain2.htm"
domain = " "

// Please enter the folder name for IBIMR_folder.
// Example: folder = "#sfolder1"
folder = " "

.
.
.
```

Make the following edits in this file:

1. Set MR equal to 1 to turn on Managed Reporting.
2. Enter the domain name in the quotation marks following `domain =`.
3. Enter the folder name in the quotation marks following `folder =`.

If your form file has more than one form, you will have to repeat these edits for each form.

## Designing User Interfaces for WebFOCUS Tools

---

You can use the Forms Painter to design a custom user interface for WebFOCUS tools, such as the Report Painter. Thus you can design an application that allows users to create reports, graphs, and other objects.

The FOCUSHELL command language consists of commands that enable you to manage WebFOCUS objects and tools from a procedure or form.

The FOCShell commands are described in the following table:

FOCSHELL Command	Description
FOCSHELL WORKBENCH toolname EDIT	Invokes a tool to edit an existing object.
FOCSHELL WORKBENCH toolname NEW [file_description]	Invokes a tool to create a new object.
FOCSHELL WORKBENCH toolname RUN	Runs an existing object.
FOCSHELL WORKBENCH toolname SAVE focexec_name	Saves an object to a FOCEXEC.
FOCSHELL WORKBENCH toolname LOAD focexec_name	Invokes a FOCEXEC containing a single object. (FOCEXECs containing more than one object are not supported.)
FOCSHELL WORKBENCH LOCK	Disables the Workbench toolbar buttons (except for the Run button) when a tool is invoked.
FOCSHELL WORKBENCH UNLOCK	Enables the Workbench toolbar buttons that have previously been disabled.
FOCSHELL WORKBENCH toolname BUTTON {HIDE} {SHOW}	Hides or shows a Workbench toolbar button.
FOCSHELL SHELL SHOW {MIN} {MAX} {NORMAL}	Specifies the Workbench window display.
FOCSHELL SHELL VIEW {ADHOC} {APPLICATION} {MR}	Specifies the Workbench window view (operation) mode. For additional FOCShell commands that pertain to Managed Reporting, see on page .

<b>FOCSHELL Command</b>	<b>Description</b>
FOCSHELL FOCUS SHOW {MIN} {MAX} {NORMAL}	Specifies the WebFOCUS Session window display.
FOCSHELL STATBAR text	Writes text into the WebFOCUS Workbench status bar.
FOCSHELL APPLICATION OPEN application_name	Opens an application window in the Application Manager.  <b>Note:</b> This command is not available in runtime applications.
FOCSHELL APPLICATION ADD application_name file name	Adds a file to an application.
FOCSHELL APPLICATION SHOW {MIN} {MAX} {NORMAL}	Specifies the application window display.  <b>Note:</b> This command is not available in runtime applications.
FOCSHELL APPLICATION VIEW {ICONIC} {TABULAR}	Specifies how objects are displayed within an application.  <b>Note:</b> This command is not available in runtime applications.

The table below describes the valid values for the tool name parameter.

<b>Toolname</b>	<b>Tool</b>
REPORT (TABLE)	Report Assistant
RPAIN	Report Painter
GRAPH	Graph Assistant
FILEDEF	Filedef

Toolname	Tool
DEFINE	Define
JOIN	Join
SET	Set
TED	Text Editor
USE	Use
FILETOOL	Master File Editor

**Note:**

- TABLE is an optional name for the Report tool.
- FOCShell APPLICATION commands are not available in runtime applications.

**Syntax**

**Managed Reporting Commands**

The following commands are available for the FOCShell command language when using a version of WebFOCUS that supports Managed Reporting View:

FOCShell Command	Description
FOCShell SHELL VIEW MR	Changes the current view to the Managed Reporting View.
FOCShell DOMAIN OPEN	Opens the managed reporting domain that was last used.
FOCShell APPLICATION ADD application_name filename AS 'description'	Migrates an existing application to a Managed Reporting application, enabling the description to appear in the My Reports tab.
FOCShell DOMAIN OPEN 'domaindescription'	Opens the specified reporting domain. If the domain name contains spaces, it must be enclosed in single-quotes.
FOCShell DOMAIN NETWORKDIR {network_repository_name}	Determines the location of the repository on the LAN.
FOCShell DOMAIN SERVERNAME {name:port}	Controls the name and port for the WebFOCUS Reporting Server.

FOCSHELL Command	Description
<code>FOCSHELL DOMAIN STORELOCAL {ON OFF}</code>	For a WebFOCUS repository, controls whether My Reports are stored in a LAN (local) directory or on the Reporting Server.
<code>FOCSHELL DOMAIN OCADIR {directory_name}</code>	Controls the location of the My Reports directory.

For example, to switch to the Managed Reporting View and open a domain named Corporate Personnel Information System, the following syntax could be used in a WebFOCUS procedure:

```
FOCSHELL SHELL VIEW MR
FOCSHELL DOMAIN OPEN 'Corporate Personnel Information System'
```

## Using the FOCShell Command Language

The FOCShell commands enable you to design a form from which a user can access one or more tools, such as the Report Painter or Graph Assistant.

You can include FOCShell commands in a form as an inline procedure (FOCEXEC) when specifying the trigger action for a form object. For example, to add a push button to a form that will enable a user to create a report:

1. Add a push button.
2. In the Triggers dialog box for the push button, enter the following inline FOCEXEC:

```
FOCSHELL WORKBENCH LOCK
FOCSHELL WORKBENCH REPORT NEW
FOCSHELL WORKBENCH REPORT SAVE MYREPORT.FEX
```

## Migrating Existing WINFORMs to Web Forms

---

If you are migrating your application to WebFOCUS, we suggest that you migrate your Winforms to Webforms and edit them in FrontPage. Using this method you can convert both the presentation and process layers of your WINFORMs. You would not use the Forms Painter again once you migrate a Winform to a Webform.

The migration process is as follows:

- Migrate the Winform to a Webform.
- Include any other components and procedures in your application.
- Debug your application.
- Deploy to the Web.

For details see Chapter 3, *Designing a User Interface for a Web Application*.

## **Exiting the Forms Painter**

---

To exit the Forms Painter, select *Exit* from the Forms Painter's File menu. You are prompted to save your changes.

If your form is covering the menu bar, press Alt+F4 to exit the Forms Painter.



---

---

# Index

value 21, 23

## A

---

Action drop-down combo box 21, 23

ampersand variables 2

applications 2

    executing procedures within 2

## C

---

check boxes 48

combo boxes 45

control menus 18

    displaying 18

    hiding 18

cursors 19

    moving between control objects 19

## E

---

entry fields 34

    adding to forms 34

## F

---

FOCEXECs 21, 23

FOCSHELL command 72, 75 to 76

form objects 2, 10 to 11

    adding 10 to 11

    aligning 10

    aligning groups 13 to 14

    boxes 62

    cutting and pasting 10, 12

    deleting 10, 12

    editing 70

    fonts 19

    formatting text in 19

    hiding 67

form objects 2, 10 to 11 (*continued*)

    moving 10, 13

    properties 33

    resizing 10, 15

form windows 17

    adding menus to 15

    customizing 9

    keyboard access characters 15, 17

    movable 15

    moving 18

    properties 17

    resizing 18

    specifying titles for 17

Formatting toolbar 6 to 7, 19

forms 1

    adding to existing form files 8

    closing 4

    displaying 66

    editing 8 to 9, 67, 72

    embedded 62, 64

    entry fields 34

    examples 25

    FOCSHELL command in 76

    form box tool 62

    List Boxes 37

    minimizing 18

    MultiSelect option 40

    navigating 4, 67

    radio buttons in 26, 28 to 29

    renaming 9

    saving as HTML 72

    specifying parameters in 33

    specifying triggers in 21 to 23, 33

    storing 5

    tools 72

    user interfaces 72

    using WINFORM command with 66

    values of 71

## Index

forms 1 (*continued*)  
variables and 24

Forms Painter 6 to 7, 72, 77  
editing forms from 8  
exiting 77  
Formatting toolbar 6 to 7  
Menu bar 6  
rulers 6 to 7  
Status bar 6 to 7  
Title bar 6

## G

---

GET command 71

graphs 57  
creating 72  
output display boxes 57

grid boxes 60

group boxes 64

## H

---

hyper regions 59

## K

---

keyboard access characters 15, 17, 52 to 53

## L

---

List Boxes 37, 40

List buttons 41

## M

---

Menu bar 6

menus 15  
adding separator bars to 15  
adding to form windows 15

MultiSelect option 40

## O

---

output display boxes 57, 59

## P

---

path type hierarchy 4

picture boxes 54

picture buttons 55

procedure AREA 26, 32

procedure RETAILP 26, 32

procedures 21, 23

Properties dialog box 19, 33

push buttons 52

## R

---

radio button list 26, 28 to 29, 43

reports 57  
creating 72  
drill-down 57, 59  
output display boxes 57, 59

rulers 6 to 7

Run push button 26, 30

## S

---

separator bar 15

Status bar 6 to 7

## T

---

text 19

text boxes 49

Title bar 6

toolbox 10 to 11

triggers 19 to 21  
actions 21, 23

- triggers 19 to 21 (*continued*)
  - default 21
  - editing 21
  - events 21 to 22
  - specifying in forms 21 to 23

## **U**

---

- user interfaces 72

## **V**

---

- variables 24
  - assigning value to 24
  - customizing forms with 24
  - reading 24
- VBX (Visual Basic Extension) custom controls 65
- Visual Basic Extension(VBX) custom controls 65

## **W**

---

- WINFORM command 66 to 67, 71
- Winforms 5, 76
  - creating in procedures 5
  - editing 5 to 6
  - migrating to Webforms 76
  - saving 9

