Chapter 15
Adding Help to Your Applications

Objective

This lesson shows you how to implement an online help system for your CA-Visual Objects applications.

Overview

An online help system can mean the difference between the success and failure of your application. Applications are often judged on the existence and quality of their online help.

The creation of an online help system is non-trivial and its design should be included at the start of your application’s development.

When designing an online help system, you must decide upon the level of detail you want to provide, and how best to organize the hierarchy of the help topics. Once the system is designed, you must provide the written text and build the help system.

Using the Windows help system offers several advantages to you, as it is installed with every copy of Microsoft Windows 3.x. First of all, it automatically loads the help system within your application when context-sensitive help is requested. Additionally, users can be expected to have some familiarity with its operation, because most Windows applications provide an online help system.

Context-Sensitive Help

CA-Visual Objects not only provides the capability for attaching help to your application, but also provides the mechanism to invoke *context-sensitive* help. Context-sensitive help is based on the current state of the application, or *context*, in which the user requests help.

F1

Pressing the F1 key invokes help based on the menu command or control that currently has focus. For example, to obtain help on a menu command or control, the user need only highlight it and press F1.

If no control or menu command has focus when F1 is pressed, the default behavior is to display the Contents topic defined in the associated help file.

Shift+F1

Pressing Shift+F1 allows the user to select an item using a special help cursor that consists of an arrow and a question mark. To get help on a control or menu command, the user presses Shift+F1, then locates and clicks on the control or menu command with the help cursor.

If no help is available for the item selected, the default behavior is to display the Contents topic defined in the associated help file.

Exercise

During this exercise, you will discover how the South Seas Adventures help system is implemented. You will also learn where in your application to specify the associated help file, and how to set context-sensitive help—for a window, a control in a window, and a menu command.

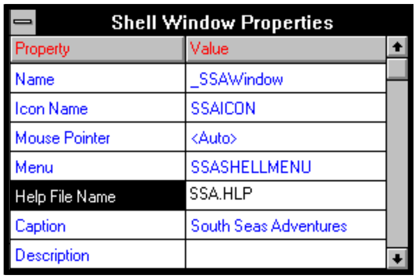
Implementing Context-Sensitive Help

Typically, an application has only one help file for the entire application. When you assign a help file to a shell window, each of its child windows, unless another help file is specified, uses the same help file as its parent. However, the architecture used by CA-Visual Objects makes it very easy to define a separate help file for each window.

Attaching Your Help File

Let's see how attaching a help file is accomplished in the South Seas Adventures application:

1. Double-click on the South Seas Adventures application button in the Application Browser.
2. Open the SSA Shell:Forms module by double-clicking on its button in the Module Browser.
3. Find the _SSAWindow window entity, and double-click on it to open the Window Editor.
4. Find the Help File Name property in the Shell Window Properties window:



Shell Window Properties

Property	Value
Name	_SSAWindow
Icon Name	SSAICON
Mouse Pointer	<Auto>
Menu	SSASHELLMENU
Help File Name	SSA.HLP
Caption	South Seas Adventures
Description	

By assigning the Help File Name property to SSA.HLP the following line of code is generated in the Init() method of the _SSAWindow:

```
SELF:HelpDisplay := HelpDisplay{"SSA.HLP"}
```

The HelpDisplay class in this code establishes a link between the South Seas Adventures shell window (_SSAWindow) and the SSA.HLP help file.

5. Close the Window Editor by double-clicking on its system menu.
6. Close the Entity Browser for SSA Shell:Forms by double-clicking on its system menu.

HelpRequest Event System

When help is requested by the user, a HelpRequest event is generated. The CA-Visual Objects dispatcher then invokes the window's HelpRequest() method, passing it a HelpRequestEvent object. The HelpRequestEvent class is used to describe the context, and item combinations, for which help is requested.

The default HelpRequest() method (from the Window class) invokes help, based on the information in the HelpRequestEvent object:

```
METHOD HelpRequest(oHRE) CLASS Window
        ...
        SELF:HelpDisplay:Show(oHRE:HelpContext)
        ...
        RETURN NIL
```

The HelpDisplay:Show() method invokes WinHelp, with the *oHRE:HelpContext* parameter representing the keyword that WinHelp searches for in the help file.

Help Context Property

You can specify Help Context strings for windows, menu commands, controls, and field specs. The Help Context property is the means by which CA-Visual Objects enables your application to supply context-sensitive help. CA-Visual Objects extracts the Help Context property of the item (such as control or menu command) for which the user requests help. This Help Context is then used in the call to WinHelp.

WinHelp performs a keyword search for a topic that matches the Help Context string. If a match is found, the topic is displayed. If you do not specify a Help Context property for the specified item, or if the

Help Context specified is not found, WinHelp displays the Contents topic of your help file.

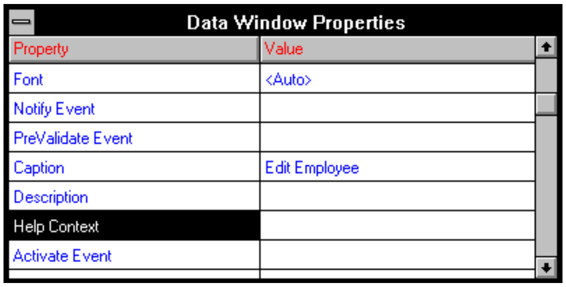
Help for a Window

The South Seas Adventures help file has a topic, called Employees, which describes editing employee data. If the user requests help when an Edit Employee window is open, you want WinHelp to display this topic. The following steps show how this is accomplished.

1. Open the Employee:Forms module by double-clicking on its button in the Module Browser.

2. Locate and double-click on the EditEmployeeWindow entity to open the Window Editor.

3. Scroll through the Data Window Properties window until you see the Help Context property:



Data Window Properties

Property	Value
Font	<Auto>
Notify Event	
PreValidate Event	
Caption	Edit Employee
Description	
Help Context	
Activate Event	

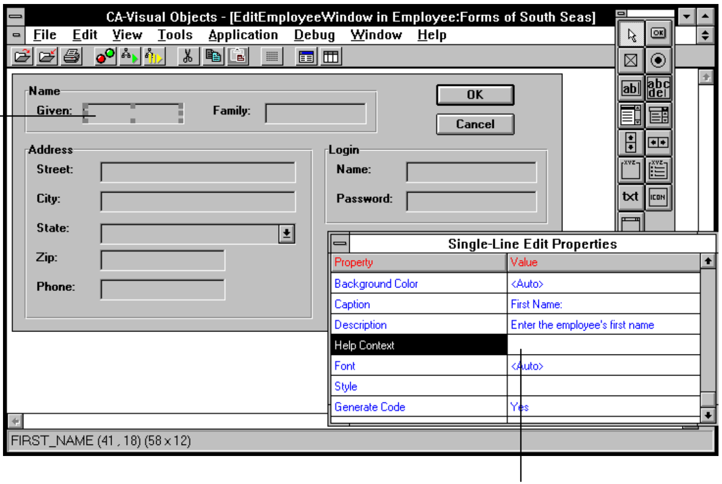
4. Set the Help Context property to Employees.

   When help for this window is requested, WinHelp now displays the Employees topic.

Help for Controls

The South Seas Adventures help file has another topic, called Employee Controls, which describes all of the controls in the EditEmployee window. When help is requested for any of these controls, the following steps enable WinHelp to display this topic:

1. Select the First_Name single-line edit control in the upper-left corner of the EditEmployee window:



Property	Value
Background Color	<Auto>
Caption	First Name:
Description	Enter the employee's first name
Help Context	
Font	<Auto>
Style	
Generate Code	Yes

FIRST_NAME (41 , 18) (58 x 12)

First_Name edit control

Click here to enter Help Context property

2. Select the Help Context property from the Single-Line Edit Properties window. Enter the text **Employee Controls** as the Help context; make sure to leave a space between the two words.

3. The Help Contexts for the other fields have already been supplied with the same value, so choose the Save toolbar button to save your changes.

   A Warning Box could appear because you have imported this window as part of the SSA.AEF file and it has not previously been edited and saved.

4. Choose Yes to save.

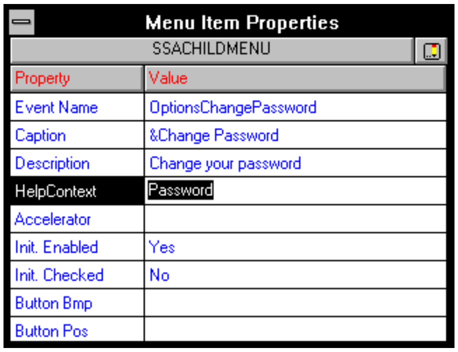
5. Close the Window Editor by double-clicking on its system menu.
6. Close the Entity Browser for Employee:Forms by double-clicking on its system menu.

Help for a Menu Command

The South Seas Adventures help file has another topic, called Password, which describes how to change the current user’s password. The following steps show how this help topic is associated with the Change Password command in the Options menu:

1. Open the SSA Child:Menu module by double-clicking on its button in the Module Browser.
2. Find the SSAChildMenu menu entity and double-click on it to open the Menu Editor.
3. Scroll through the Menu Editor and click on the Change Password menu item under the Options menu.

   The Help Context property for this item is already set to Password, as shown in the Menu Item Properties window:



Property	Value
Event Name	OptionsChangePassword
Caption	&Change Password
Description	Change your password
HelpContext	Password
Accelerator	
Init. Enabled	Yes
Init. Checked	No
Button Bmp	
Button Pos	

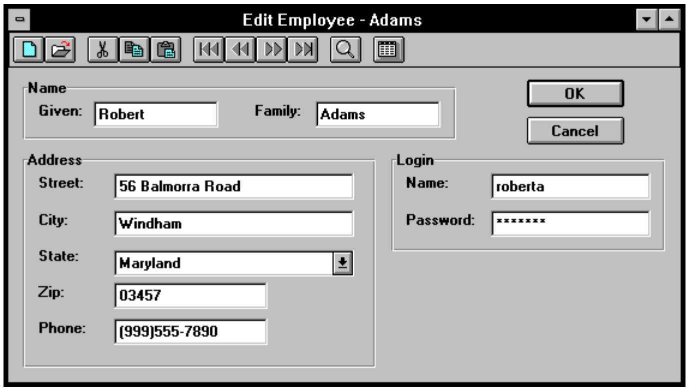
4. Close the Menu Editor by double-clicking on its system menu.
5. Close the Entity Browser for SSA Child:Menu by double-clicking on its system menu.

Invoking Context-Sensitive Help

Now, let’s test the context-sensitive help for your application.

1. Choose the Build toolbar button.
2. Choose the Execute toolbar button.
3. Log in to the application as usual (Name: **User**, Password: **Trainee**).
4. Select the Open command from the File menu.
5. Click on the Employee radio button, and choose OK.
6. From the Employee Browser, choose Edit.

   The Edit Employee window is displayed:



Edit Employee - Adams

Name

Given: Robert

Family: Adams

Address

Street: 56 Balmorra Road

City: Windham

State: Maryland

Zip: 03457

Phone: (999)555-7890

Login

Name: roberta

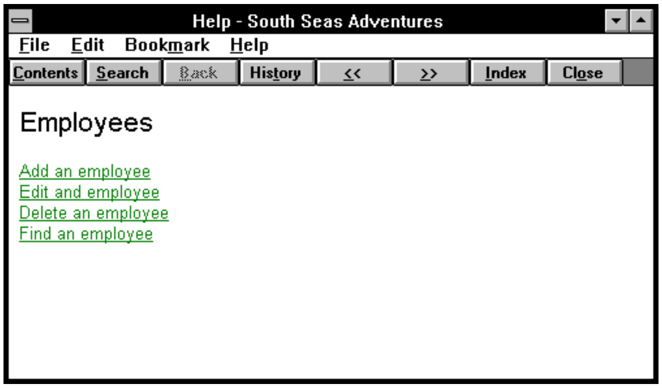
Password: *******

Help for the Window

This section shows you how to obtain help for the entire Edit Employee window.

1. Press Shift+F1 to display the help cursor.
2. Click on the Edit Employee window’s title bar.

   The South Seas Adventures Help window appears, displaying the Employees topic:

   Employees

   Add an employee
   Edit and employee
   Delete an employee
   Find an employee

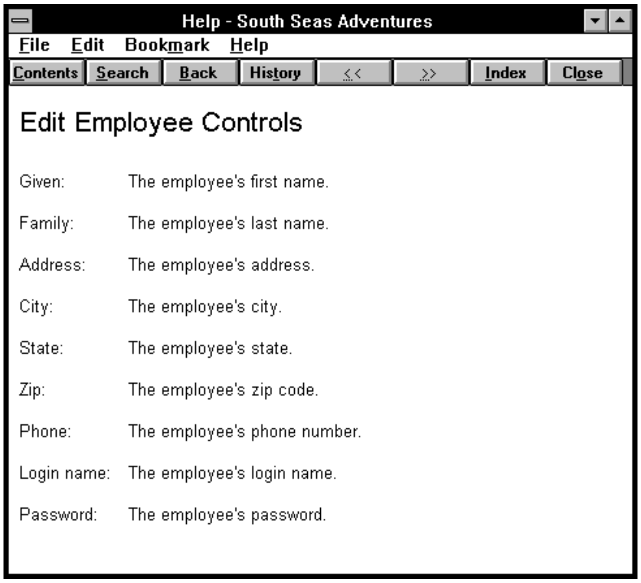
3. When you are finished reviewing the help text, close the Help window by double-clicking on its system menu.

Help for a Control

This section shows you how to obtain help for a control within a window.

1. With the Given Name edit control in focus, press F1.

   The South Seas Adventures Help window appears, displaying the Edit Employee Controls topic:



Edit Employee Controls

Given:	The employee's first name.
Family:	The employee's last name.
Address:	The employee's address.
City:	The employee's city.
State:	The employee's state.
Zip:	The employee's zip code.
Phone:	The employee's phone number.
Login name:	The employee's login name.
Password:	The employee's password.

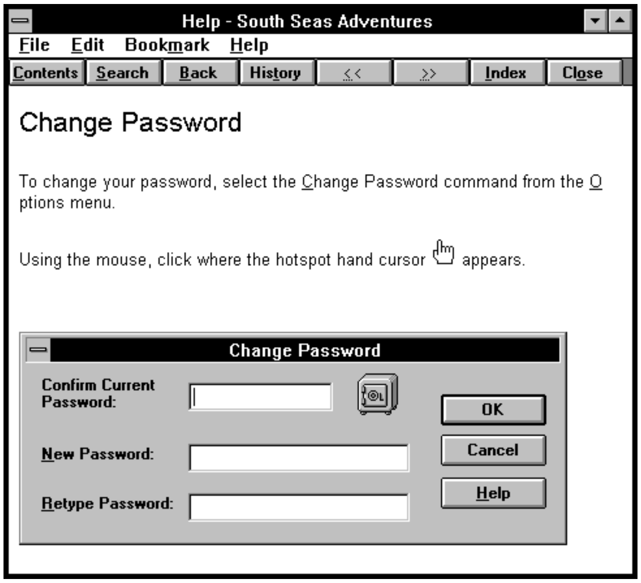
2. When you are finished reviewing the help text, close the Help window by double-clicking on its system menu.
3. Choose Cancel to close the Edit Employee window.
4. Choose Close to close the Employee Browser.

Help for a Menu Command

To get help for a menu command:

1. Press Shift+F1 to display the help cursor.
2. Using the help cursor, choose the Change Password command from the Options menu.

   The South Seas Adventures Help window appears, displaying the Change Password topic:



Change Password

To change your password, select the Change Password command from the O ptions menu.

Using the mouse, click where the hotspot hand cursor appears.

3. When you are finished reviewing the help text, close the Help window by double-clicking on its system menu.

Implementing Direct Calls to Help

Many Windows applications provide their users with help from other sources. Some of these sources are menu commands, button controls, and event processes.

Menu Commands

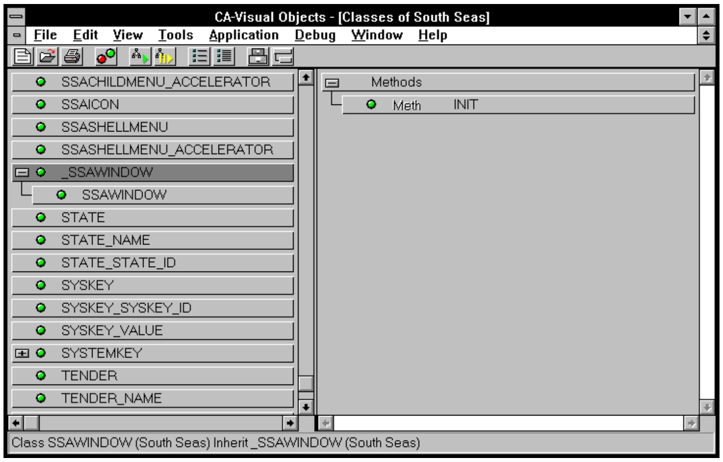
Help can be invoked in response to a menu command selection. To implement help from a menu command you can:

- Trap menu selection events, using the MenuSelect() event handler method
- Create a separate menu event method for the menu command

This section show how the South Seas Adventures application creates a separate menu event method for a menu command.

1. Choose the Class Browser command from the Tools menu to open the South Seas Adventures Class Browser.
2. Locate the _SSAWindow class in the left pane of the browser, and click on the Expand button to its left.

   The SSAWindow class appears below _SSAWindow in a tree-like structure:



Class SSAWINDOW (South Seas) Inherit _SSAWINDOW (South Seas)

3. Click on the SSAWindow class.

4. Find the SSAWindow:HelpContents() method in the right pane of the Class Browser, and double-click on it to open the Source Code Editor. The following code is displayed:

```
METHOD HelpContents() CLASS SSAWindow

    SELF:HelpDisplay:Show("HelpIndex")

    RETURN SELF
```

This method illustrates how to call the help system directly using the HelpDisplay:Show() method. You can pass a help context keyword as the parameter for HelpDisplay:Show() function or, as shown in this example, one of the following reserved keywords to access a standard help feature.

Help Context	**Help**
HelpIndex	Displays the Contents topic as defined in the application help file
HelpOnHelp	Displays the Contents topic as defined in your Windows help file (normally WINHELP.HLP, in which the Contents topic is How to Use Help)

5. Close the Source Code Editor by double-clicking on its system menu.

Push Button Controls

Calling help in response to a push button click is implemented in the same manner as menu commands. In the South Seas Adventures application, this is illustrated using the Help push button on the Change Password dialog box:

1. Locate the _NewPasswordDialog class in the left pane of the Class Browser, and click on the Expand button to its left.

   The NewPasswordDialog class appears below _NewPasswordDialog in a tree-like structure.

2. Click on the NewPasswordDialog class.

3. Find the NewPasswordDialog:HelpButton() method in the right pane of the Class Browser, and double-click on it to open the Source Code Editor.

   In this case, HelpDisplay:Show() is called using the Password help context keyword:

   ```
   METHOD HelpButton() CLASS NewPasswordDialog

       SELF:HelpDisplay:Show("Password")

       RETURN SELF
   ```

   Therefore, the same help topic associated with the Options Change Password menu command is displayed when the user clicks on the Help button of the Change Password dialog box.

4. Close the Source Code Editor by double-clicking on its system menu.

5. Close the Class Browser by double-clicking on its system menu.

Event Processing by Name

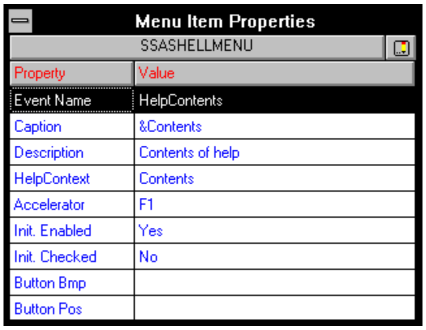
The NewPasswordDialog:HelpButton() method and the SSAWindow:HelpContents() method are linked to their respective controls by name. In the case of the Help Contents menu command, the method name to call is defined as part of the SSAShellMenu menu entity. Let's see how this is accomplished:

1. Double-click on the SSA Shell:Menu module button to open its Entity Browser.

2. Double-click on the SSAShellMenu menu entity to open it in the Menu Editor.

3. Scroll through the Menu Editor and select the Contents menu item under the Help menu.

   The Properties window is updated to display the properties for the selected menu command. It is the Event Name property that determines the name of the method to invoke when this menu command is chosen. In this case, it is set to HelpContents, as expected:



Menu Item Properties

SSASHELLMENU

Property	Value
Event Name	HelpContents
Caption	&Contents
Description	Contents of help
HelpContext	Contents
Accelerator	F1
Init. Enabled	Yes
Init. Checked	No
Button Bmp	
Button Pos	

4. Close the Menu Editor by double-clicking on its system menu.

5. Close the Entity Browser by double-clicking on its system menu.

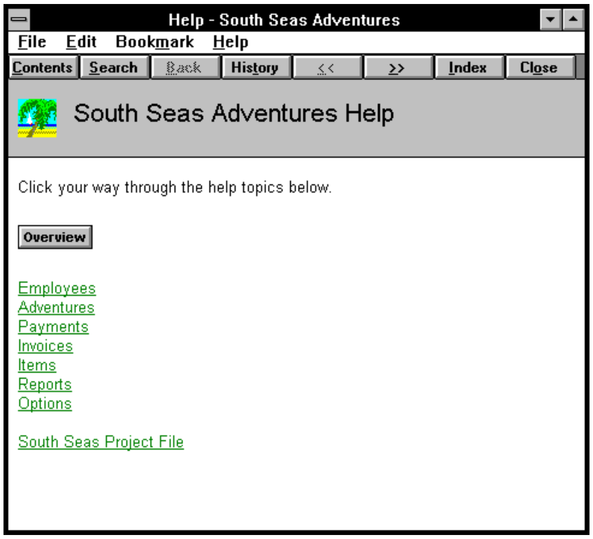
If you follow steps similar to those above to load the _NewPasswordDialog window entity (located in the Password:Forms module) in the Window Editor, you will see that the Event Name property for the bottom-right push button is set to HelpButton.

Invoking Help Using Menu Commands and Push Buttons

Now, let’s run the South Seas Adventures application again to see the menu command and push button help calls in action:

1. Choose the Execute toolbar button to run South Seas Adventures.
2. Log in to the application as usual (Name: **User**, Password: **Trainee**).
3. Choose the Contents command from the Help menu.

   The South Seas Adventures Help window appears, displaying the main table of contents:



Help - South Seas Adventures
File Edit Bookmark Help
Contents Search Back History << >> Index Close
Overview

South Seas Adventures Help

Click your way through the help topics below.

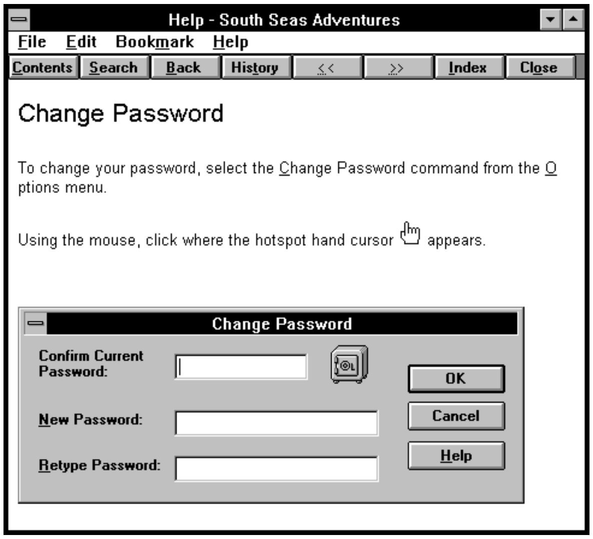
Employees
Adventures
Payments
Invoices
Items
Reports
Options

South Seas Project File

4. When you are finished reviewing the help text, close the Help window by double-clicking on its system menu.
5. Choose the Change Password command from the Options menu.

6. When the Change Password dialog box is displayed, choose the Help push button.

   The South Seas Adventures Help window appears, displaying the Change Password topic:



Help - South Seas Adventures
File Edit Bookmark Help
Contents Search Back History << >> Index Close
Change Password
Confirm Current Password:
New Password:
Retype Password:
OK
Cancel
Help

Change Password

To change your password, select the Change Password command from the O ptions menu.

Using the mouse, click where the hotspot hand cursor appears.

7. When you are finished reviewing the help text, close the Help window by double-clicking on its system menu.
8. Choose Cancel to close the Change Password dialog box.
9. Close the South Seas Adventures application by double-clicking on its system menu, and answering Yes when prompted.

Creating Help Files

Creating Windows help support for your application is a three-step process:

1. Create the required help system source files (such as topic and project files).
2. Compile the source files, using the HC31.EXE help compiler that comes with CA-Visual Objects. This creates a help file that is ready to be used by the WINHELP.EXE program.
3. Create the links in your application that use the help system.

The complete help subsystem, provided with the South Seas Adventures application, resides in your CA-Visual Objects SAMPLES\SSATUTOR\HELP subdirectory. If you are interested in how the help system is put together, you can study the files described in this section in more detail.

Topic Files

A topic file contains the text of your help file for one or more topics. The topic file also contains the codes needed to link topics together.

The help compiler requires topic files coded in Rich Text Format (RTF). To create a file in this format, you can use a simple ASCII text editor, or you can use a word processor that can export files in this format.

Using a simple text editor requires you to explicitly code your help topics using the rich text coding syntax. On the other hand, using a word processor that can export in RTF format can reduce your task dramatically. Using simple formatting commands and footnotes to create the document, you then export it in RTF format. The document is translated into the appropriate commands and codes, that can then be compiled by the Help compiler.

For more information on creating the help source files, see the CA-Visual Objects online help for Creating Help Files.

The South Seas Adventures topic file, SSA.DOC, was created using Microsoft Word for Windows. The file was then exported to rich text format as SSA.RTF. You can retrieve the SSA.RTF file with your program editor to view the RTF codes.

Project File

The project file contains a list of source files required to create the help file. It also contains window definitions and compiler directives.

The South Seas Adventures help project is stored in the SSA.HPJ file.

Graphic Files

All other file types allowed are graphic files, and are optional. These include the following:

- **.BMP –** Contains a single graphic in the Windows bitmap format.
- **.WMF –** Contains a single graphic in the Windows metafile format.
- **.SHG –** Graphic file produced by SHED.EXE (Hotspot Editor). The Hotspot Editor allows you to create links to help topics from a selected area of a graphic image, and is supplied with CA-Visual Objects.
- **.MRB:** Contains more than one version of the same bitmap at different screen resolutions. This is a multiple resolution bitmap file produced by the MRBC.EXE.

Summary

In this lesson you have implemented online help for a CA-Visual Objects application and, through the Help Context property, you now know how to implement context-sensitive help. You have also seen how to make calls directly to the Windows Help system, and are now familiar with the files required to create a help file.

In the next lesson, you will use the Windows API functions in a CA-Visual Objects program to provide functionality not found in the native CA-Visual Objects language classes.