

Chapter 13

Reporting with CA-RET

Objective

In this lesson, you will learn how to define effective reports using CA-RET, the Computer Associates Report Engine Technology.

Overview

CA-RET is a powerful reporting facility that offers ease of use through its intuitive user interface. Using CA-RET, you will learn how to define, customize, preview, and print a report.

Exercise

During this exercise, you are going to define the Customer List report. As you develop this report, you will:

- Gain a working knowledge of CA-RET's menus and toolbars
- Learn how to create report headers and footers
- Provide your own titles and pictures
- Insert, delete, and sort report data fields
- Use the CA-RET help facility
- Learn how to pass parameters to your CA-RET report

Starting the Report Editor

In this exercise, you will create a report that works on any user's machine, independent of the installation drive and directory. This is accomplished by using data servers that do not contain any path information for data and index files. For more information, refer to Appendix A, "Creating a Path-Independent Application".

1. Open the South Seas Adventures Module Browser by double-clicking on its button in the Application Browser.
2. Select the Customer:Reports module by clicking on its button in the Module Browser.
3. Choose the Open Entity toolbar button, and select Report Editor from the local pop-up menu that appears.



The CA-Visual Objects Report Editor dialog box displays:

CA-VO Report Editor

Report Definition file name:

Data server:

DB server

SQL server

Report style:

Tabular Letter

Form Free style

Labels

Available DBF entities:

- Advdtl
- Advhdr
- Customer
- Dup_customer
- Employee
- Invdtl
- Invhdr

Available SQL entities:

OK

Cancel

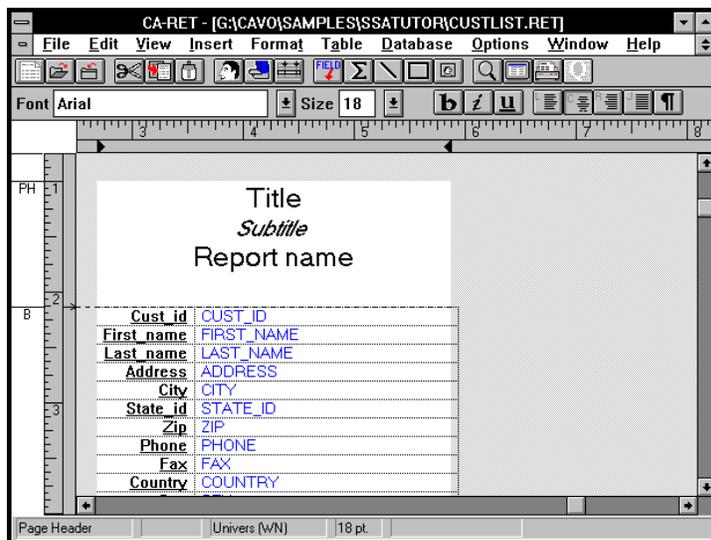
4. Type **CustList** in the Report Definition File Name edit control.
The name you enter here is used by the Report Editor to generate the CustList class that is used to run the report from your application.
5. The CustList report will use data from the CUSTOMER.DBF file, for which we have already defined a data server (see the "Working with Data Servers" chapter). Therefore, select Customer from the Available DBF Entities list box.

6. Because the Available DBF Entities list box allows multiple selections, you must also deselect Advdtl.
7. Click on the Form radio button in the Report Style group box.

When you create a new report, the Report Editor allows you to choose from various default report styles. Based on the style that you select, CA-RET generates a default report.

8. Choose OK.

CA-RET will now be launched. With just a few selections, you have a predefined report—with titles, labels, and data.



9. Preview the report using the Print Preview toolbar button.
10. When you are finished previewing the report, select Close from the File menu.

A Quick Tour

At first, CA-RET may remind you of your favorite word processor. That's because it uses much of the same functionality you have seen in word processors. If you want text on the report, just type it in. To format your report, you will use CA-RET's font toolbar and also rely heavily on tables, both of which are available in most word processors.

The main report editing area is known as the Report Definition. Two rulers, one along each axis of the report page, allow you to see where you are in your report at all times.

The *Section Name* window, along the left side of the report definition area, displays letters identifying each section of your report: PH (Page Header), B (Body), and PF (Page Footer).

Adding Your Personal Touch

CA-RET has created a page header with the default text "Title," "Subtitle," and "Report Name." Let's customize this report header to better meet our needs:

1. Change the title by highlighting Title and typing **Customer List**.
2. Change the subtitle by highlighting Subtitle and typing **by Customer #**.
3. Highlight Report Name, and type **as of** with a space character after it.

You will next add a field to this line so that it displays the current date.

Adding Fields

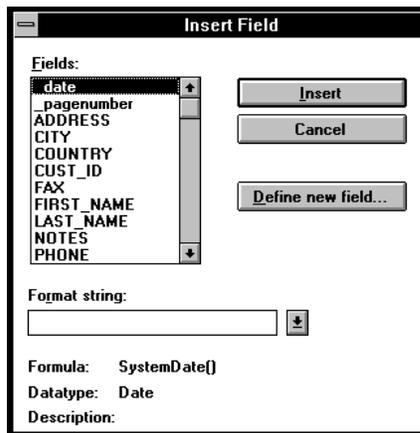
CA-RET allows you to add fields to your report. Fields can be the following types: database, computed, parameter, system-defined, or user-defined.

Next, add the system date as part of the header:

1. If your cursor is not already positioned after the “as of” text that you just entered, position it there.
2. Choose the Insert Field toolbar button.



The Insert Field dialog box appears:

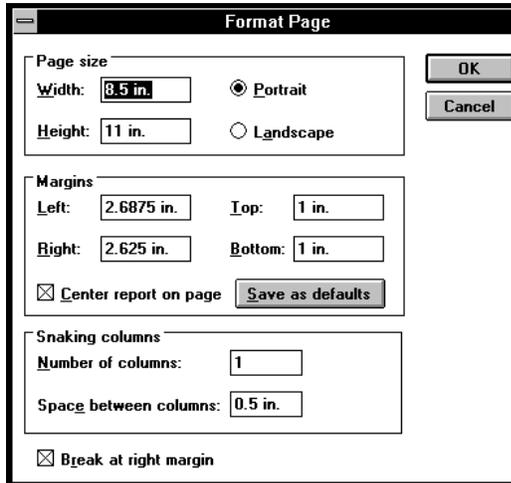


3. From the Fields list box, select `_date` and choose Insert.

The `_date` field is a system-defined computed field that returns the current date. It now appears as a place holder in your report name (as of `_date`).

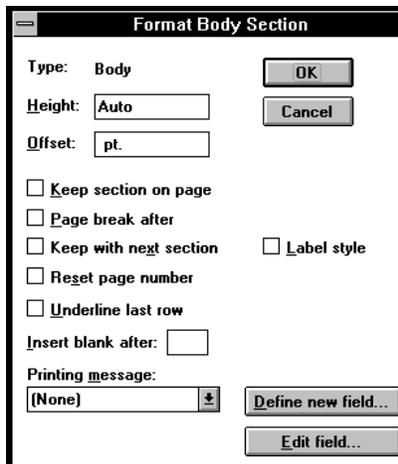
4. Select the Page command from the Format menu to format the margins of the report.

The Format Page dialog box appears:



5. In the Margins group box, adjust both the Left and Right margins to 1.5 inches, and then choose OK.
6. Click at any point on the body section of the report (anything below the header section), and then select the Section command from the Format menu.

The Format Body Section dialog box appears:

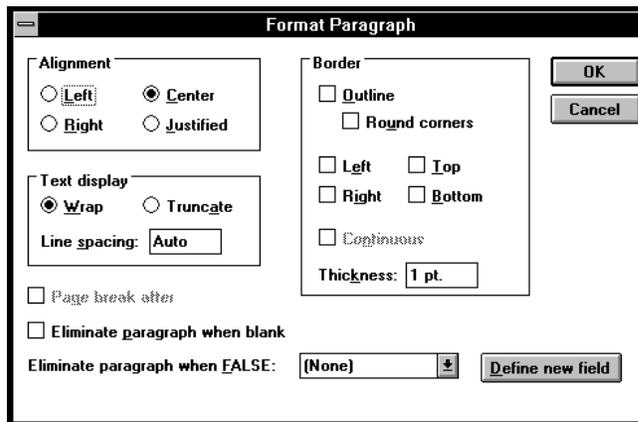


7. Select the Keep Section on Page check box in order to prevent the section from being split across two pages, and then choose OK.

Now, let's format the page header and footer:

1. Click on the last line of the page header section, and then select the Paragraph command from the Format menu.

The Format Paragraph dialog box appears:



2. To underline each page header, select the Bottom check box in the Border group box, and then choose OK.

Note: Each line in the page header section has a distinct paragraph setting. The change that you just made will apply only to the last line in the header, since that line was selected when you chose the Format Paragraph command. To apply the same paragraph setting to all lines in the header, highlight all the lines before choosing the Format Paragraph command.

The next few steps place the company information in the page footer.

3. Place your cursor on the first line of the Page Footer section and type the following, pressing Enter after each line:

**South Seas Inc.
1234 Seashore Drive
Montego Bay, Jamaica**

4. Double-click on the first line of the Page Footer.
The Format Paragraph dialog box appears again.
5. To draw a line above each page footer, check the Top check box in the Border group box, and then choose OK.
6. By now, you probably want to see what your report looks like, so select the Print Preview toolbar button.



In preview mode, CA-RET provides you with additional toolbar buttons that allow you to perform the following actions:

Button	Action
	Toggle back and forth between page and normal view
	Center the report on the screen
	View the next page of the report
	View the first page of the report
	View the last page of the report
	Print the report

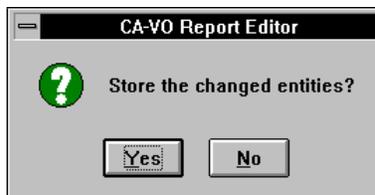
7. Once you are done previewing and/or printing your report, choose Cancel to return to the design mode.

Saving Your Work



1. Save your report by choosing the Save toolbar button. If you do not save the report before exiting, CA-RET will remind you to save your changes.
2. Close CA-RET by selecting the Exit command from the File menu.

The CA-Visual Objects Report Editor prompts you to store the changed entities, as follows:



3. Choose Yes to save the changed entities. The Report Editor then generates the CustList class in the Customer:Reports module.

Running Your Report Within Your Application

To run a report from within your application, open the Source Code Editor for the Init() method (of the CustList class) in the Customer:Reports module. The code, which was generated when you saved the report design, is shown below (SELF refers to the report's owner window).

```
oReport := CustList{SELF}
```

Instantiating the CustList class will cause a preview of the report to be displayed. That is because the generated CustList:Init() method has the following code by default:

```
METHOD Init(oOwner) CLASS CustList
    SUPER:Init(oOwner,"caretrun")
    SELF:Open(CustList_File)

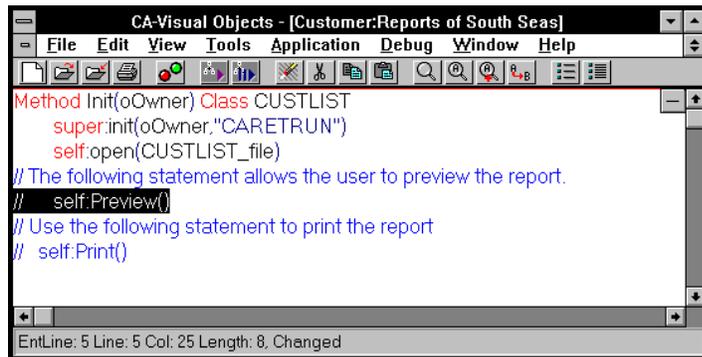
// The following statement allows the user to
// preview the report.
    SELF:Preview()
// Use the following statement to print the report
//    SELF:Print()
```

Note: You can also use CustList as a menu item's event name or as the name of a push button. When the menu item or push button is selected, the report will run.

For the purposes of this report, however, you will be passing a parameter later in this exercise. To accommodate the use of some special report preview dialog boxes, you need to comment out the following line:

```
SELF:Preview()
```

1. Double-click on the CustList:Init() method from the Customer:Reports Entity Browser (which should still be displayed) to bring up the Source Code Editor:



```
Method Init(oOwner) Class CUSTLIST
  super:Init(oOwner,"CARETRUN")
  self:open(CUSTLIST_file)
  // The following statement allows the user to preview the report.
  // self:Preview()
  // Use the following statement to print the report
  // self:Print()
```

2. Move the cursor to the line of code reading SELF:Preview(), and insert the characters // at the beginning of the line.
3. Close the Source Code Editor by double-clicking on its system menu, and answering Yes when prompted.

For more information on reports, see the ReportQueue Class topic in CA-Visual Objects online Help.

Report Parameters

In the previous section, you were briefed on running your report within your application. Often, it is necessary to *filter* the report based on information at runtime (for example, printing a Payment Report for the month of June or printing Invoice number “04729”).

In the South Seas Adventures application, the Customer List report should be able to print either all customers or a single customer. To do this, you can define parameters for the report. These parameters can then be inserted into the query for the report.

First, let’s define a parameter for the report.

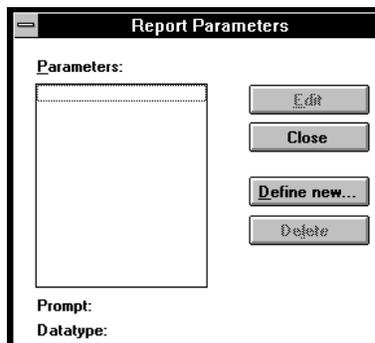
1. To reopen your report, double-click on the CustList report entity:



This will bring you back into CA-RET.

2. Select the Report Parameters command from the Edit menu.

The Report Parameters dialog box appears:



3. Choose the Define New button.

The Define Parameter dialog box appears:

The screenshot shows a dialog box titled "Define Parameter". It has a standard Windows-style title bar with a close button. The dialog contains the following elements:

- Parameter name:** A text field containing "_parameter1".
- Prompt:** A text field containing "Enter value:". This field is highlighted with a red box in the original image.
- Default value:** An empty text field.
- Description:** An empty text field.
- Datatype:** A section with a label "Datatype" and four radio button options:
 - Character
 - Number
 - Date
 - Bucket Table
- Buttons:** "OK" and "Cancel" buttons are located on the right side of the dialog.

When you invoke a report that expects parameters, the application may either have the parameters hard coded in the program or allow the user to enter them at runtime. The Prompt edit control determines what the end user will see, if the application does not pass the required parameters.

4. For this report, type **Enter a customer #:** in the Prompt edit control.

Note: The Default Value edit control determines the default to use if the end user does not type anything.

5. In the Description edit control, type **Enter a 5 digit number identifying a customer.**
6. Choose OK.

Now that you have defined the parameter, you must include it in the report query before you can use it. You can also display it on your report, since it appears in the Insert Field dialog box.

When the report was initially created, CA-Visual Objects constructed a SQL statement based on the servers that you selected in the CA-Visual Objects Report Editor dialog box. The generated statement is as follows:

```
SELECT Customer.* FROM ;
        customer.dbf (cust1.ntx/USE, cust2.ntx)
```

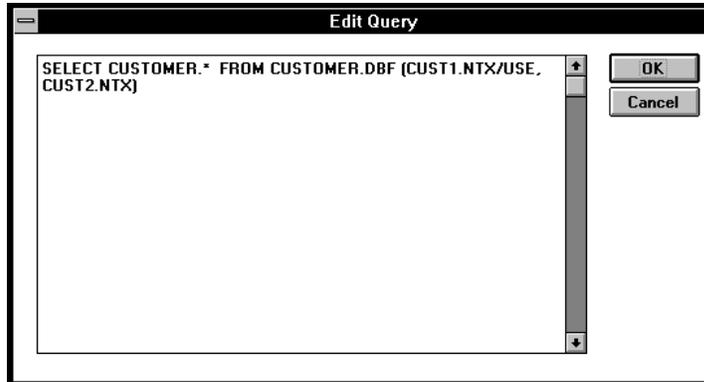
In English, this statement reads “include all fields from the customer table and order the records using CUST1.NTX.”



In the steps that follow, you will alter this SQL statement to use the parameter that you defined in the previous steps.

1. Choose the Edit Query toolbar button.

The Edit Query dialog box appears:



2. Move the cursor to the end of the last line in the list box.
3. Insert a blank space, and add the following WHERE clause to the existing SQL statement:

```
WHERE ((Cust_ID = '['_parameter1']') OR
('['_parameter1']' IS NULL))
```

Important! Make sure there are no extra spaces between the “[_ and 1]” characters.

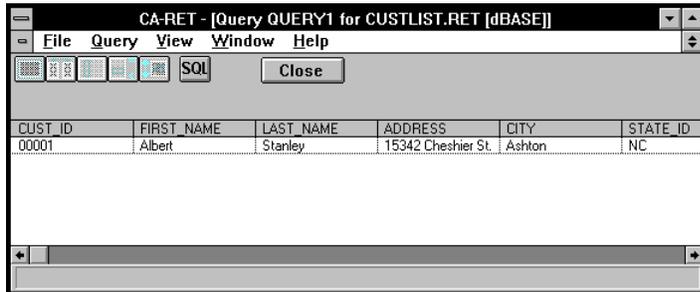
This WHERE clause serves a dual purpose, allowing the report to be printed either for all customers or a single customer. When a valid customer number is passed (Cust_ID = '['_parameter1]'), the report will include only that customer. If an empty string is passed ('['_parameter1']' IS NULL), the report will include all customers. Although this is a fairly simple query, CA-RET is also capable of handling very complex queries.

4. Choose OK to accept the new statement.

CA-RET now prompts you for each parameter used in the query when you preview the report.

- At the prompt, type **00001**.

The query table displays only the record for customer 00001:



The screenshot shows a window titled "CA-RET - [Query QUERY1 for CUSTLIST.RET [dBASE]]". The window has a menu bar with "File", "Query", "View", "Window", and "Help". Below the menu bar is a toolbar with several icons and a "Close" button. The main area of the window displays a table with the following data:

CUST_ID	FIRST_NAME	LAST_NAME	ADDRESS	CITY	STATE_ID
00001	Albert	Stanley	15342 Cheshier St.	Ashton	NC

Note: CA-RET comes with a powerful Query Builder tool. This can help you create queries without knowing SQL. For more on the Query Builder, see the Query Builder Overview topic in CA-Visual Objects online Help.

- Choose Close to cancel the report.
- Choose the Save toolbar button to save your changes.
- To close CA-RET and return to CA-Visual Objects, double-click on the system menu.

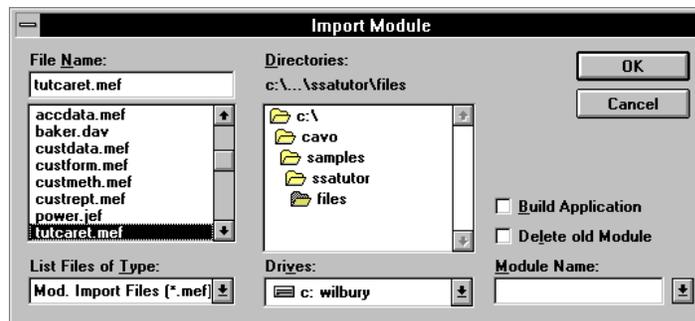


Passing Parameters to CA-RET from CA-Visual Objects

The ReportQueue class, from which the CustList class was derived, allows you to pass parameters to your CA-RET report. To see an example of this, you can import a previously created module with code designed to work with the Customer List report.

1. Select the Import command from the File menu.

The Import Module dialog box appears:



The module that you want to import is called TUTCARET.MEF, and it is located in your CA-Visual Objects SAMPLES\SSATUTOR\FILES subdirectory.

2. Select TUTCARET.MEF and choose OK.
3. Open the new module, Tutorial:CA-RET, by double-clicking on its button in the Module Browser.
4. Open the SSAWindow:CustomerReport() method by double-clicking on its button in the Entity Browser.

The Source Code Editor window, displaying the CustomerReport() method, appears:

```

METHOD CustomerReport CLASS SSAWindow

LOCAL oDialog AS CustRptDialog
LOCAL oReport AS CustList
LOCAL aParams AS ARRAY
(oDialog := CustRptDialog{self}):Show()
IF oDialog:IPrintOk

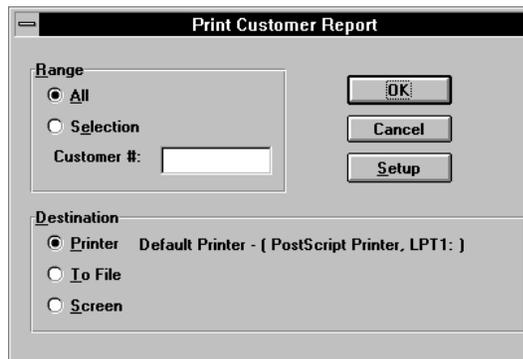
    oReport := CustList{self}

    aParams := { oDialog:cCust_Id } // Print only this customer
DO CASE
CASE oDialog:nDestination == PRINT_PRINTER
    oReport:Print( aParams )
CASE oDialog:nDestination == PRINT_SCREEN
    oReport:Preview( aParams )
CASE oDialog:nDestination == PRINT_FILE
    oReport:SaveToFile( oDialog:ToFileFS:FullPath, ;
        RIGHT( oDialog:ToFileFS:Extension, 3 ), aParams )
ENDCASE

ENDIF
RETURN self
    
```

Line: 23 Col: 1, Changed

This method causes the following Print Customer Report dialog box to appear:



This allows the user to include all customers or only a single customer on a report, and also allows the report to print to various destinations.

Note: This dialog box is defined in the Customer:Reports module as `_CustRptDialog`. The subclass, `CustRptDialog`, contains additional methods defined to work with this dialog box. You may want to refer to the source code for these methods to completely understand the source code you are viewing for this imported module.

Once the Print Customer Report dialog box closes, an array is created, as follows:

```
aParams := {oDialog:cCust_Id}
```

The `ReportQueue Print()`, `Preview()`, or `SaveToFile()` method is then called using this array as a parameter, like so:

```
CASE oDialog:nDestination == PRINT_PRINTER
    oReport:Print(aParams)
CASE oDialog:nDestination == PRINT_SCREEN
    oReport:Preview(aParams)
CASE oDialog:nDestination == PRINT_FILE
    oReport:SaveToFile(..., aParams)
```

The elements of the array will be passed to the CA-RET report—one for each parameter required.

5. When you are finished examining this source code, close the Source Code Editor by double-clicking on its system menu.

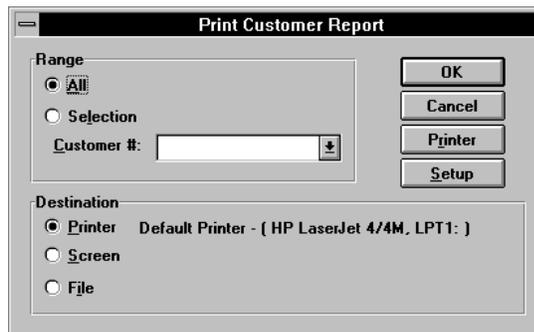
Verifying the Results



Verify the results using the following steps:

1. Build the application by clicking the Build toolbar button.
2. Run the South Seas Adventures application by clicking the Execute toolbar button.
3. Log in to the application as usual (Name: **User**, Password: **Trainee**).
4. Select the Customers command from the Reports menu.

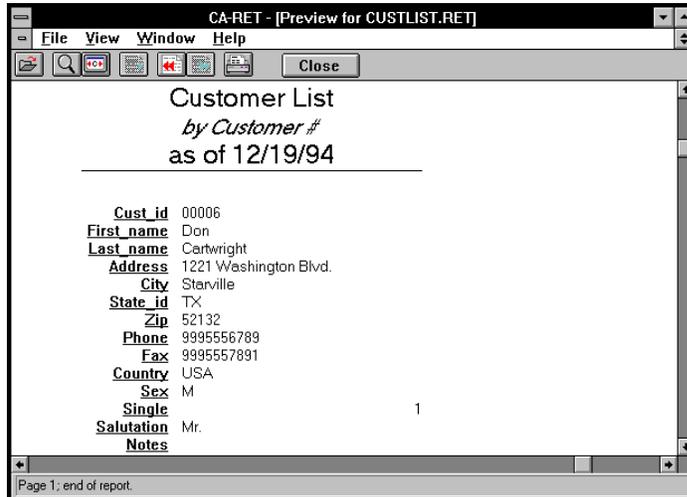
The Print Customer Report dialog box appears:



5. Click on the Selection radio button, and choose Cartwright from the Customer # combo box.
6. Click on the Screen radio button to print the report to the screen.

7. Choose OK to run the report.

The runtime CA-RET engine is launched, with your report in full view:



8. When you are done, choose the Exit command from the File menu to close CA-RET.
9. Exit the South Seas Adventures application by double-clicking on its system menu, and choosing Yes when prompted.

Summary

In this lesson, you have learned how to use CA-RET to create and customize a basic report, and how to add parameters. You have also learned how to modify the SQL Query (which defines which records will be included in your report), and how to pass parameters to your CA-RET report from your CA-Visual Objects application.

In the following lesson, you will learn how to quickly and easily debug your application. You will use the Error Browser to fix compiler errors, and the CA-Visual Objects Debugger to locate and fix logic or runtime errors.