Chapter 4
Using the R&R Custom Control

Introduction

Microsoft Visual Basic (VB) provides for extensions to its functionality by means of Custom Controls, sometimes called VBXs. These controls are represented in VB at design time by tools in the Toolbox that can be placed onto your VB forms to perform various tasks. As noted in Chapter 1, the Custom Control provides one of three methods for running R&R reports using the R&R Runtime. The other methods are explained in Chapter 2, "Using the Runtime Executable," and Chapter 3, "Accessing the Runtime DLL."

Like the report control in the Professional Edition of VB, the R&R Custom Control allows you to incorporate database reporting into your VB applications. However, it provides the following additional advantages:

- ❑ Access to the more powerful reporting capabilities of R&R;
- ❑ More control over how your report is printed by means of over 50 properties that can be set at design time or run time;
- ❑ Far more design-time support in the form of dialog boxes that allow you to point and click to override settings in your report, such as tables, sorting and grouping, user parameters, and destination.

The R&R Custom Control is explained in the following sections:

- Installation
- Using the Custom Control
- Determining Report Status
- Using RRW.INI for Default Information
- Custom Control Properties

Installation

If you select "Visual Basic and C/C++ Files" in R&R Setup, the R&R Custom Control will be added to your VB Toolbox in all new VB projects. It looks like this:

Setup also copies a sample VB project into the VB subdirectory of your R&R program directory.

To add the R&R Custom Control to an existing VB project, select “Add File” on the VB File menu to add the file RRW.VBX, which Setup installs into your Windows System directory. For information about other files you will need to distribute when you use the R&R Custom Control in your applications, see Chapter 7, “Distributing Reports.”

Using the Custom Control

You use the R&R Custom Control just like any other custom control. Simply click on the R&R tool in the VB Toolbox. Then move the mouse pointer over your form, press the left mouse button down, and drag the mouse. When you release the mouse button, the custom control will be placed onto the form.

To change the value of a custom control property, click on the property in the Properties list. The data type of the property determines how you change it:

- You change an integer property by simply typing a value into the settings box.
- You change an enumerated property by typing a number into the settings box or by selecting a value from the drop-down list that appears when you click on the down arrow next to the settings box. You can also double-click on an enumerated property to cycle through the list of values.
- You change string properties by typing a string in the settings box. You can change many string properties by means of dialog boxes that appear when you click on the ellipsis (...) to the right of the settings box, or when you double-click on the property.

Note that the R&R Custom Control will be visible on your VB form at design time only. At run time, it will not be displayed, but will be “at your service” to print your reports to a printer; to a preview window; or to a text, database, or spreadsheet file. It does this by invoking the Runtime executable (described in Chapter 2) through the Runtime DLL (described in Chapter 3). As you will see, the R&R custom control properties correspond closely to the control parameters used by the Runtime. (See Appendix A, “Runtime Equivalencies.”)

To print or display a report in your VB program, you must set at least three properties:

1. Set the ReportLibrary property to the file name of the report library.
2. Set the ReportName property to the name of the report you want to print.
3. Set the Action property to 1 to trigger the printing of the report.

The ReportLibrary and ReportName properties can be set at design time or at run time. The Action property must be set at run time in your procedure code (e.g., Report1.Action = 1). You can also set many other properties (Destination, SortFields, etc.) to override the values saved in the report.

Determining Report Status

When you use the Custom Control to print a report, you may want to know the status of the report, such as whether the report printed successfully or, if not, which error occurred while printing. How the Custom Control returns this status to you depends on which method of report printing you use.

The Custom Control supports two methods of report printing, *synchronous* and *asynchronous*. Synchronous printing means that the report will complete printing before the next line of procedure code is executed. Asynchronous printing means that the report will be printed while the remaining lines of procedure code are executing. Each method has its own way of returning status information to you.

You print a report synchronously by setting the Action property to 1 (e.g. Report1.Action = 1). When you print a report synchronously, the status of the report is returned in the following properties:

- ❑ LastErrorCode, which will contain the type of error that occurred, or 0 for no error;
- ❑ LastErrorString, which will contain a text message describing the error, if any;
- ❑ LastErrorPage, which will contain the page number of the last page printed.

You print a report asynchronously by setting the Action property to 2 (e.g., Report1.Action = 2). When you print a report asynchronously, the status of the report is not returned in a property, but is written into the Runtime status file, which is a text file called RRWRUN.OUT in the current working directory. See the section entitled **Understanding the Runtime Status File** in Chapter 2 for a description of the contents of this file.

Using RRW.INI for Default Information

If RRW.INI, the R&R configuration file, is in the Windows directory, the Runtime will use the default settings specified in that file. However, custom control properties take precedence; any setting you specify using a custom control property will always override the corresponding RRW.INI setting(see Figure 2.2 in Chapter 2 for a list of RRW.INI settings that the Runtime will use).

If you distribute reports to other users, you can customize RRW.INI for each user and distribute it with the other Runtime files. However, the custom control properties provide a more reliable way to control or override the defaults saved with a report.

Custom Control Properties

The following properties are available in the R&R Custom Control:

(About)

Description

Double-click About to display the version of the R&R custom control.

Availability

Design time only

Action

Description

Action is a property that triggers the print, display, or export of the report.

Usage

[form.]Report.Action [=Action%]

Example

```
Report1.Action = 1
```

« Prints, displays, or exports the report, depending on the Destination property, and does not return until the report is completed. »

Comments

Set the Action property to 1 or 2 in your procedure code to print, display, or export the report in response to a user event. In most cases, it will be more convenient to set this property to 1.

If set to 1, the action is synchronous, which means that the next line of Visual Basic procedure code will not execute until the report is completed. The status of the report will be returned in the LastErrorCode, LastErrorString, and LastErrorPage properties.

If set to 2, the action in asynchronous, so that the report may still be running when the next line of Visual Basic procedure code is executed. When the report completes, its status is written into the status file.

Data Type

Integer

Availability

Write-only at run time

CopiesToPrinter

Description

Specifies the number of copies to be printed if you are printing to a printer (if the Destination property is set to 1).

Usage

[form.]Report.CopiesToPrinter[= NumCopies%]

Example

```
Report1.CopiesToPrinter = 3
```

« Prints three copies of the report. »

Comments

This property is optional. The number must be between 0 and 999, inclusive. If you leave this property blank or enter 0, the Runtime prints the number of copies saved with the report.

Data Type

Integer

Availability

Design time; Run time

DataDirectory

Description

Specifies the default directory where the Runtime will look for tables, indexes, and text memo files to be used when the report is printed.

Usage

[form.]Report.DataDirectory[= DirectoryName$]

Example

```
Report1.DataDirectory = "c:\mis\data"
```

« Looks for data files in a directory called “c:\mis\data.” »

Comments

If the tables, indexes, and text memo files used in the report are not in the saved directories, then the Runtime will look in this directory for these files.

Data Type

String

Availability

Run time

Destination

Description

Specifies the destination to which your report is to be printed or exported (Preview Window, Printer, Text File, DBF File, WKS File, or RTF File).

Usage

[form.]Report.Destination[= Destination%]

Example

```
Report1.Destination = 1
```

« Sends the report to a preview window. »

Comments

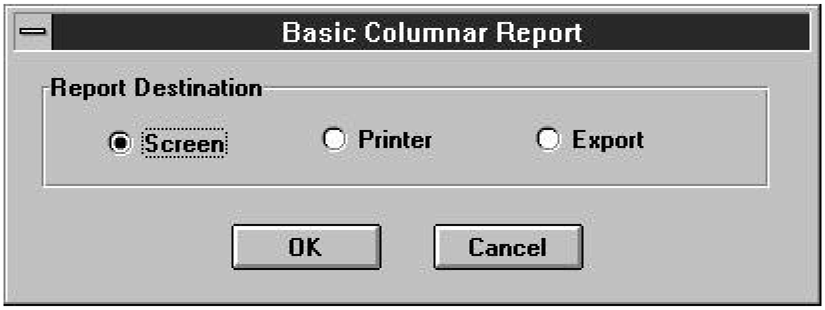
This property is optional. Set the property to 0 (the default) to print to the printer saved with the report (or to the printer specified in the Printer property). It can contain one of the following values:

0 – Saved (uses the printer saved with the report);
1 – Window (sends the report to a preview window);
2 – Printer (sends the report to a printer);
3 – Text File (exports the report to an ASCII text file);
4 – DBF File (exports the report to a DBF database file);
5 – WKS File (exports the report to a WKS spreadsheet file);
6 – Prompt user (asks user for destination);
7 – RTF File (exports to a Rich Text Format file);
8 – Text Data File (exports to a Text Data file);
9 – Word Merge file (exports to a Word Merge file).

If you specify 3 (Text File), 4 (DBF File) 5 (WKS File), 7 (RTF File), 8 (Text Data file), or 9 (Word Merge file), you can also set the PrintFileName property to provide the name of the destination file to override the saved destination file name.

If you select Window, the report will be sent to the display, allowing the user to preview the report before printing it. After previewing the report, the user can select the Print button in the Preview window to send the report to the printer saved with the report or specified in the Printer property. Note that if the value of Destination is 1 and the PrintFileName property has been set, the report will be output to the file specified in PrintFileName when the user selects Print in Preview.

Setting this property to 6 allows the user to select the print destination at run time. The user will see the dialog box shown in Figure 4.1. If the WindowTitle property is set, the title bar will contain the WindowTitle value. If WindowTitle is empty, the title bar will contain the report name.



Basic Columnar Report

Report Destination

● Screen ○ Printer ○ Export

Figure 4.1 Print Destination Dialog Box

The user can select Screen to preview the report, Printer to print it, or Export to display the Export dialog. If the user selects Cancel, the report will not run, and the “Canceled” status message will be returned in the LastErrorString property or the Runtime status file.

Data Type

Integer (Enumerated)

Availability

Design time; Run time

DisplayError

Description

Specifies whether errors are to be displayed when a report is printed.

Usage

[form.]Report.DisplayError [= {True|False}]

Example

```
Report1.DisplayError = True
```

« Specifies that any errors that occur when a report is printed will be displayed. »

Comments

This property is optional. If DisplayError is True, Runtime error messages are displayed in addition to being returned in the LastErrorString property. In this case, the Runtime stops processing a report when it encounters an error and displays an error message dialog. The user must then select OK to acknowledge the error and resume processing.

If DisplayError is False, Runtime error messages are not displayed, but are returned in the LastErrorString property or the Runtime status table.

Data Type

Integer (Boolean)

Availability

Design time; Run time

DisplayStatus

Description

Specifies whether or not status information is to be displayed when a report is printed.

Usage

[form.]Report.DisplayStatus [= {True|False}]

Example

```
Report1.DisplayStatus = False
```

« Specifies that status information will not be displayed when a report is printed. »

Comments

The DisplayStatus property enables you to specify whether the Runtime should display a Print Status window while it is generating a report. If the property is set to True, the Runtime will display a Status window. If NoEscape is set to False, the Status window will contain a Cancel choice that allows the user to terminate a report in progress.

If DisplayStatus is set to False, the Runtime will not display a Status window but will instead display as an icon while it is running.

Data Type

Integer (Boolean)

Availability

Design time; Run time

EndPage

Description

Specifies at which page of the report to end printing.

Usage

[form.]Report.EndPage[= Page%]

Example

```
Report1.EndPage = 20
```

« Specifies that the report should end printing at the completion of page 20. »

Comments

This property is optional. The StartPage and EndPage properties allow you to override the starting and ending page numbers saved with the report. The default value for these properties is zero.

To specify page numbers, include a StartPage value, an EndPage value, or both. If you specify both, EndPage must be equal to or greater than StartPage. For example, users can restart a canceled report where it was interrupted by specifying the starting page number as the StartPage value and 999999999 as the EndPage value. To reprint one or more consecutive pages, specify the page numbers in the StartPage and EndPage properties. To print just one page, specify the same value for both.

Data Type

Integer

Availability

Design time; Run time

ExportDestination

Description

Specifies the destination (display, file, or printer) when exporting to an Excel PivotTable or Chart.

Usage

[form.]Report.ExportDestination[= Destination%]

Example

```
Report1.ExportDestination = 2
```

« Sends the PivotTable or Chart export to a file. »

Comments

Set this property to one of the following values if you are exporting to an Excel PivotTable or Chart..

0 – Window;
1 – Printer;
2 – File.

If you set the value to 2 (File), use PrintFileName to specify a file name for the export.

Data Type

Integer (Enumerated)

Availability

Design time; Run time

Filter

Description

Specifies a query to select records to be used when printing the report.

Usage

[form.]Report.Filter[=Filter$]

Example

```
Report1.Filter = "Year > 1995"
```

Comments

The optional Filter property specifies a logical expression that will override the query saved with a report, if any, when the value in Query is 2.

The syntax of the Filter expression is identical to that of a calculated field expression that returns a logical value. The Filter expression can be up to 1024 characters long. When an expression is specified, the Runtime selects all records where the value of the Filter expression is true. The expression can refer to any data or calculated fields that are available in the report.

For example, if you enter the expression **CITY="Dallas"**, the Runtime will select all records where the value of this expression is true, in other words all records where the value in the CITY field is Dallas. If the city name were in a character field named NOTE, the expression **NOTE="*Dallas*"** would select all records in which the NOTE field contained the word "Dallas".

Entering the expression **PASTDUE=T** tells the Runtime to select all records where the value in the PASTDUE field is true. Entering **AMOUNT>=200** will select all records where the value in the AMOUNT field is equal to or greater than 200. Entering the following expression will select all records where the date in the INVDATE field of the RRORDERS table is January 31, 1995:

```
RRORDERS->INVDATE={01/31/95}
```

Compound expressions can be entered by using parentheses. For example, the following expression selects all records where the value in the CITY field is either Dallas or Houston and where the value in the SALES field is greater than 50,000:

```
(CITY="Dallas" or CITY="Houston") and SALES>50000
```

Note that the value of Query *must* be 2 in order for the Filter override to take effect. If you omit Query, the Filter value will be ignored and the report will be run using the saved query (if any).

When setting this property at run time, make sure that you enclose your query expression in double quotes. If your query expression contains internal quotes, such as:

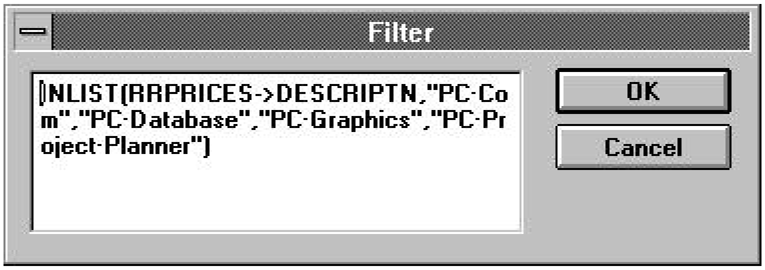
LNAME = "Jones"

make sure to change all of the internal double quotes to single quotes and then put double quotes around the entire query expression, such as:

"LNAME = 'Jones'"

At design time, you can change this property array in two ways:

□ Double-click this property to see the dialog shown in Figure 4.2, which will contain the query expression contained in the report., if any.



```
INLIST(RRPRICES->DESCRIPTN,"PC-Com","PC-Database","PC-Graphics","PC-Project-Planner")
```

Figure 4.2 Query Expression Dialog Box

□ Simply enter the query expression into the settings box.

Data Type

String

Availability

Design time; Run time

GroupFields

Description

Specifies the field(s) to be used to group the data in your report.

Usage

[form.]Report.GroupFields(ArrayIndex)[= "+|-GroupField"]

Example

```
Report1.GroupFields(0) = "Division"
```

« Use "Division" as the first group field. »

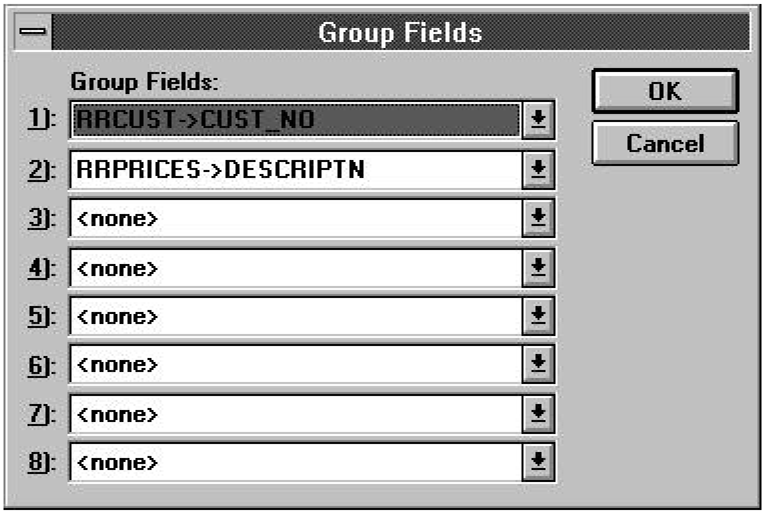
Comments

Group fields can be database fields, calculated fields, or total fields.

When setting this property at run time, use a separate line of code to specify each group field. The first group field you specify must be assigned array index 0, the second group field must be assigned array index 1, etc. The index values you assign must be continuous; no gaps are allowed (0,1,2 would be correct, but 0,1,3 would be wrong).

At design time, you can change this property array in two ways:

- Double-click this property to see the dialog shown in Figure 4.3, which contains a list of group fields in the report. Clicking on the down arrow next to each group field will drop down a list of all fields used in the report from which you can select.



Group Fields:
1): RRCUST->CUST_NO
2): RRPRICES->DESCRIPTN
3): <none>
4): <none>
5): <none>
6): <none>
7): <none>
8): <none>

Figure 4.3 Group Fields Dialog Box

- Enter group field names separated by semicolons. To override some group fields, but not all of them, you must use a semicolon as a place-holder. For example, to change the first and third group field, you would enter “Division;;Region”.

Data Type

Array of strings

Availability

Design time; Run time

HighScope

Description

Specifies the high scope of the records to be used when the report is printed.

Usage

[form.]Report.HighScope[= Scope$]

Example

```
Report1.HighScope = "Smith"
```

« Specifies that the scope of records to be used should stop just after “Smith” when the report is printed. »

Comments

Set this property if you want to override the high scope value saved in the report. Be sure to set the Scope property to 2 to use the override. See **Scope** for a complete description of this feature.

Data Type

String

Availability

Design time; Run time

ImageDirectory

Description

Specifies the default directory where the Runtime will look for image files used in the report.

Usage

[form.]Report.ImageDirectory[= DirectoryName$]

Example

```
Report1.ImageDirectory = "c:\mis\images"
```

« Looks for image files in a directory called “c:\mis\images.” »

Comments

The Runtime will look for image files in this directory when they are not in the saved directory. The directory you specify with this property will override any default image directory specified in the RRW.INI file.

Data Type

String

Availability

Run time

IndexExtension

Description

Specifies the default index file extension.

Usage

[form.]Report.IndexExtension[=Extension%]

Example

```
Report1.IndexExtension=5
```

« Uses “ntx” as the default index file extension. »

Comments

This property is optional; it can contain one of the following values:

0 – None	4 – NDX (dBASE III)
1 – CDX (FoxPro)	5 – NTX (Clipper)
2 – IDX (FoxBASE+)	6 – NSX (SuccessWare)
3 – MDX (dBASE IV)	7 – WDX (WordTech)

The default index extension is used to locate indexes that are specified without extensions or that cannot be located using the extensions saved with the report.

Data Type

Enumerated integer

Availability

Run time; design time

LastErrorCode

Description

Returns the error code for the last runtime error. It will be one of the following four values:

0 = No error;

1 = User canceled;

2 = Error in runtime parameters;

3 = Error in report.

Usage

[form.]Report.LastErrorCode

Example

```
'If error occurs, display error message
Report1.Action = 1
if Report1.LastErrorCode <> 0 then
      MsgBox  Report1.LastErrorString
end if
```

« If an error occurs, this code calls up a message box that displays the error string. »

Comments

LastErrorCode is only valid after setting the Action property to 1. If you set Action to 2, the report is run asynchronously, so LastErrorCode will not be set.

Data Type

Integer

Availability

Run time (read and write)

LastErrorPage

Description

Returns the page number of the last successfully printed page.

Usage

[form.]Report.LastErrorPage

Example

```
'If error occurs, display error message
Report1.Action = 1
if Report1.LastErrorCode <> 0 then
   pagestr$ = "; last page printed was "
   + str(Report1.LastErrorPage)
   MsgBox  Report1.LastErrorString + pagestr$
end if
```

« If an error occurs, this code calls up a message box that displays the error string and the last page printed. »

Comments

LastErrorPage is only valid after setting the Action property to 1. If you set Action to 2, the report is run asynchronously, so LastErrorPage will not be set.

Data Type

Integer

Availability

Run time (read and write)

LastErrorString

Description

Returns the error string for the last runtime error.

Usage

[form.]Report.LastErrorString

Example

```
'If error occurs, display error message
Report1.Action = 1
if Report1.LastErrorCode <> 0 then
      MsgBox  Report1.LastErrorString
end if
```

« If an error occurs, this code calls up a message box that displays the error string. »

Comments

LastErrorString is only valid after setting the Action property to 1. If you set Action to 2, the report is run asynchronously, so LastErrorString will not be set.

Data Type

String

Availability

Run time (read and write)

LowScope

Description

Specifies the low scope of the records to be used when the report is printed.

Usage

[form.]Report.LowScope[= Scope$]

Example

```
Report1.LowScope = "Jones"
```

« Specifies that the scope of records to be used should start at "Jones" when the report is printed. »

Comments

Set this property if you want to override the low scope value saved in the report. Be sure to set the Scope property to 2 to use the override. See **Scope** for a complete description of this feature.

Data Type

String

Availability

Design time; Run time

MasterIndex

Description

Specifies the master index to be used in the report.

Usage

[form.]Report.MasterIndex[= MasterIndex$,
[IndexType$, IndexTag$]]

Example

```
Report1.MasterIndex = "c:\mis\sales.mdx, N, CUSTID"
```

« Uses the file c:\mis\sales.mdx, a numeric index with a tag of CUSTID in the report. »

Comments

The MasterIndex property enables you to specify a master index for a report that was saved without one, override the master index saved with a report, or remove a saved master index from a report. To specify a master index or override the saved one, the specification can consist of any or all of three values in the following order:

```
  <index>,<index type>,<tag>
```

In this specification, <index> represents the index file specification, which can be a complete path and file name, a directory, or a file name.

- ❑ If you specify both a directory and a file name, this directory is the only directory searched and this file name is the only file the Runtime searches for.
- ❑ If you specify a directory without a file name, the Runtime searches the specified directory for the master index name saved with the report.
- ❑ If you specify a file name without a directory, the Runtime searches for a file with the specified name in the directory of the master index saved with the report, then in the current master database directory, then in the default data directory specified via the DataDirectory property or in RRW.INI. If no default is specified, the Runtime searches for the file in the current directory.

In this specification, <index type> is the data type of the new index, represented as **N** for numeric, **D** for date, or **C** for character. You can omit this value if you are overriding a saved master index and the replacement index is the same type (although it is good practice to include the data-type specifier). If the index named in the specification is a multiple-field index file (MDX, CDX, or WDX), the <tag> part of the specification represents an index tag. If you omit the <index> or <index type> part of the specification, use a comma as a place holder.

For example, the following specification replaces the master index saved with the report with an index named CUST94.MDX in the CUSTOMER directory on drive C. Since the replacement index is the same data type as the original index, the specification includes a comma in place of the <index type>. It also replaces the saved index tag with the CUSTID index tag:

```
C:\CUSTOMER\CUST94.MDX,,CUSTID
```

The following specification changes only the index tag of the master index saved with the report:

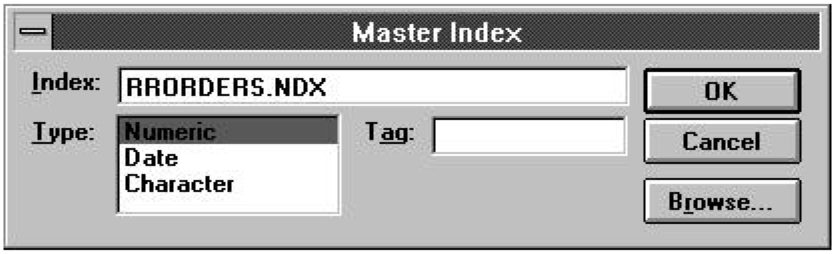
```
,,CUSTID
```

To remove the master index saved with the report without specifying a new master index, use the MasterIndex value **,R**.

If you omit this field or leave it blank, the Runtime uses the master index saved with the report, if any.

At design time, you can change this property in two ways:

□ Double-click this property to see the dialog shown in Figure 4.4, which allows you to select an index, type and tag.



Master Index

Index: RRORDERS.NDX

Type: Numeric
Date
Character

Tag:

Figure 4.4 Master Index Dialog Box

□ Simply enter the index, type and tag into the settings box.

Data Type

String

Availability

Design time; Run time

MasterTable

Description

Specifies the name of a table that will override the master table saved with the report.

Usage

[form.]Report.MasterTable[= MasterTableName$]

Example

```
Report1.MasterTable = "c:\mis\sales.dbf"
```

« Uses the file c:\mis\sales.dbf as the master table for the report. »

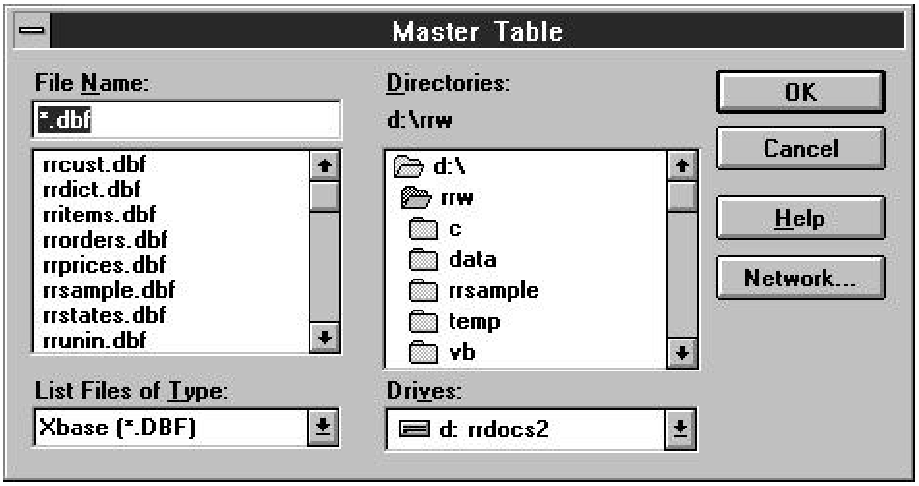
Comments

This property is optional. The master table you specify should have the same columns as the master table originally used in the report.

If you leave this property blank, the Runtime uses the master table saved with the report.

At design time, you can change this property in two ways:

- □ Double-click this property to see the dialog shown in Figure 4.5, which allows you to select a table.



Master Table

File Name: *.dbf

rrcust.dbf
rrdict.dbf
rritems.dbf
rrorders.dbf
rrprices.dbf
rrsample.dbf
rrstates.dbf
rrunin.dbf

Directories: d:\rrw

d:\
rrw
c
data
rrsample
temp
vb

List Files of Type: Xbase (*.DBF)

Drives: d: rrdocs2

Figure 4.5 Tables Dialog Box

- □ Simply enter the table name in the settings box.

Data Type

String

Availability

Design time; Run time

MemoFileName

Description

Specifies the name and optional directory location of the text memo file to be used in the report, which will override the text memo file saved with the report.

Usage

[form.]Report.MemoFileName[= MemoFileName$]

Example

```
Report1.MemoFileName = "c:\mis\q3notes.txt"
```

« Selects the memo file named “q3notes.txt” in the c:\mis directory. »

Comments

This property is optional.

- ❑ If both a directory and a file name are specified, this directory is the only directory searched and this file name is the only file the Runtime searches for.
- ❑ If a directory is specified without a file name, the Runtime searches the specified directory for the text memo file name saved with the report.
- ❑ If a file name is specified without a directory, the Runtime searches for a file with the specified name in the directory saved with the report, then in the default data directory as specified in the DataDirectory property or in RRW.INI.

If you leave this property blank, the Runtime uses the text memo file saved with the report, if any.

At design time, you can change this property in two ways:

- ❑ Double-click this property to see the MemoFileName common dialog, which allows you to select a memo file and browse drives, directories, and files to which you have access.

- ❑ Simply enter the file name into the settings box.

Data Type

String

Availability

Design time; Run time

NoEscape

Description

Specifies whether a report can be canceled.

Usage

[form.]Report.NoEscape [= {True|False}]

Example

```
Report1.WindowNoEscape = True
```

« Specifies that a report cannot be canceled once it begins to be printed. »

Comments

This property is optional, and can be set to either True or False. True means the Cancel button in the status window is not active while reports are being output. False means the user may select Cancel during report output to pause or end the job. The default value is false.

Note that the Status window appears only when the DisplayStatus property is set to true.

If the user cancels synchronous execution of the report, the LastErrorCode property is set to C. If the report is run asynchronously, the RO_ECODE entry in the status file contains a C (see the section in Chapter 2 entitled **Understanding the Runtime Status File**.)

Data Type

Integer (Boolean)

Availability

Design time; Run time

Parameters

Description

Specifies user parameter values to be used when the report is printed.

Usage

[form.]Report.Parameters(ArrayIndex)
[= ParameterName$=ParameterValue$]

Enter a “name=value” string for each RIPARAM( ) function in your report for which you want to define a value. Use a separate line of code for each change.

The order of strings in the array does not matter, since each RIPARAM( ) function is identified by name.

Example

```
Report1.Parameters(0) = "Title=Cumulative Earnings"
```

« Uses the value “Cumulative Earnings” wherever the function RIPARAM(“Title”) appears in the report. »

Comments

Use this property to define values for the RIPARAM( ) functions in your report. You can specify up to six (6) different parameters in the custom control (Parameters(0) – Parameters(5)). See the **Parameter Passing** section of Chapter 2 for information on using this feature.

At design time, you can change this property array in two ways:

- ❑ Double-click this property to see the dialog box shown in Figure 4.6, which contains a list of parameters and values in the report.

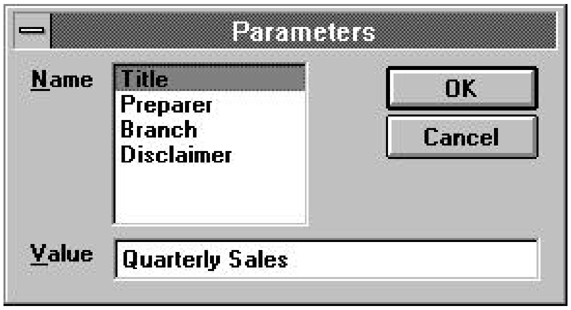


Figure 4.6 Parameters Dialog Box

- ❑ Enter the parameter/value pairs separated by semicolons.

Data Type

Array of strings

Availability

Design time; Run time

Port

Description

Specifies the name of the printer port to which the report is to be printed.

Usage

[form.]Report.Port [= PortName$]

Example

```
Report1.Port = "LPT1:"
```

« Prints the report to the printer connected to “LPT1:". »

Comments

This property is optional. Enter a value such as “LPT1:” to override the printer port (and the printer associated with that port) saved with the report. Note that the colon is required.

You can also use the question mark (**?**) value for this property. When the Port property contains a question mark, the user will see the Printers dialog box shown in Figure 4.7. (See the description of the Printer property.)

At design time, you can change this property in two ways:

- ❑ Double-click this property to see the Printer common dialog, which allows you to select and configure printers that have been installed on various ports on your system.
- ❑ Simply enter the port name into the settings box.

Data Type

String

Availability

Design time; Run time

Printer

Description

Specifies the name of the printer to which the report is to be printed.

Usage

[form.]Report.Printer [= PrinterName$]

Example

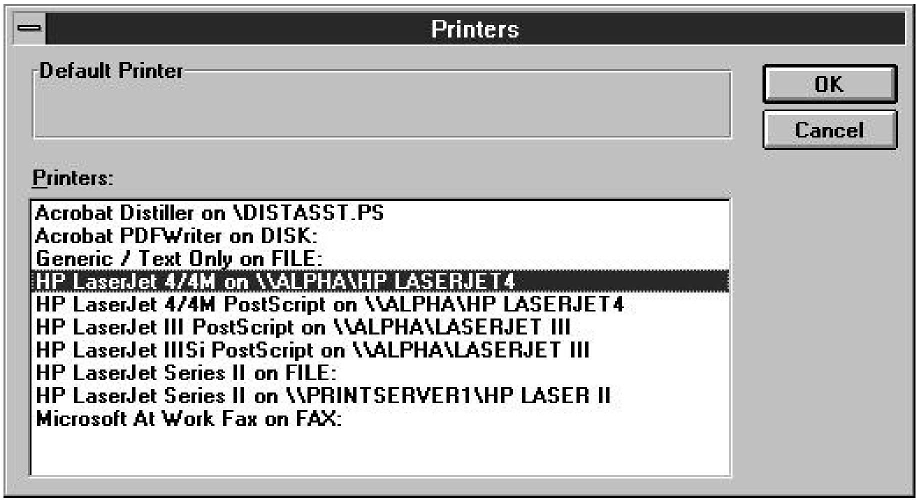
```
Report1.Printer = "HP LaserJet 4/4M"
```

« Prints the report to a printer called "HP LaserJet 4/4M." »

Comments

This property is optional. Enter a value to override the printer saved with the report. This property can have one of two values:

- ❑ The name of an available Windows printer. Available Windows printers are listed in the R&R Printers dialog (accessed by selecting File ⇒ Printers in interactive R&R). The value is case insensitive (that is, you can enter it in upper, lower, or mixed case).
- ❑ The question mark (**?**) value, to allow the user to select a printer at runtime. When the Printer property contains a question mark, the Printers dialog will display, as shown in Figure 4.7.



Acrobat Distiller on \DISTASST.PS
Acrobat PDFWriter on DISK:
Generic / Text Only on FILE:
HP LaserJet 4/4M on \\ALPHA\HP LASERJET4
HP LaserJet 4/4M PostScript on \\ALPHA\HP LASERJET4
HP LaserJet III PostScript on \\ALPHA\LASERJET III
HP LaserJet IIISi PostScript on \\ALPHA\LASERJET III
HP LaserJet Series II on FILE:
HP LaserJet Series II on \\PRINTSERVER1\HP LASER II
Microsoft At Work Fax on FAX:

Figure 4.7 Printers Dialog Box

Printer Setup in the Windows Control Panel controls which printers are listed in the Printers dialog box. R&R initially highlights the printer saved with the report. The user can select another printer and port, or click the Setup button to change other print characteristics.

If this property is blank, the printer saved with the report will be used.

At design time, you can change this property in two ways:

- ❑ Double-click this property to see the Printer common dialog, which allows you to select and configure printers that have been installed on various ports on your system.
- ❑ Simply enter the printer name into the settings box.

Data Type

String

Availability

Design time; Run time

PrintFileName

Description

Specifies the name of the file to which the report is to be printed or exported.

Usage

[form.]Report.PrintFileName[= FileName$]

Example

```
Report1.PrintFileName = "c:\output\q3sales.txt"
```

« Prints the report to a file named “q3sales.txt” in c:\output. »

Comments

Use this property if you have set the Destination property to 3, 4, 5, or 7 and you want to override the saved destination.

At design time, you can change this property in two ways:

- ❑ Double-click this property to see the PrintFileName common dialog, which allows you to browse drives, directories, and files to which you have access.
- ❑ Simply enter the file name into the settings box.

Data Type

String

Availability

Design time; Run time

Query

Description

Specifies which, if any, query to use when the report is run.

Usage

[form.]Report.Query [= QueryOption%]

Example

```
Report1.Query = 2
```

« Ignores the query in the report and uses the query expression in the Filter property in place of it. »

Comments

The optional Query property allows you to control whether a query is applied to the report. Query can have one of four values:

0 – Saved. Run the report using the query saved with it, if any. The expression in the Filter property will be ignored and the report will be run exactly as it was saved.

1 – Entire. Run the entire report, ignoring any query saved in the report or contained in the Filter property.

2 – Override. Override the saved query with the expression in the Filter property. The report will be run with the records selected by the Filter property expression.

3 – Prompt user. Display a dialog box allowing the user to enter a query expression or edit the query saved with the report. If no query was saved with the report, the Insert Condition dialog will display, as shown in Figure 4.8.

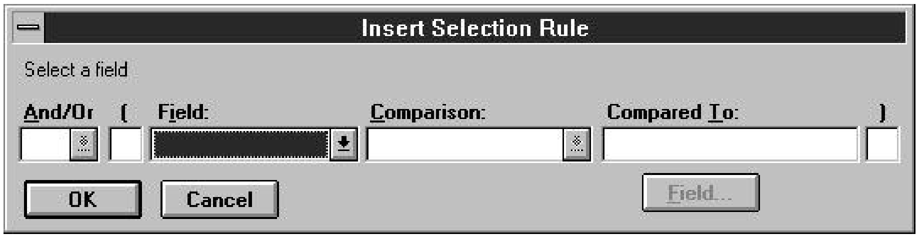
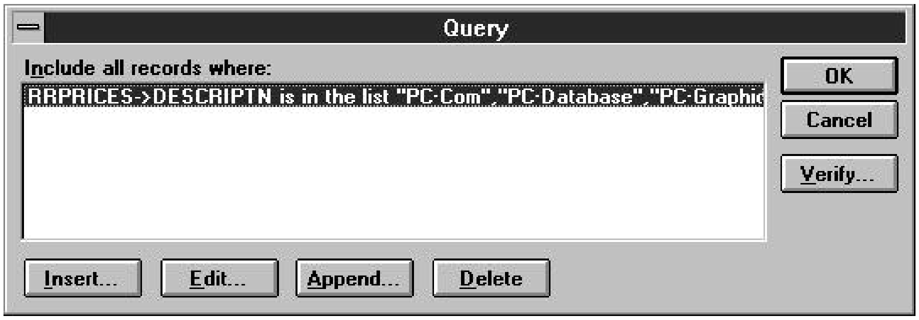


Figure 4.8 Insert Condition Dialog Box

If a query was saved with the report, the Query dialog box will display, as shown in Figure 4.9.



RRPRICES->DESCRIPTN is in the list "PC-Com","PC-Database","PC-Graphic

Figure 4.9 Query Dialog Box

When you set Query to 3, the value of the Filter property is always ignored.

Data Type

Integer (Enumerated)

Availability

Design time; Run time

RelatedTables

Description

Specifies related tables to override those saved with the report.

Usage

[form.]Report.RelatedTables(ArrayIndex)
[= Alias$=TableName$,<IndexName$>,<TagName$>]

Example

```
Report1.RelatedTables(0) = "FIRST=c:\q2\first.dbf"
Report1.RelatedTables(1) = "SECOND=c:\q2\second.dbf"
```

« Changes the first and second related tables in the report. »

Comments

These properties are optional. If you do not specify any related table overrides, the Runtime uses the tables saved with the report. It searches for these tables using the rules explained in Chapter 7.

When setting this property at run time, use a separate line of code for each change. Up to nine (9) different related tables may be specified (RelatedTables(0) – RelatedTables(8)).

At design time, you can change this property array in two ways:

❑ Double-click this property to see the dialog box shown in Figure 4.10, which contains a list of related tables in the report. This is the preferred method, since it is easier and minimizes the possibility of syntax errors.

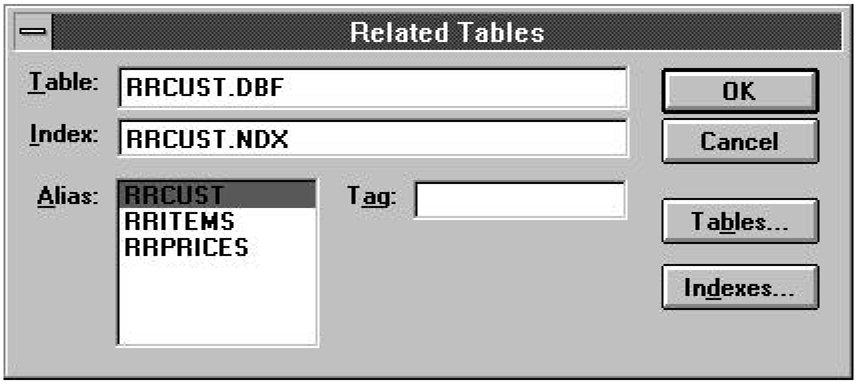


Table: RRCUST.DBF
Index: RRCUST.NDX
Alias: RRCUST
RRITEMS
RRPRICES
Tag:

Figure 4.10 Related Tables Dialog Box

❑ Enter the related table entries separated by semicolons. If you want to override some related tables, but not all of them, you must use a semicolon as a place-holder. For example, to change the first and third related table, you would enter:

"FIRST=c:\mis\first.dbf;;THIRD=c:\mis\third.dbf".

Data Type

Array of strings

Availability

Design time; Run time

ReportDirectory

Description

Specifies a default directory where the Runtime may look for the report library specified in ReportLibrary.

Usage

[form.]Report.ReportDirectory[= DirectoryName$]

Example

```
Report1.ReportDirectory = "c:\mis\reports"
```

« Looks for the report library in a directory called “c:\mis\reports.” »

Comments

If the report library file name specified in the ReportLibrary property does not contain full path information, then the Runtime will look for it in this directory. The default library directory you specify with this property will override any default library directory specified in the RRW.INI file.

Data Type

String

Availability

Run time

ReportLibrary

Description

Specifies the library that contains the report to be printed.

Usage

[form.]Report.ReportLibrary[= LibraryFileName$]

Example

```
Report1.ReportLibrary = "c:\rrw\rrsample\rrsample.rp5"
```

« Selects the report library named "rrsample.rp5" in the c:\rrw\rrsample directory. »

Comments

This property is required. It identifies the library that contains the report. The library name can include a path. The .RP5 extension is optional.

If you don't include a path, the Runtime searches for the file in the default library directory specified in the ReportDirectory property. If no default is specified in ReportDirectory, the Runtime searches for the library in the default directory specified in the RRW.INI file. If RRW.INI is not present and no default library directory is specified, the Runtime searches for the library in the current directory.

If you leave this property blank or if the library you specify cannot be found or read, the Runtime will return error status and, optionally, display an error message box (see DisplayError).

At design time, you can change this property in two ways:

- ❑ Double-click this property to see the ReportLibrary common dialog, which allows you to select a report library file and browse drives, directories, and files to which you have access.
- ❑ Simply enter the file name into the settings box.

Data Type

String

Availability

Design time; Run time

ReportName

Description

Specifies the name of the report to be printed.

Usage

[form.]Report.ReportName[= ReportName$]

Example

```
Report1.ReportName = "Order Invoice"
```

« Selects the report named "Order Invoice." »

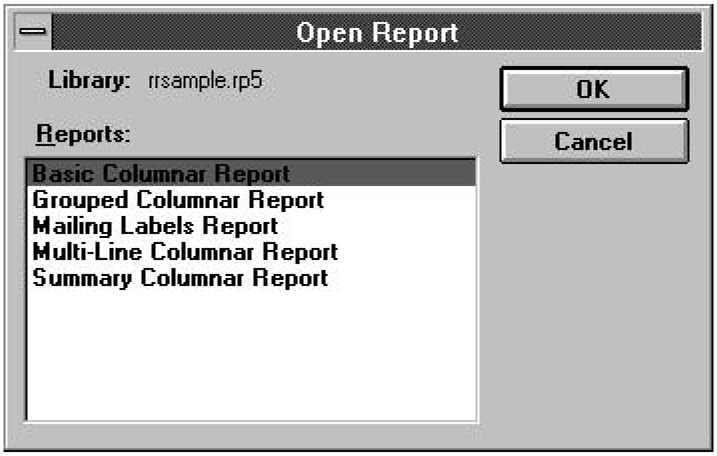
Comments

This property is required (unless ReportPick is set to 1 or 2). It contains the name under which the report was saved. The report you select must be in the library specified by the ReportLibrary property.

If you leave this property (and the ReportPick property) blank or if the report you select cannot be retrieved, the Runtime will return error status and, optionally, display an error message box (see DisplayError).

At design time, you can change this property in two ways:

- ❑ Double-click this property to see the dialog box shown in Figure 4.11, which contains a list of reports in the report library specified in the ReportLibrary property.



Open Report

Library: rrsample.rp5

Reports:

Basic Columnar Report
Grouped Columnar Report
Mailing Labels Report
Multi-Line Columnar Report
Summary Columnar Report

Figure 4.11 Open Report Dialog Box

- ❑ Simply enter the report name into the settings box.

Data Type

String

Availability

Design time; Run time

ReportPick

Description

Allows the user to pick one or more reports to be printed from a list of reports in ReportLibrary.

Usage

[form.]Report.ReportPick [= PickOption%]

Example

```
Report1.ReportPick = 1
```

« Displays a list of reports in ReportLibrary, and prints the report when the user selects OK. »

Comments

This property is optional, and can contain one of the following values:

0 – Pick none (use report in ReportName property);

1 – Pick one (allow user to select one report);

2 – Pick many (allow user to select several reports).

If you set this property, you do not need to set the ReportName property; if you include both ReportPick and ReportName values, Runtime ignores the ReportName.

To have the Runtime prompt the user to select a succession of reports, set this property to 2 . When the value is 2, Runtime will prompt the user to select a report from the library specified in ReportLibrary. After Runtime executes the selected report, the user will then be prompted to select another report. This prompt for report selection will repeat after each report until the user selects Cancel.

Set this property to 1 to prompt the user to select just one report from the library specified in ReportLibrary. When the value is 1, Runtime will prompt the user to select a report (as with the 2 value), but will not prompt for an additional report selection after the report has been executed.

Data Type

Integer (Enumerated)

Availability

Design time; Run time

Scope

Description

Specifies the scope of records to use when the report is printed.

Usage

[form.]Report.Scope [= ScopeOption%]

Example

```
Report1.Scope = 2
```

« Ignores the low and high scope saved in the report and uses the scope expressions in the LowScope and HighScope properties in place of them. »

Comments

This property is optional. Scope allows you to control the range of master table records that should be included in the report. You can specify a range of record numbers or index key values, ignore the scope saved with a report, or prompt the user to enter a range at run time. When you specify scope values, the Runtime reads only the records in the master table whose record number or index key is within the specified range. You can often speed up a report by using scope values with a master index. If you omit this field or leave it blank, the Runtime uses the saved scope values.

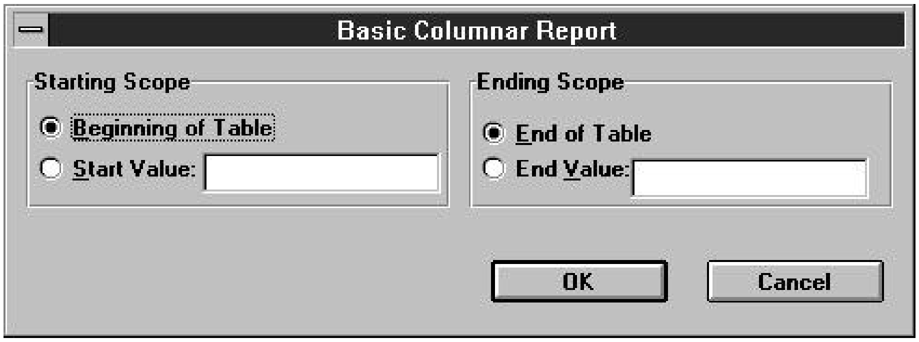
Scope can contain one of four values:

0 – Saved. Use the scope values saved with the report.

1 – Entire. Ignore any scope values.

2 – Override. Override the saved scope values with the values in the LowScope and HighScope properties.

3 – Prompt user. Allow the user to enter or change scope values at run time. When Scope is set to 3, the dialog box shown in Figure 4.12 displays. If WindowTitle is specified, the title bar will contain the WindowTitle. If WindowTitle is blank or missing, the title bar will contain the report name.



Basic Columnar Report

Starting Scope

● Beginning of Table

○ Start Value:

Ending Scope

● End of Table

○ End Value:

Figure 4.12 High and Low Scope Dialog Box

If Scope is set to 2, the LowScope property specifies the starting value of the scope and the HighScope property specifies the ending value of the scope. If Scope is set to 0, 1, or 3, the Runtime ignores LowScope and HighScope.

Each scope value can contain either a record number or an index key value up to 250 characters wide. If no master index was saved with the report (or added using the MasterIndex property), the Runtime assumes the value is a record number. Otherwise, the Runtime assumes the value is a key value in the master index. In this case, the report begins reading the master table at the first record greater than or equal to the LowScope value and stops reading the master table after the last record found that is equal to or less than the HighScope value.

The range fully includes the end points. In other words, if you enter A as the low value and M as the high value, the Runtime reads the first record in which the value begins with A through the last record in which the value begins with M. For example, if you have a customer table indexed on last name and you want to print invoices for all customers whose name begins with a letter between A and M, enter 2 in Scope, the letter A in LowScope, and the letter M in HighScope.

All scope values must be character strings. Note that a date scope value must be in the format mm/dd/yy or mm/dd/yyyy.

Data Type

Integer (Enumerated)

Availability

Design time; Run time

SortFields

Description

Specifies the field(s) that are to be used to sort your data when the report is printed.

Usage

[form.]Report.SortFields(ArrayIndex)[= "+|-SortField$"]

Example

```
Report1.SortFields(0) = "+CUST.LNAME"
```

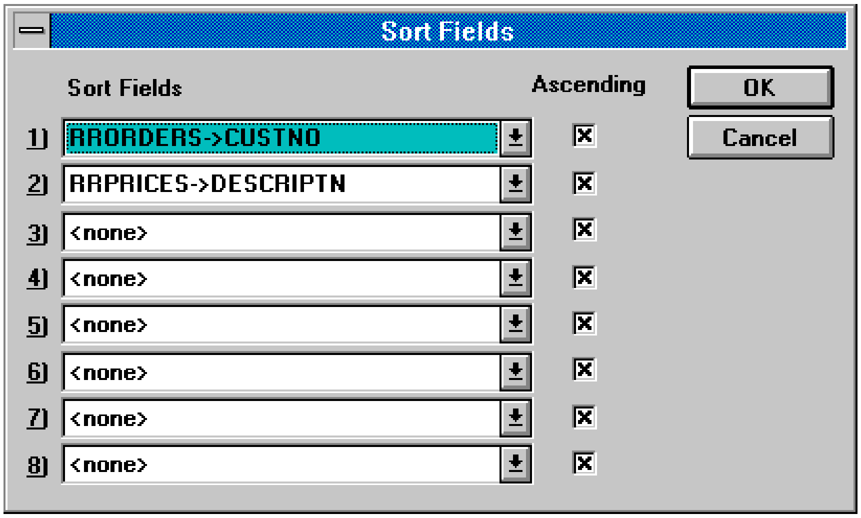
Comments

Sort fields can be database fields, calculated fields or total fields.

When setting this property at run time, use a separate line of code to specify each sort field. The first sort field you specify must be assigned array index 0, the second sort field must be assigned array index 1, etc. The index values you assign must be continuous; no gaps are allowed (0,1,2 would be correct, but 0,1,3 would be wrong).

At design time, you can change this property array in two ways:

- Double-click this property to see the dialog box in Figure 4.13, which contains a list of sort fields in the report. Clicking on the down arrow next to each group field will drop down a list of all fields used in the report from which you can select.



	Sort Fields	Ascending
1)	ARORDERS->CUSTNO	☒
2)	ARPRICES->DESCRIPTN	☒
3)	<none>	☒
4)	<none>	☒
5)	<none>	☒
6)	<none>	☒
7)	<none>	☒
8)	<none>	☒

Figure 4.13 Sort Fields Dialog Box

❑ Enter the sort field names separated by semicolons. If you want to override some sort fields, but not all of them, you must use a semicolon as a place-holder. For example, to change the first and third sort field, you would enter “Division;;Region”.

Data Type

Array of strings

Availability

Design time; Run time

StartPage

Description

Specifies the page of the report to start printing.

Usage

[form.]Report.StartPage[= Page%]

Example

```
Report1.StartPage = 10
```

« Specifies that the report should start printing at page 10. »

Comments

This property is optional. The StartPage and EndPage properties allow you to override the starting and ending page numbers saved with the report. The default value for these properties is blank.

To specify page numbers, include a StartPage value, an EndPage value, or both. If you specify both, EndPage must be equal to or greater than StartPage. For example, users can restart a canceled report where it was interrupted by specifying the starting page number as the StartPage value and 999999999 as the EndPage value. To reprint one or more consecutive pages of a report, specify the page numbers in the StartPage and EndPage properties. To print just one page, specify the same page number for both properties.

Data Type

Integer

Availability

Design time; Run time

StatusFileName

Description

Specifies the name and, optionally, the path for the Runtime status file.

Usage

[form.]Report.StatusFileName[=StatusFileName$]

Example

```
Report1.StatusFileName = "C:\TEMP\STATUS.TXT"
```

« Writes the status information into a file named STATUS.TXT in the TEMP directory on drive C. »

Data Type

String

Availability

Design time; Run time

TestPattern

Description

Specifies whether or not to print a test pattern showing the layout of the report on the page.

Usage

[form.]Report.TestPattern[= {True|False}]

Example

```
Report1.TestPattern = True
```

« Specifies that a test pattern of the report should be printed. »

Comments

This property is optional, and can be either True or False. True means to display a prompt before printing the report to allow the user the option of printing a test pattern. False means don't offer a choice to print a test pattern.

A test pattern is useful for aligning forms in the printer. The user can print the test pattern as many times as necessary and then print the report. If you enter **True**, the Runtime displays a box containing OK, Cancel, and Print buttons. The user can select OK and print as many test patterns as necessary to align the forms. Once the forms are aligned, the user can select Print to begin printing the actual report.

Data Type

Integer (Boolean)

Availability

Design time; Run time

WindowBorderStyle

Description

Specifies the type of border for the preview window.

Usage

[form.]Report.WindowBorderStyle[= BorderStyle%]

Example

```
Report1.WindowBorderStyle = 2
```

« Sets a sizable border style for the preview window. »

Comments

Set this property to one of the following border styles if you are printing to a preview window (if Destination = 1).

0 – None (a window with no border);
1 – Fixed Single (a window of a fixed size with a single line border);
2 – Sizeable (a window that can be resized by the user);
3 – Fixed Double (a window of fixed size with a double line border).

Data Type

Integer (Enumerated)

Availability

Design time; Run time

WindowControlBox

Description

Specifies whether the preview window is to have a control (system menu) box in the upper left corner when the report is printed to a preview window.

Usage

[form.]Report.WindowControlBox[= {True|False}]

Example

```
Report1.WindowControlBox = True
```

« Specifies that a control box (system menu) is to appear in the preview window. »

Comments

Set this property to True if you are printing to a preview window (if Destination = 1) and if you want the window to contain a control box.

Data Type

Integer (Boolean)

Availability

Design time; Run time

WindowHeight

Description

Sets the height of the window when the report is printed to a preview window.

Usage

[form.]Report.WindowHeight[= Height%]

Example

```
Report1.WindowHeight = 300
```

« Sets the height of the preview window to 300 pixels, or about 3 inches on most displays. »

Comments

The value for this property is expressed in pixels. Set this property if you are printing to a preview window (if Destination = 1).

Data Type

Integer

Availability

Design time; Run time

WindowLeft

Description

Sets the distance, in pixels, that the preview window is to appear from the left edge of the screen.

Usage

[form.]Report.WindowLeft[= Distance%]

Example

```
Report1.WindowLeft = 100
```

« Sets the left edge of the preview window 100 pixels from the left edge of the screen, about one inch on most displays. »

Comments

The value for this property is expressed in pixels. Set this property if you are printing to a preview window (if Destination = 1).

Data Type

Integer

Availability

Design time; Run time

WindowMaxButton

Description

Specifies whether the preview window is to have a maximize button when the report is printed to a preview window.

Usage

[form.]Report.WindowMaxButton[= {True|False}]

Example

```
Report1.WindowMaxButton = False
```

« Specifies that no Maximize button is to appear in the preview window. »

Comments

Set this property to True if you are printing to a preview window (if Destination = 1), and you want the window to contain a maximize button.

Data Type

Integer (Boolean)

Availability

Design time; Run time

WindowMinButton

Description

Specifies whether the preview window is to have a minimize button when the report is printed to a preview window.

Usage

[form.]Report.WindowMinButton[= {True|False}]

Example

```
Report1.WindowMinButton = True
```

« Specifies that a Minimize button is to appear in the preview window. »

Comments

Set this property to True if you are printing to a preview window (if Destination = 1). and you want the window to contain a minimize button.

Data Type

Integer (Boolean)

Availability

Design time; Run time

WindowParentHandle

Description

Specifies the handle of the window whose child you wish the preview window to be.

Usage

[form.]Report.WindowParentHandle[= ParentHandle%]

Example

```
Report1.WindowParentHandle = Form1.hWnd
```

« Sets the WindowParentHandle to the handle of Form1. »

Comments

This is a run-time-only property. Set this property if you are printing to a preview window (if Destination = 1), and you want the preview window to be a child of another window. To make the preview window a child of the main Runtime window, simply set this property to 0 or do not specify a value for this property.

Data Type

Integer

Availability

Run time (read and write)

WindowTitle

Description

Specifies the title you want to appear in the preview window title bar when the report is printed to a preview window.

Usage

[form.]Report.WindowTitle[= Title$]

Example

```
Report1.WindowTitle = "Revenue Summary"
```

« Sets the title of the preview window (the string that appears on the title bar) to “Revenue Summary.” »

Comments

This property is optional. Set this property if you are printing to a preview window (if Destination = 1), to specify a report title (for example, “Quarterly Profits”) to be displayed in the following places:

- ❑ The Title Bar of the Preview window;
- ❑ The Print Status window (if DisplayStatus = True);
- ❑ The title bars of the dialog box that displays when the Printer, Port, Destination, or Scope value is a question mark.

If this property is blank, the report name will be used for the title.

When setting this property at run time, make sure that the title is enclosed in quotes.

Data Type

String

Availability

Design time; Run time

WindowTop

Description

Sets the distance, in pixels, that the preview window is to appear from the top edge of the screen.

Usage

[form.]Report.WindowTop[= Distance%]

Example

```
Report1.WindowTop = 100
```

« Sets the top edge of the preview window 100 pixels from the top of the screen, or about one inch on most displays. »

Comments

The value for this property is expressed in pixels. Set this property if you are printing to a preview window (if Destination = 1).

Data Type

Integer

Availability

Design time; Run time

WindowWidth

Description

Specifies the width of the preview window in pixels.

Usage

[form.]Report.WindowWidth[= Width%]

Example

```
Report1.WindowWidth = 500
```

« Specifies a preview window 500 pixels wide, or about five inches on most displays. »

Comments

The value for this property is expressed in pixels. Set this property if you are printing to a preview window (if Destination = 0).

Data Type

Integer

Availability

Design time; Run time

WriteAllow

Description

Controls whether database users can modify the tables and indexes in use by reports.

Usage

[form.]Report.WriteAllow[={True|False}]

Example

```
Report1.WriteAllow=False
```

« Prevent other applications from writing to the databases in use by the report. »

Data Type

Integer (Boolean)

Availability

Design time; Run time

XbaseEditor

Description

Specifies the type of memo editor that was used to create database memos for reports.

Usage

[form.]Report.XbaseEditor[={True|False}]

Example

```
Report1.XbaseEditor = False
```

« Memo fields used in the report have been created using a memo editor other than the standard Xbase memo editor. »

Data Type

Integer (Boolean)

Availability

Design time; Run time