One Design
One Server
One User Experience

Developing Reports

using e.Report Designer Professional

This documentation has been created for software version 11.0.5.
It is also valid for subsequent software versions as long as no new document version is shipped with the product or is published at https://knowledge.opentext.com.

Open Text Corporation
275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1
Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Copyright © 2017 Actuate. All Rights Reserved.
Trademarks owned by Actuate
"OpenText" is a trademark of Open Text.

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

Document No. 170215-2-130301 February 15, 2017

Contents

About *Developing Reports using e.Report Designer Professional* xxi

Part 1
Getting started

Chapter 1
Building your first report 3
Working with Actuate e.Report Designer Professional 4
Preparing to create your first report 4
Preparing to access a data source 4
Making report components available to your report 5
Setting up the report design environment 6
Creating a report that groups data 10
Opening the listing report wizard 10
Choosing a data source for a listing report 11
Selecting tables and columns for a listing report 13
Organizing the data by table 14
Adding summary data 16
Sorting the data 17
Selecting a layout style for a listing report 17
Saving the report design 18
Viewing the query 21
Understanding report structure 23
Enhancing a report design 25
Running a report 27

Chapter 2
Understanding report design 29
Overview of the design process 30
Understanding report design concepts 32
About components 32
About component relationships 33
About reusable components 34
About the report design environment 34
About the application window elements 36
Menus 36
Toolbars 36
Work area 36
Status bar 36

Managing workflow 37
About the design perspective 38
About the data perspective 39
About the debug perspective 39
About the view perspective 39
Personalizing the application window 41
Moving, minimizing, and stacking windows 41
Closing and restoring windows in a perspective 43
Moving, hiding, and changing toolbars 43
Restoring a perspective to its default layout 44
About the main design tools 44
About the Layout window 44
About the Report Structure window 44
Performing common operations 46
Placing components in a report design 46
Naming a report component 47
Searching for a component 48
Understanding a component search regular expression 50
About the results of a component search 52
Examining class hierarchies 55
Setting a component's properties 56
Copying components 57
Deleting components 58
Saving, running, and previewing a report 58
Building the report 58
Running the report 59
Viewing the report 59
Closing a report 59
Going beyond the graphical tools 60
About object-oriented design 60
Customizing reports through programming 61

Part 2
Designing a report

Chapter 3
Planning and creating a report 65
Planning the report 66
Deciding the content of the report 66
Determining how the report will be viewed 67
Deciding if the report will be used by international users 67
Deciding the layout and format of the report 68

Designing a prototype on paper . . . 68
Deciding how to start a new report . . . 69
Using a wizard . . . 69
About the Quick Report wizard . . . 69
About the Listing Report wizard . . . 70
Starting with a template . . . 71
Starting with a blank report design . . . 72
Configuring report development resources . . . 72
Creating a report without a wizard . . . 72
Starting a new report design . . . 73
Accessing data . . . 73
Setting up the connection . . . 73
Specifying the data to use in the report . . . 74
Grouping and sorting data . . . 75
Placing data . . . 75
Summarizing data . . . 76
Formatting data . . . 76
Specifying a report title . . . 76
Setting page size, orientation, and printable area . . . 76
Naming an Actuate report . . . 77
Performing other report design tasks . . . 78

Chapter 4
Laying out a report . . . 79
About laying out a report . . . 80
Preparing to lay out a report . . . 81
Understanding the layout process . . . 81
Grouping report data . . . 81
About sorting and sort keys . . . 83
Creating a group . . . 83
Examining how groups appear in the report structure . . . 87
Grouping report data by intervals . . . 88
Grouping string data by intervals . . . 90
Grouping numeric data by intervals . . . 91
Grouping date and time data by intervals . . . 91
Deciding which tool to use to group data by interval . . . 91
Grouping on a custom key . . . 93
Placing data in the report . . . 94
Deciding where to place data . . . 96
Placing data in a report . . . 97
Placing database fields . . . 99
Placing computed fields . . . 100
Associating data with controls . . . 104

Changing a control's value 105
Changing a control's class 105
Adding column headings 106
Adding group headings 109
Placing group headings 110
Specifying expressions for group headings 110
Adding aggregate information 114
Examining sample aggregates 114
Refining aggregate functions 118
Using a lookahead aggregate in a conditional section or sequential section 120
Inserting a page break before or after each group 122
Changing the placement of the report title 123
Sorting report data 124
Sorting data at the group level 124
Sorting data at the detail level 125
Enhancing the appearance of a report 126
Testing a report design 126
Storing report components in the scratch pad 127
Testing the layout 128
Verifying values of computed fields 128
Verifying results of groups and sorts 129
Checking the running headers and footers 129
Correcting a report design with missing classes 129
Checking the structure of a report design 129

Chapter 5
Formatting report content 131
About formatting 132
Arranging components 134
Selecting multiple components 134
Moving components 135
Sizing components 136
Aligning components 137
Arranging components relative to the frame 138
Layering components 139
Formatting data 140
Formatting numeric data 141
Formatting date and time data 142
Formatting text data 143
Using code mapping to control the appearance of text 144
Specifying how data appears if its value overflows the control size 144
Adjusting the spacing between rows of data 146
Adding borders and colors to components 147

Creating a drop shadow effect within a frame 148
Adding shapes and lines 150
Adding images 151
Planning to use an image file 152
Inserting an image in a report 152
Inserting an image using a database reference 154
Formatting report content based on conditions 154
Applying conditional formatting 155
Inheriting conditional formatting 159

Chapter 6
Customizing page layout 161
About page layout 162
Understanding the relationship between page layout and report layout 163
Viewing the page layout 163
About the page layout components 165
Planning the page layout 167
Specifying page size, orientation, and margins 167
Specifying a fixed page size 168
Specifying a dynamic page size 169
Setting page orientation 170
Setting page margins 172
Understanding the effects of page size on online and printed reports 173
Comparing online and printed output for reports with fixed page size 173
Comparing online and print output for reports with dynamic page size 174
Printing a report with large report pages 176
Specifying margins for the printed report 177
Specifying headers and footers 179
Comparing headers and footers in page layout and report layout 180
Specifying page numbers 181
Using a standard page number format 181
Using a custom page number format 182
Specifying a date and time stamp 182
Designing background patterns 183
Adding color and borders to flows and pages 183
Adding a watermark 184
Designing a multicolumn page layout 185
Designing multiple page layouts 187
Creating a left and right page layout 187
Creating a title and body page layout 190

Chapter 7
Understanding charting concepts 195
About Actuate charts 196

Recognizing the parts of a chart 196
Understanding types of charts 198
About area charts 199
About bar charts 199
About bubble charts 200
About line charts 200
About pie charts 201
About scatter charts 202
About step charts 203
About stock charts 203
Understanding chart layers 204
Using an overlay chart layer 204
Using a study chart layer 205
Understanding chart data 206
Specifying data for an area, bar, line, or step chart 207
Specifying data for a pie chart 208
Specifying data for a scatter chart 210
Specifying data for a stock chart 212
Specifying data for an overlay or study chart layer 213
Using a chart hyperlink 214

Chapter 8
Displaying data in a chart 217
Creating a chart 218
Inserting a chart control 218
Selecting a chart type 218
Specifying chart data 219
Sorting chart data 229
Grouping chart data 230
Setting up a group 230
Adding missing groups 232
Adding a static group 235
Modifying a chart 237
Working with chart titles 238
Formatting the chart area 239
Working with a chart legend 241
Working with axes 243
About the x-axis 243
About the y-axis 244
Formatting an axis 244
Working with axis scale 246
Working with axis labels 247
Working with series and categories 250

Modifying series labels or series color 250
Working with point labels 252
Positioning data points and axis labels on an x-axis 255
Working with an empty value in a series 256
Displaying data point markers or series lines 257
Working with up or down bars 258
Working with high-low lines 259
Working with drop lines 259
Working with trendlines 260
Working with bars in a chart 265
Working with pie chart sectors 267
Using a chart hyperlink 269
Working with sample chart data 270
Rendering a chart on Windows and UNIX 271

Chapter 9
Drawing with SVG 273
Working with SVG 274
Supporting SVG in e.Report Designer Professional 275
Understanding measurements and scaling 276
Using drawing planes 277
Drawing on charts 278
Utilizing chart information 281
Using text metrics 283
Examining the SVG code in a drawing control 283
Examining the SVG header information 286
Examining defined referenced elements 287
Examining drawing functions 288
Debugging an SVG program 289
Examining the DebugSVG control 290
Using the DebugSVG control 292

Chapter 10
Presenting data in a cross tab 295
About cross tabs 296
Comparing a cross tab to a listing report 297
Understanding cross tab limitations 298
Placing a cross tab in a report 299
Understanding the effects of placing a cross tab in different frames 299
Understanding the effects of placing a cross tab in different sections 299
Placing multiple cross tabs in a frame 302
Building a cross tab 303
Grouping data by more than two fields 308

Displaying multiple aggregate fields 310
Displaying computed values in row and column headings 312
Displaying static text in row and column headings 312
Limiting the amount of data the cross tab displays 314
Sorting cross tab data 315
Grouping cross tab data by intervals 316
Adding missing groups 318
Specifying the range of data to use when adding missing groups 321
Customizing the appearance of a cross tab 325
Resizing cross tab rows and columns 326
Formatting borders of a cross tab 326
Formatting numbers in a cross tab 329
Adding color to a cross tab 330
Understanding cross tab sizing 331
Printing a cross-tab report 331
Changing the margins for the printed pages 332
Setting print options to join multiple pages 333
Eliminating blank space on the right of the cross tab 334

Chapter 11
Displaying variable-length text data 337
Displaying memo field data 338
Using dynamic text controls 338
Comparing a dynamic text control with a text control 339
About the supported text formats 339
Using full text justification in a dynamic text control 341
Considering performance issues 341
Placing a dynamic text control in a report design 341
Developing content for a dynamic text control 342
Building complex HTML-tagged text 344
Changing text size 344
Creating HTML lists 344
Building complex RTF-tagged text 344
Displaying Unicode characters in RTF-tagged text 345
Indenting RTF-tagged text 345
Using fonts and colors in RTF-tagged text 345
Modifying the default resizing behavior 347
Shrinking a dynamic text control 347
Moving a control up when a dynamic text control above it shrinks 348
Setting the maximum height of a dynamic text control 348
Increasing the width of a dynamic text control 348
Setting the maximum width of a dynamic text control 349
Using dynamic text controls in nested frames 350

Using dynamic text controls in different types of frames 351
Using dynamic text controls in Content, Before, and After frames 351
Using dynamic text controls in PageHeader frames 351
Using dynamic text controls in PageFooter frames 351
Adjusting text formats for different rendering environments 352
Adjusting the space on the right of the text 352
Adjusting the space between lines of text 353
Adjusting the space between paragraphs 355
Adjusting the space between tabbed text 356
Controlling how a frame and its contents fit in multiple pages 358
Specifying how a frame and its contents should split 358
Splitting frames that contain dynamic text controls 359
Splitting frames that do not contain dynamic text controls 360
Specifying an area of the frame that should not be split 361
Setting margins for the segments of a split dynamic text control 362
Setting margins for the segments of a split frame 365
Controlling how lines in a paragraph break 365
Understanding how dynamic text controls affect report layout 365
Building a report with dynamic text controls 366
Solving common output problems 372

Chapter 12
Formatting a report with dynamically sized components 375
About dynamically sized components 376
Formatting controls in a frame that contains a dynamically sized component 376
Understanding how Actuate software resizes and repositions components 377
Understanding vertical resizing and repositioning 378
Understanding horizontal resizing and repositioning 380
Modifying the default resizing and repositioning of controls 382
Resizing a control to match the height of a dynamically sized component 383
Resizing a control to match the width of a dynamically sized component 384
Resizing a control relative to the height of the frame 384
Resizing a control relative to the width of the frame 386
Repositioning a control vertically relative to the frame 387
Repositioning a control horizontally relative to the frame 388
Modifying how pages resize 390
Enforcing a fixed page size 391
Resizing a page to fit the contents of a dynamic text control 392
Solving common output problems 394

Chapter 13
Building subreports and conditional sections 399
About building a report that contains subreports 400

Setting up the structure of the main report 400
About the components that define a report's structure 401
Placing section components 402
Placing components in a report section 404
Placing components in a sequential section 404
Placing components in a parallel section 404
Placing components in a conditional section 405
Specifying the condition for a conditional section 405
Using an aggregate function in a conditional section or sequential section 406
Creating a page layout for a subreport 406
Using a subpage 407
Creating a multiflow page layout for parallel subreports 408
Building a report with side-by-side subreports 410
Building a report with sequential subreports 417
Building a report with linked subreports 421
Building a report with conditional sections 427
Examining a report containing multiple sections and subreports 433

Chapter 14
Adding interactive viewing features 439
About interactive viewing events 440
Providing balloon help 440
Providing search functionality in a report 440
Naming a component in a search 442
Creating versatile searches 443
Searching for multiple instances of the same class 444
Searching for multiple classes 445
Retrieving data from controls in multiple parallel or sequential sections 446
About page headers and page footers in searches 447
Designing searchable reports 448
About search analysis 448
Searching reports with parallel and sequential sections 452
Developing hyperlink functionality 453
About LinkExp syntax 454
About destination URL syntax 455
About search expression syntax 457
About dynamic report parameters syntax 459
Creating a hyperlink that uses a dynamic parameter value 462
Calling JavaScript functions with LinkExp 464
Defining universal hyperlinks 464
Designing a hyperlink URL 465
Developing a table of contents 466
About table of contents properties 466

Developing a table of contents for a grouped report 466
Developing a table of contents for other reports 468

Chapter 15
Designing a report for multiple platforms and environments 471
Designing a report to display and print data across platforms 472
Understanding viewing and printing requirements 472
Designing for the delivery environment 473
Designing for overlapping controls 473
Designing for the print environment 473
Understanding font issues 473
Getting font information on Windows and UNIX servers 474
Getting font information on desktop-only environments 475
About the master fonts file 475
Embedding font information in a report object executable file 476
Defining custom fonts 476
Using a custom font file in a desktop environment 479
Designing with fonts on multiple platforms 479
Designing a report for multiple viewing environments 480
Displaying different controls in DHTML and PDF output 480
Setting whether a control appears while printing or viewing 481

Chapter 16
Debugging a report 483
About debugging 484
Understanding the types of errors 484
About compilation errors 484
About run-time errors 485
About logic errors 485
About e.Report Designer Professional's debugging tools 485
About the Debug menu 486
Viewing errors in the Output window 487
About Variables 488
About Call Stack 489
About the Watch window 491
Saving parameters for successive debugging runs 491
Starting a debugging session 492
Controlling code generation 494
Running to a breakpoint 494
Stepping through code 495
Resuming code generation 495
Stopping code generation 496
Examining and modifying variable values 496

Tracing object reference variables 496
Monitoring and changing the run-time value of a variable 498
Inserting and arranging variables in Watch 498
Editing a variable value 499

Part 3
Displaying multiple reports from a single report design

Chapter 17
Designing for ad hoc reporting 503
About ad hoc reporting 504
About parameters 504
Understanding how a report prompts the user for input 506
Planning to use parameters 507
About ad hoc parameters 507
About static parameters 508
Specifying how parameters appear to the user 509
Specifying a parameter's display name, group, and default value 510
Providing the user with a list of choices 511
Typing a list of parameter values 512
Creating a query that retrieves a list of parameter values 513
Providing the user with true or false options 518
Specifying the order in which parameters appear 519
Hiding a parameter 520
Deleting parameters 521
Testing parameters 523
Saving and reusing parameter values 524
Suppressing parameter prompting 526
Verifying parameter values that the user supplies 526

Chapter 18
Controlling user access through page-level security 529
About page-level security 530
About security IDs 531
Defining page-level security rules 531
Understanding cascading security rules for nested sections 532
Designing a report that uses page-level security 532
Displaying page numbers in a report that uses page-level security 534
Testing a report that uses page-level security 534
Setting up a report design to test page-level security 535
Testing page-level security 538
Creating a virtual security ID 541

Customizing an access control list 543
Controlling access to subsections 543
Setting CascadeSecurity 543
Overriding GetFullACL( ) 544
Generating a custom ACL 544
Generating a custom ACL using SetSecurity( ) 544
Generating a custom ACL using an Actuate Basic function 545

Chapter 19
Designing a master report to generate multiple reports 547
About report bursting 548
Understanding report bursting techniques 548
Understanding report bursting tasks 549
Examining report bursting examples 549
Examining a subreport bursting report 550
Examining the BuildFromRow( ) method for a subreport bursting report 551
Examining the SuggestRoiName( ) method for a subreport bursting report 551
Examining the summary entry's LinkExp property for a subreport bursting report 551
Examining a group bursting report 552
Examining the SuggestRoiName( ) method for a group bursting report 552
Examining the summary entry's LinkExp property for a group bursting report 553

Part 4
Optimizing report development tasks for the enterprise environment

Chapter 20
Designing reusable components 557
About designing reusable components 558
Working with custom properties 558
Creating, editing, and deleting a custom property 558
Creating a property 561
Editing a custom property 563
Deleting a property 564
Setting a property as a value expression 564
Filtering a property 565
Working with structures 566
Working with enums 569

Chapter 21
Publishing reusable components in a library 571
Understanding component and function libraries 572

About component libraries 572
About function libraries 572
About included and available libraries 573
About Library Organizer 574
Planning a component library 574
Tailoring the library to the task 575
Optimizing report performance 575
Creating a component library 575
Creating a function library 578
Including a component or function library in a report design 579
Placing a library component in a report design 584
Creating a component reference 584
Creating a subclass of a library component 585
Publishing a component to a library 586
Providing access to a library using a configuration file 588
Including or making available a single library 589
Making all libraries in a directory available 590
Protecting a component or function library from modifications 591

Chapter 22
Publishing a report ... 593
About publishing a report 594
Creating an Actuate iServer profile 594
Setting publication options 597
Managing file versions 597
Copying file properties 597
Publishing a report to an Encyclopedia volume 598

Chapter 23
Generating information objects from Actuate Basic report designs ... 601
Generating an information object from an Actuate Basic report design 602

Chapter 24
Customizing the design environment ... 607
About design environment customization 608
Setting design environment options 608
Setting a font in the designer 609
Setting the default font for labels, data, and source editor 610
Setting the default font for a chart 611
Setting the font for Report Structure 613
Setting units of measurement 613
Setting the default unit of measurement 614

Customizing unit suffixes 615
Setting design search paths 616
Setting general options in the designer 619
Customizing the length of the recently opened files list 620
Setting the Management Console URL 620
Setting Basic source encoding 621
Customizing the classes of the toolbox 621
Customizing the class association of the toolbox 623
Saving and reusing class associations of components 624
Changing the item list for Create New Report 626
Hiding a report wizard item that appears in Create New Report 627
Adding or removing a template in the item list 628

Part 5
Integrating report output

Chapter 25
Embedding report content in a web page 633
About embedding report content in a web page 634
About the source HTML for the web page example 634
About supported components for embedded report content 635
About changes to report capabilities in embedded report content 636
Identifying the report content to display in a web page 636
Displaying or hiding report content that is embedded in a web page 636
Identifying the report content to embed in a web page 637
Importing report content using a search URL 638
Importing a report page into a web page 638
Importing a report section into a web page 639
Working with a Reportlet 640
Printing and downloading a Reportlet 640
Enabling a hyperlink in a Reportlet 640

Chapter 26
Providing web functionality in a report 641
Using web browser code inside a report 642
Including custom browser code in a report design 642
About the context block 644
Clipping the output of custom browser code 645
Aligning the output of custom browser code 646
Debugging custom browser code 646
Including global custom browser code 647
Creating an HTML form 648

Using Flash within a browser scripting control 649
Generating custom browser code dynamically 650
Generating output specific to the viewing environment 650
Determining the application execution context 651
Generating output for different browsers 651
Adjusting for the current scaling factor 651
Displaying values from one or more rows in a control 652
Printing and viewing a browser scripting control in non-DHTML formats 653
Understanding an example browser scripting library for Internet Explorer 654
Designing a frame to support an HTML form 654
Overriding methods in DHTMLForm 655
Subclassing DHTMLForm 655
Designing a browser menu control 656
Using BuildFromRow( ) 657
Using OnRow( ) 657
Using Finish( ) 657
Using BrowserCode( ) 658
Creating a drop-down menu in a web browser 658
Displaying custom browser code in the view perspective 658

Chapter 27
Designing for XML output 661
About the XML data format 662
About document type definitions 662
Examining a simple XML document 662
About the View process 663
Publishing an Actuate report as XML 664
Mapping XML to an Actuate report component 665
About XML properties 666
About AcReport XML properties 667
About AcReportComponent XML properties 667
About AFC XML methods 668

Part 6
Previewing reports in Actuate e.Report Designer Professional

Chapter 28
Running a report 675
About running a report 676
Running a report locally 676
Viewing pages during report generation 677
Understanding checkpoints 678

Refreshing the report in the view perspective 678
Running a report that uses parameters 678
Specifying parameter values on Requester 678
Using a text box to set a parameter value 679
Using radio buttons or a drop-down list to set a parameter value 680
Setting values for date parameters 680
Using null values with structure and table parameters 681
Specifying a value for a structure parameter 681
Setting a value for a table parameter 681

Chapter 29
Viewing a report using the view perspective 685
Opening and viewing a report 686
Navigating a report 687
Using the paging commands 687
Going to a specific page 688
Using the table of contents 688
Using hyperlinks to navigate report data 690
Examining objects and classes 692
Personalizing the view perspective 693
Using screen arrangement options 693
About zooming 695
Copying report data to another document 696

Chapter 30
Searching for report data 699
About the search feature in Actuate reports 700
About searchable report fields 700
Identifying a searchable report field 700
Searching for report data 701
Understanding how a multifield search works 701
Creating search conditions 702
Using a search expression in the search conditions 704
Using operators in a search expression 704
Using a wildcard character in a search expression 705
Using a pattern in a search expression 706
Searching for the ?, *, #, and [ ] characters 706
Searching for a string that contains a special character 707
Using a nonprinting ASCII character in a search expression 708
Using search results to navigate through the report 708
Specifying data to display in Results 709
Creating a search definition 711

Chapter 31
Printing a report 713
Printing a report from your desktop 714
Running a report before printing 717

Chapter 32
Rendering to PDF 719
About the Render architecture 720
Working with multiple Render profiles 720
Selecting a Render profile at design time 721
Selecting a Render profile at generation time 721
Selecting a Render profile at view time 721
About the PDF Writer 721
Using the new PDF Writer in e.Report Designer Professional 721
Using the new PDF Writer in iServer 721
Enabling the Render system 721
Enabling Render in a report design 722
Disabling the new PDF Writer 722
Creating a PDF document during report generation 723
Render profile XML reference 723
FontFamily element 723
IncludeProfile element 726
Parameter element 726
ParameterCollection element 727
ParameterGroup element 728
Profile element 728
RenderProfiles element 729
Render profile parameter reference 729
AbortOnInvalidImage parameter 729
AddDebugPages parameter 729
AddTOC parameter 729
Author parameter 729
ClipText parameter 730
CompressionLevel parameter 730
DrawingDPI parameter 731
DrawingQuality parameter 731
EmbedDebugInfo Parameter 731
EnableExternalHyperlinks parameter 731
InitialView parameter 731
Generator parameter 732
MaximumNumberOfPages parameter 732
Multithread parameter 733
PickPrinterTrayByPageSize parameter 733

SplitOversizePages parameter 733
Writer parameter 733

Appendix A
Working with RTF and HTML tags in a dynamic text control 735
Working with RTF tags 736
Working with HTML tags 740

Appendix B
Upgrading reports in batches 745
Upgrading reports 746
Using the erdpro command to upgrade reports 746

Glossary 749

Index 861

About *Developing Reports using e.Report Designer Professional*

Developing Reports using e.Report Designer Professional provides information about using the e.Report Designer Professional report design tools. This manual explains how to use and customize the design environment and how to work with items such as report components, report sections, and data streams.

Developing Reports using e.Report Designer Professional includes the following chapters:

- *About Developing Reports using e.Report Designer Professional.* This chapter provides an overview of this guide.
- *Part 1. Getting started.* This part explains how to build a report using a report wizard and also introduces basic reporting concepts.
- *Chapter 1. Building your first report.* This chapter explains how to create a report using a report wizard.
- *Chapter 2. Understanding report design.* This chapter provides an overview of the report design process and environment. The chapter also explains basic report design concepts and tasks.
- *Part 2. Designing a report.* This part provides information about designing a report, from the planning stage to working with specialized controls such as charts and cross tabs.
- *Chapter 3. Planning and creating a report.* This chapter describes the decisions to make when planning a report and explains the basics of creating a report.
- *Chapter 4. Laying out a report.* This chapter describes the procedures for grouping, sorting, and placing data in a report.
- *Chapter 5. Formatting report content.* This chapter describes how to make a report visually appealing and effective.

- *Chapter 6. Customizing page layout.* This chapter describes how to set page attributes and design page layouts.
- *Chapter 7. Understanding charting concepts.* This chapter explains charting concepts and describes the types of charts that you can use in a report design.
- *Chapter 8. Displaying data in a chart.* This chapter describes the various chart controls and how to use them in a report design.
- *Chapter 9. Drawing with SVG.* This chapter describes how to draw with SVG, and how to include SVG drawings in charts and drawings.
- *Chapter 10. Presenting data in a cross tab.* This chapter describes how to design a cross-tab report, in which every cell displays a calculated value.
- *Chapter 11. Displaying variable-length text data.* This chapter introduces the underlying concepts that govern working with a dynamic text control.
- *Chapter 12. Formatting a report with dynamically sized components.* This chapter describes how to use dynamic text controls and how to format a report that contains dynamically sized components.
- *Chapter 13. Building subreports and conditional sections.* This chapter describes how to build reports that contain subreports and conditional sections, and provides several tutorials.
- *Chapter 14. Adding interactive viewing features.* This chapter describes how to create features that enhance the viewing experience, such as balloon help and context menus.
- *Chapter 15. Designing a report for multiple platforms and environments.* This chapter describes techniques for ensuring that a report adheres as closely as possible to its design criteria in different printing and viewing environments.
- *Chapter 16. Debugging a report.* This chapter describes how to test code and resolve errors in a report.
- *Part 3. Displaying multiple reports from a single report design.* This part explains how to use report parameters, page-level security, and report bursting techniques.
- *Chapter 17. Designing for ad hoc reporting.* This chapter describes how to create report parameters that prompt the user to provide information to determine what data to retrieve and display when the user runs the report.
- *Chapter 18. Controlling user access through page-level security.* This chapter describes how to create a report design that generates different reports depending on the report user's access privileges.
- *Chapter 19. Designing a master report to generate multiple reports.* This chapter describes how to create a single report object executable (.rox) file that generates multiple report object instance (.roi) files.

- *Part 4. Optimizing report development tasks for the enterprise environment.* This part explains how to create a library of reusable components, distribute reports throughout the enterprise, and customize your report design environment.
- *Chapter 20. Designing reusable components.* This chapter describes how to create and work with custom properties, enums, and structures.
- *Chapter 21. Publishing reusable components in a library.* This chapter describes how to create and use a library of report components.
- *Chapter 22. Publishing a report.* This chapter describes how to publish a report to an Actuate iServer Encyclopedia volume.
- *Chapter 23. Generating information objects from Actuate Basic report designs.* This chapter describes how to use the query in an Actuate Basic report design to create an information object that BIRT Studio can use as a data source.
- *Chapter 24. Customizing the design environment.* This chapter describes how to customize the e.Report Designer Professional tools.
- *Part 5. Integrating report output.* This part explains how to use design elements to simplify integrating your reports on the web and across platforms.
- *Chapter 25. Embedding report content in a web page.* This chapter describes how to access, retrieve, and display a portion of an Actuate Basic report in a web page.
- *Chapter 26. Providing web functionality in a report.* This chapter describes how to include custom browser code such as HTML in a report design.
- *Chapter 27. Designing for XML output.* This chapter describes how to design an XML report.
- *Part 6. Previewing reports in Actuate e.Report Designer Professional.* This part explains how to view, run, and print a report, and how to search for data in a report.
- *Chapter 28. Running a report.* This chapter explains how to run a report.
- *Chapter 29. Viewing a report using the view perspective.* This chapter describes the functionality of the view perspective.
- *Chapter 30. Searching for report data.* This chapter explains e.Report Designer Professional's search functionality.
- *Chapter 31. Printing a report.* This chapter explains how to print a report.
- *Chapter 32. Rendering to PDF.* This chapter explains how to use the Render architecture to output a report to PDF.
- *Appendix A. Working with RTF and HTML tags in a dynamic text control.* This appendix lists the RTF and HTML tags that dynamic text controls support.

- *Appendix B. Upgrading reports in batches.* This appendix describes how to upgrade Actuate Basic reports from an earlier release using the Command Window or a batch file.
- *Glossary.* This section provides definitions of terms used in Actuate e.Report Designer Professional product and documentation.

Part One

Getting started

Chapter

1

Building your first report

This chapter contains the following topics:

- Working with Actuate e.Report Designer Professional
- Preparing to create your first report
- Creating a report that groups data
- Enhancing a report design
- Running a report

Working with Actuate e.Report Designer Professional

A report developer uses Actuate e.Report Designer Professional to design, build, and distribute structured content. The content can be a richly formatted report, a library of reusable report components, or a template for building reports. You can use specialized report components, such as charts and cross tabs, to display data in the format of your choice. You can use page-level security to ensure that data is available only to those individuals authorized to view or work with the data. After you create and preview a report design, you can use e.Report Designer Professional to generate the output and publish the executable file to Actuate iServer.

You can create a report using a wizard or you can start with a basic report and add components in the design environment.

This chapter introduces the tasks involved in creating a report using a wizard. You learn how to:

- Configure a data source.
- Access a configuration file to make report components available to your report.
- Set up the report environment.
- Design a report that groups data from three tables.
- Enhance the appearance of the report.
- Run the report.

Preparing to create your first report

This section describes three procedures to complete before you begin building a report design. First, ensure that e.Report Designer Professional can access your data source. Optionally, you can use a configuration file to specify report components, such as database connections and component libraries, that are available to all your reports. Then, you can set up e.Report Designer Professional so that your report designs look like those in the illustrations in this chapter.

Preparing to access a data source

e.Report Designer Professional can access a variety of data sources. The procedures to set up a data source depend on the type of data source. For example, to access a database using an open database connectivity (ODBC) driver, you must configure your computer to access the database using the ODBC driver. The installation process of e.Report Designer Professional configures your

computer to use two ODBC data sources that you need to run the examples in this chapter.

For more information about configuring and accessing ODBC and other data sources, see *Accessing Data using e.Report Designer Professional*.

Making report components available to your report

Before designing a report, you can specify a configuration file. A configuration file can list the connections, data sources, component libraries, templates, and other report components that you can use in a report design. In the following procedure, you use the sample configuration file that installs with e.Report Designer Professional. This file points to a library that makes a sample database and other components available to the report that you create later in this chapter.

How to specify the sample configuration file

1. From the task bar, choose Start➛All Programs➛Actuate 11 ➛Actuate e.Report Designer Professional. e.Report Designer Professional starts. Welcome appears.
2. In Welcome, choose Cancel. A blank design window appears in the design perspective, as shown in Figure 1-1.

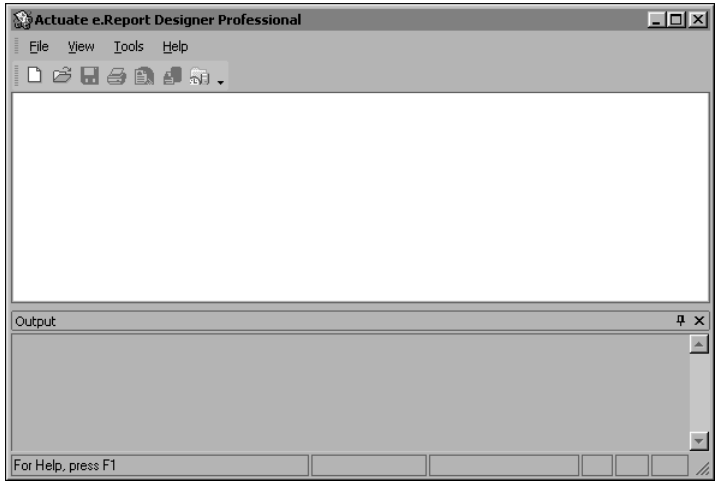


Figure 1-1 Design window

3. Choose Tools➛Options.
4. In Options—Design Editor, choose General. Options—General appears.

5. In Configuration file, choose Ellipsis.
6. In Locate configuration file, specify the configuration file:

1 Navigate to <eRDPro_HOME>\Examples\ConfigurationFile.

2 Select sample_configuration_file.xml. Choose Open. The path to the configuration file appears in Options—General, as shown in Figure 1-2.

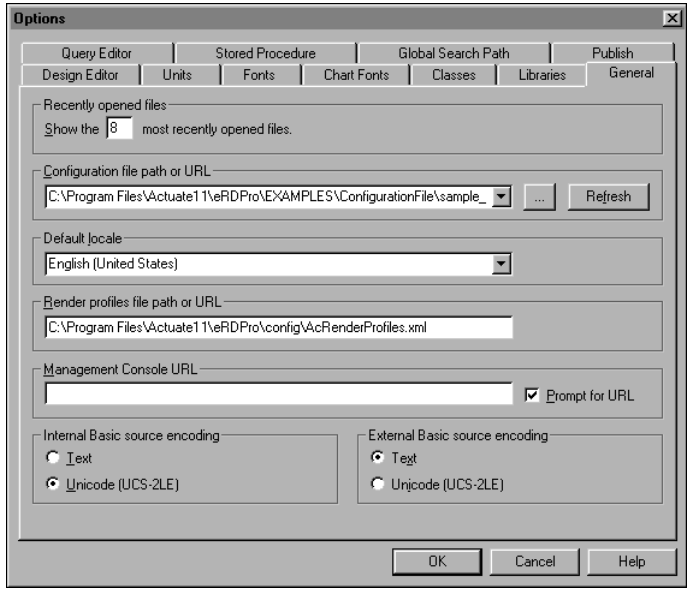


Figure 1-2 Configuration file in General

7 Choose Refresh. e.Report Designer Professional reads the configuration file. A confirmation message appears.

8 To close the message, choose OK.

9 In Options—General, choose OK. The connections and data sources in the configuration file are available to the report that you create in this chapter.

Setting up the report design environment

In these procedures, you set viewing options to ensure that your report appearance matches the illustrations in this book. These procedures are optional.

In the design perspective, the default user interface shows empty slots where report components can be added to a report design. For example, Figure 1-3 shows the Report Structure window, showing empty slots. Empty slots are locations in the report design available to have components placed within them that are appropriate to the container.

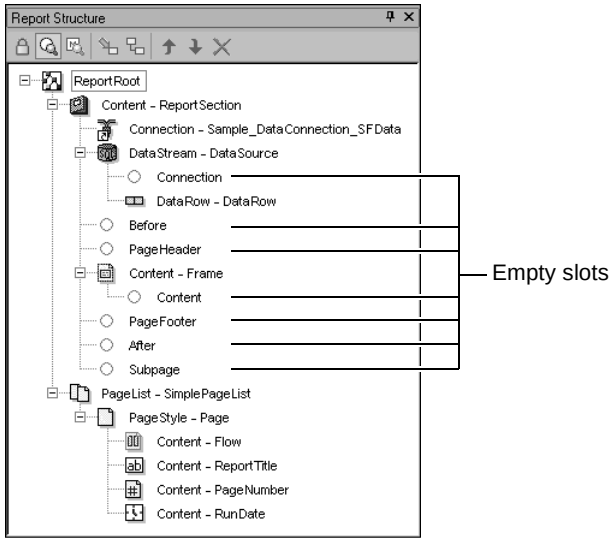


Figure 1-3 Report Structure, showing empty slots

In this manual, to show the report components in Report Structure in a more compact format, many illustrations hide the empty slots. Hiding empty slots shows only the slots that contain report components, as shown in Figure 1-4.

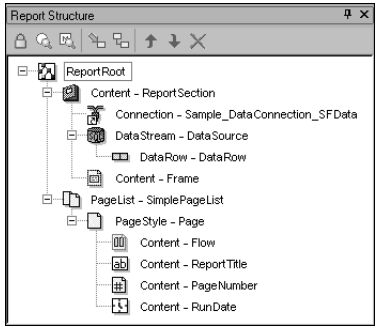


Figure 1-4 Report Structure, hiding empty slots

Additional windows in which you can show or hide empty slots include Layout, Libraries, and Scratch Pad.

How to show or hide empty slots in the design perspective

Right-click an empty area in a window to display the Show Empty Slots option, then:

- Select Show Empty Slots to display empty slots.
- Deselect Show Empty Slots to hide empty slots, as shown in Figure 1-5.

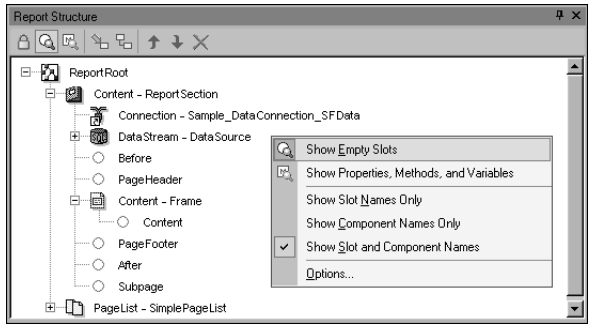


Figure 1-5 Deselecting Show Empty Slots option

How to set the default design display

1 In e.Report Designer Professional, choose Tools➛Options. Options—Design Editor appears as shown in Figure 1-6.

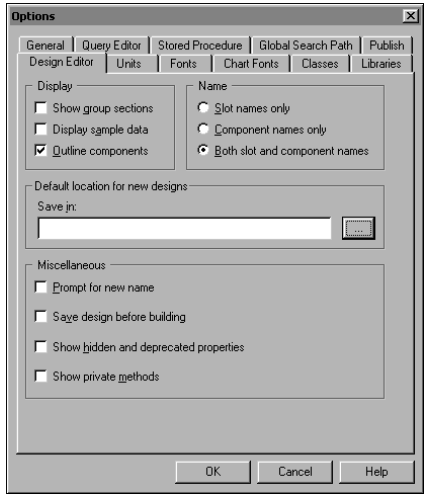


Figure 1-6 Options—Design Editor

2 Select the following items:

- Outline components
- Both slot and component names
- Save design before building

3 In Save in, type a path to a directory in which to save reports. For example, type:

```
C:\Program Files\Actuate11\My Reports
```

The My Reports directory installs with e.Report Designer Professional. You can use this directory as your default report directory or you can specify another directory. Options—Design Editor looks like Figure 1-7.

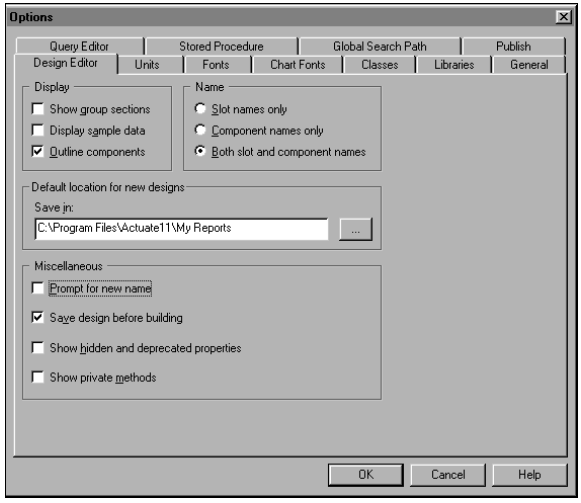


Figure 1-7 Display and Save options

Using the preceding settings, both slot and component names appear in the report design, such as Content—DetailFrame. You also save new report design files to a default directory.

4 In Options, choose Units. Options—Units appears as shown in Figure 1-8.

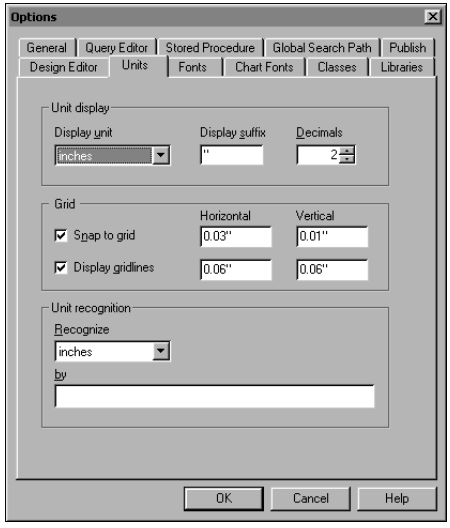


Figure 1-8 Options—Units

The Unit display section shows the default units of measurement for your locale. The Grid section indicates whether report components should align to a grid and whether to show grid lines in the report design environment.

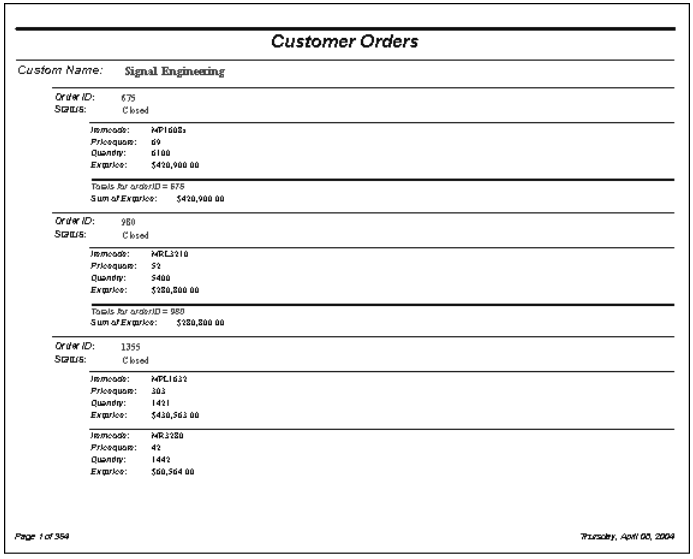
5 Click Display gridlines to turn off grid lines in the report design environment.

6 Choose OK to return to the design environment.

Creating a report that groups data

In this section, you learn how to create a listing report using the listing report wizard. A listing report, also known as a grouped report, can use data from multiple database tables. Using the wizard, you select the data, organize it into groups, and sort the data in each group.

The report that you create in this section shows customer orders and the totals for each order, each customer, and all customers. The report groups data by customer name, then by order ID. The first page of the report looks like the one in Figure 1-9.


Customer Orders

Custom Name: Signal Engineering

Order ID: 675
Status: Closed

Itemcode: MP1682x
Pricequan: 69
Quantity: 6100
Extprice: $420,900.00

Totals for orderID = 675
Sum of Extprice: $420,900.00

Order ID: 980
Status: Closed

Itemcode: MRL1210
Pricequan: 52
Quantity: 5400
Extprice: $280,800.00

Totals for orderID = 980
Sum of Extprice: $280,800.00

Order ID: 1355
Status: Closed

Itemcode: MPL1612
Pricequan: 303
Quantity: 1421
Extprice: $430,563.00

Itemcode: MR1220
Pricequan: 42
Quantity: 1442
Extprice: $60,564.00

Page 1 of 354

Thursday, April 08, 2004

Figure 1-9 First page of a report

Opening the listing report wizard

In e.Report Designer Professional, choose File➛New. Create New Report appears, as shown in Figure 1-10.

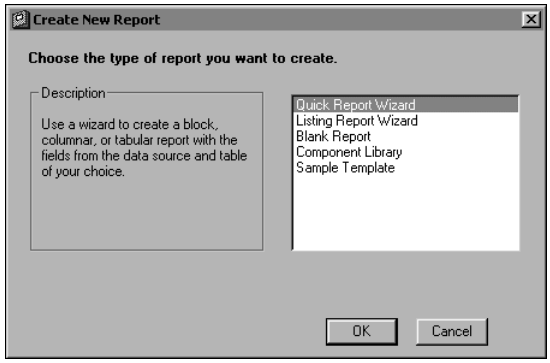


Figure 1-10 Create new report dialog box

Select Listing Report Wizard. Choose OK. Report Wizard—Data Sources appears.

Choosing a data source for a listing report

You can create a report from an existing data source, such as a data stream or flat file. You can also select an existing database connection or create a new database connection. These options are shown in the Data Sources dialog box, in Figure 1-11.

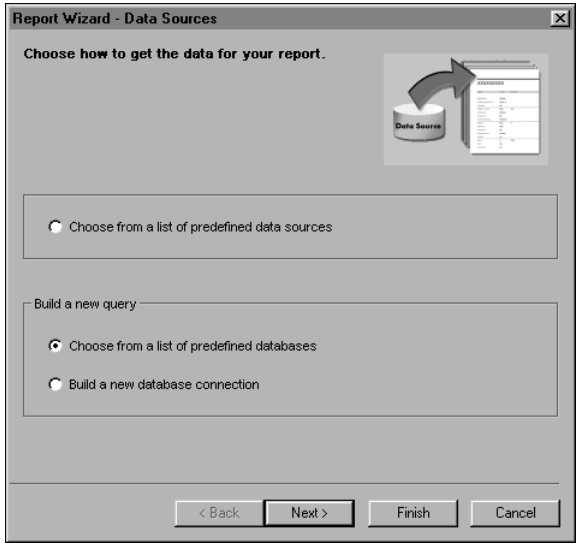


Figure 1-11 Data sources options

Report Wizard—Data Sources presents three options:

- Choose from a list of predefined data sources

This option supports choosing from the data source components available to your report design.

- Choose from a list of predefined databases

  This option supports choosing a data source that you configure for e.Report Designer Professional.

- Build a new database connection

  This option supports choosing a type of connection, such as DB2, ODBC, or Oracle, and customizing it by providing a password, user name, or other information.

How to select a data source for a listing report

1 In Report Wizard—Data Sources, select Choose from a list of predefined databases. Choose Next. Report Wizard—Choose Database displays a list of database connections available to the report, as shown in Figure 1-12.

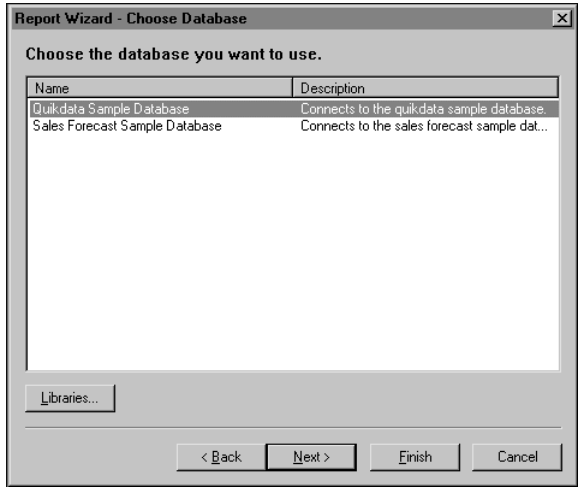


Figure 1-12 Database connections

2 Select Quikdata Sample Database. Choose Next. Database Login appears, as shown in Figure 1-13.

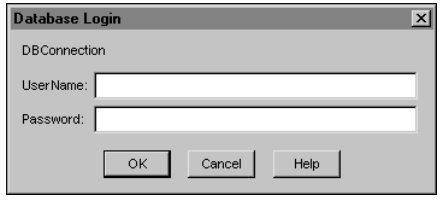


Figure 1-13 Database login

You do not need a user name or password to access Quikdata.

3 Choose OK. Report Wizard—Select Fields appears.

Selecting tables and columns for a listing report

After logging in to the database, you write the query that specifies the data to retrieve. Start by selecting the tables that contain the data for the report. Then, select database columns to appear in the report.

For this report, you use columns from the customers, orders, and items tables. The order in which you select the columns determines their display order in the report.

How to select tables and columns for a report

1 In Report Wizard—Select Fields:

1 In Table or View, select the customers table.

2 For the customers table, in Available fields, select customName. Choose Add. CustomName moves to Selected fields.

3 In Table or View, select the orders table.

4 For the orders table, in Available fields, add orderID and status.

5 In Table or View, select the items table.

6 For the items table, in Available fields, add itemcode, pricequote, quantity, and extprice.

Report Wizard—Select Fields looks like the one in Figure 1-14.

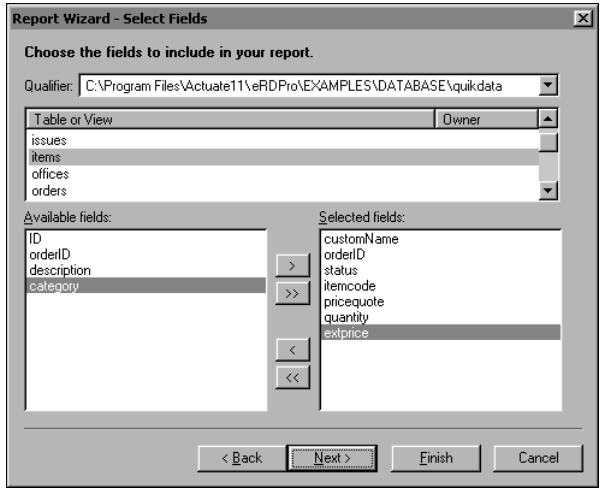


Figure 1-14 Select Fields

2 Choose Next. Report Wizard—Specify Grouping appears, as shown in Figure 1-15. The pane at the left lists the tables that contain data for this report. The pane at the right contains the fields for each table.

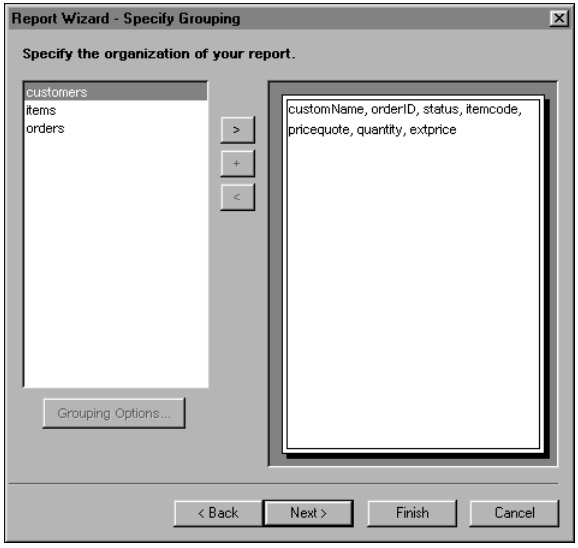


Figure 1-15 Specify Grouping

Next, you organize this data into groups that determine the structure of your report.

Organizing the data by table

You determine a report's structure by grouping data. A group is a set of data rows that have one or more field values in common. In this procedure, you create groups for the customers and orders tables.

How to organize data by table

1 In Report Wizard—Specify Grouping, in the left pane, select customers. Choose Add. The customName field that you selected from the customers table appears in a separate group in the pane at the right.

2 Select orders. Choose Add. The fields from orders appear in a separate group beneath the customers group. Specify Grouping looks like the example in Figure 1-16.

In the report, the customers group contains the orders group. In other words, the report displays orders by customer name.

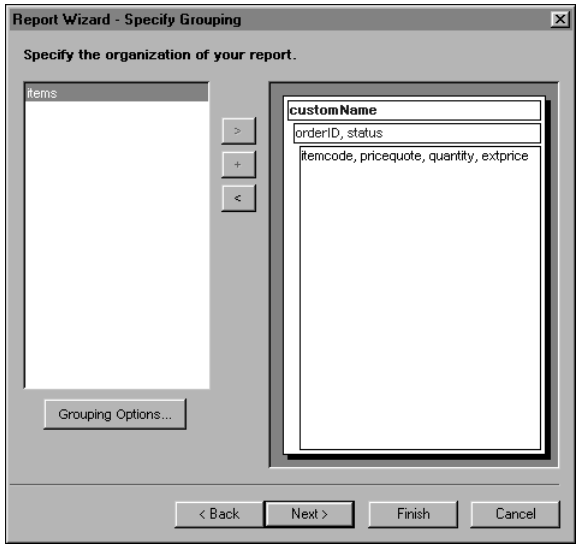


Figure 1-16 Customer name contains the order group

3 Choose Grouping Options. Grouping Options displays the group keys for the groups that you created. The default sort order for items in a group is ascending, as shown in Figure 1-17.

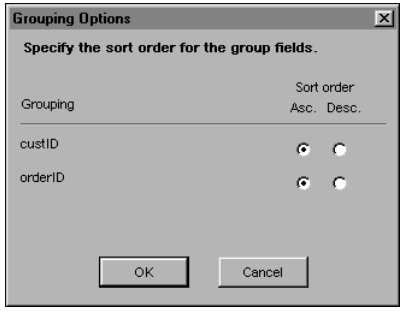


Figure 1-17 Sort order options

Choose OK.

4 Choose Next. A second Report Wizard—Specify Grouping page appears, as shown in Figure 1-18. The pane at the left shows the fields in the report. The pane at the right shows the groups in which the fields appear.

This grouping specifies that the report organizes customer data by customer name and includes a separate orders group.

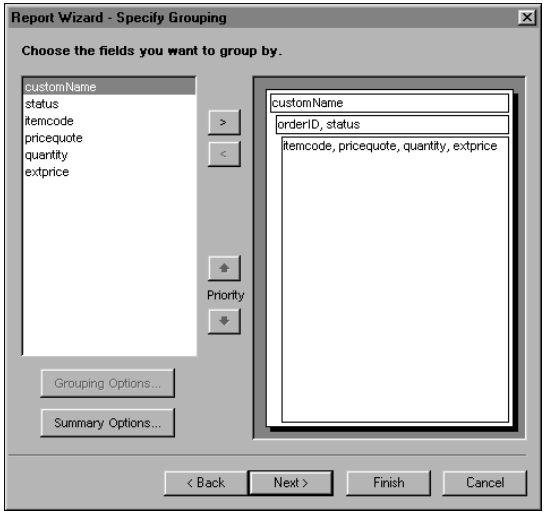


Figure 1-18 Specify Grouping

Adding summary data

Use Summary Options to create a subtotal of the extended prices for each group. Choosing Summary Options produces a window, Report Totals, on which you can work with numeric data types. You can add data, take an average of a list of numbers, and display the minimum or maximum values for a field. Report Totals lists all numeric data types, regardless of whether you can perform calculations using the value in the field.

How to summarize data

1. In Report Wizard—Specify Grouping, choose Summary Options. Report Totals appears, as shown in Figure 1-19.
2. For extprice, select Sum. The extended price is equal to (pricequote * quantity). Sum(extprice) produces a total for each group and a grand total at the end of the report. Choose OK.

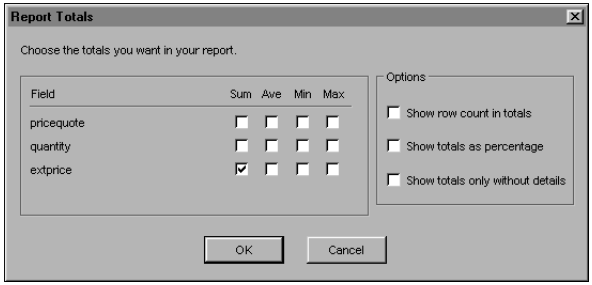


Figure 1-19 Report Totals options

3 In Report Wizard—Specify Grouping, choose Next. Report Wizard—Data Sorting appears.

Sorting the data

Detail rows contain the data. In the report you create in this chapter, the detail rows contain information about each order. To sort detail rows, you select the field or fields to sort by. In this report, you sort by itemcode.

How to sort data

1 In Report Wizard—Data Sorting, select the first drop-down list. Report Wizard—Data Sorting lists the detail data fields.

2 Select itemcode, as shown in Figure 1-20. In Sort Order, accept the default setting, Asc.

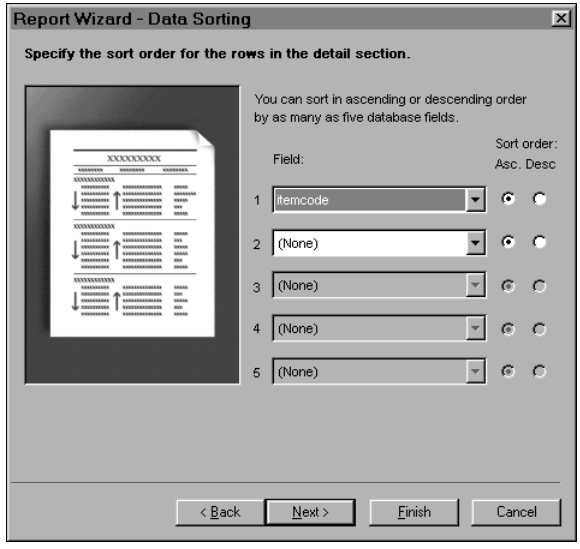


Figure 1-20 Data sorting options

Choose Next. Report Wizard—Choose Layout Style appears.

Selecting a layout style for a listing report

In this section, you select a layout style, paper size, and page orientation for the report. When you choose a layout style, a graphical representation of that style appears in the pane at the left.

How to select a layout style

1 In Report Wizard—Choose Layout Style, specify layout options:

1 In Available Layout Styles, select Tabular—Stepped.

2 In Paper Size, accept the default size of 8 1/2 by 11 Letter.

3 In Orientation, select Landscape.

Report Wizard—Choose Layout Style looks like the one in Figure 1-21.

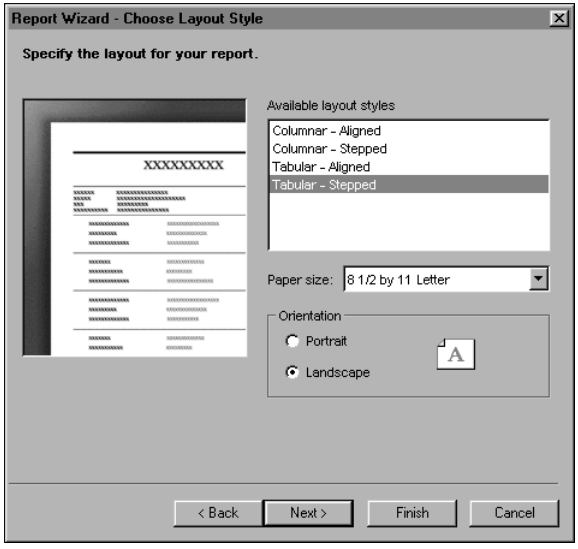


Figure 1-21 Layout style options

2 Choose Next. Report Wizard—Finish appears.

Saving the report design

When you save the report design, you provide a title for the report. Then, you can either run the report immediately or view the report design to make further modifications.

How to save a report design

1 In Report Wizard—Finish, type the following report title:

```
Customer Orders
```

This title appears once at the beginning of the report.

2 Select Change the report design, as shown in Figure 1-22.

Choose Finish. The design environment opens. The report design appears in the layout window, as shown in Figure 1-23.

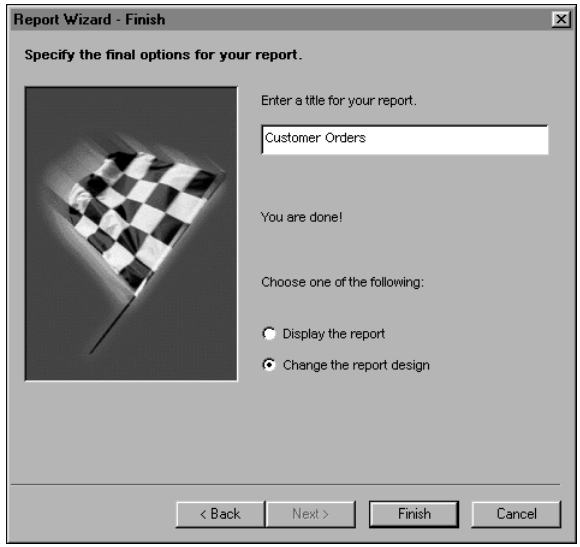


Figure 1-22 Final report options

3 To view more report components, expand Content—ReportSection by clicking the plus sign beside the component.

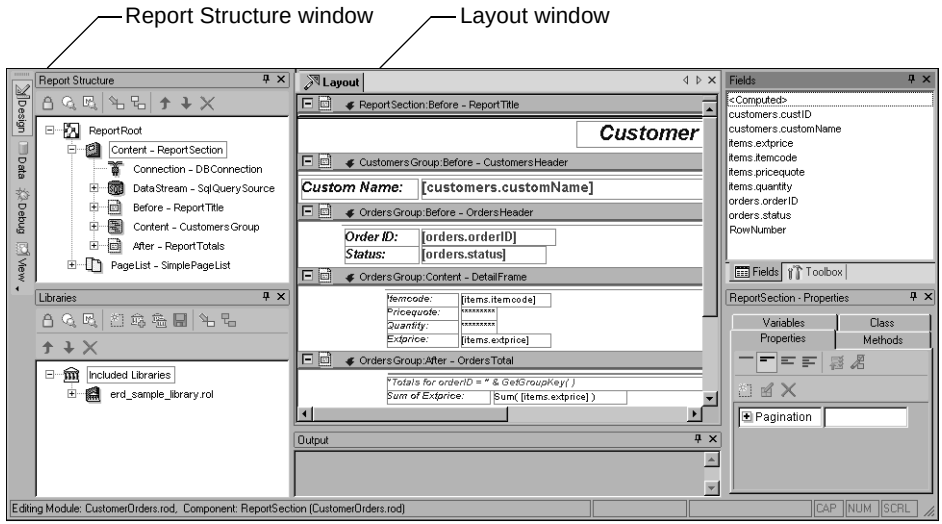


Figure 1-23 Report design environment

The left portion of the design environment shows the structure of the report and any libraries to which the report refers. The center portion is the layout window, where you can view and work with report components.

4 To see more of the report structure, minimize the Libraries window by clicking the pushpin in the Libraries title bar as shown in Figure 1-24.

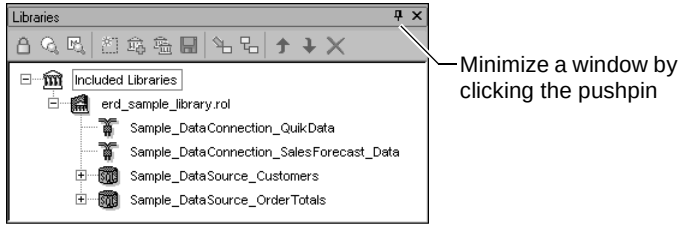


Figure 1-24 Pushpin for minimizing window size

A Libraries tab appears at the upper left of the design environment as indicated in Figure 1-25. Report Structure expands.

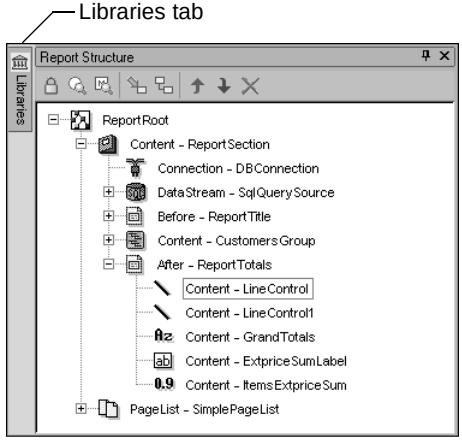


Figure 1-25 Libraries tab for expanding window size

You can then expand additional components to see more of the report structure.

5 Repeat step 4 to minimize:

- Fields
- Properties
- Output

Minimizing these windows lets you see more of the report layout.

6 Choose File➛Save As. Save As appears.

7 In File name, type:

```
CustomerOrders.rod
```

Choose Save.

Viewing the query

In the data perspective, you can view your query and modify it if necessary before running the report. You also can use this perspective to preview the data. In later sections, you learn how to use the data perspective to create report parameters and perform other tasks.

How to view the query and preview the data

1. Choose View➛Data.
2. In Database Login, choose OK. You do not need a user name or password to access Quikdata. The data perspective for this report opens.

   Database Browser appears at the left of the data perspective. Database Browser displays the path to the database that you use in this report. It also lists the tables in the database.
3. Expand the database to view the tables, by choosing the plus sign next to the database name, as shown in Figure 1-26.

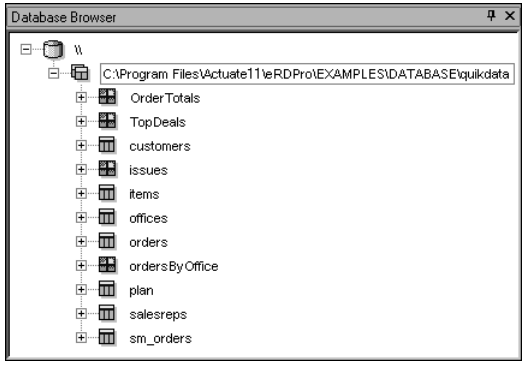


Figure 1-26 Database Browser

You can also expand an individual table to view its columns.

The window at the right shows the tables that you chose for this report. CustomerOrders—Columns displays the database fields that you chose in the order in which you chose them.

You can use the data perspective, as shown in Figure 1-27, to modify your choices.

e.Report Designer Professional creates joins between tables based on common fields. In this example, the customers and orders tables have custID in common. Orders and items have orderID in common.

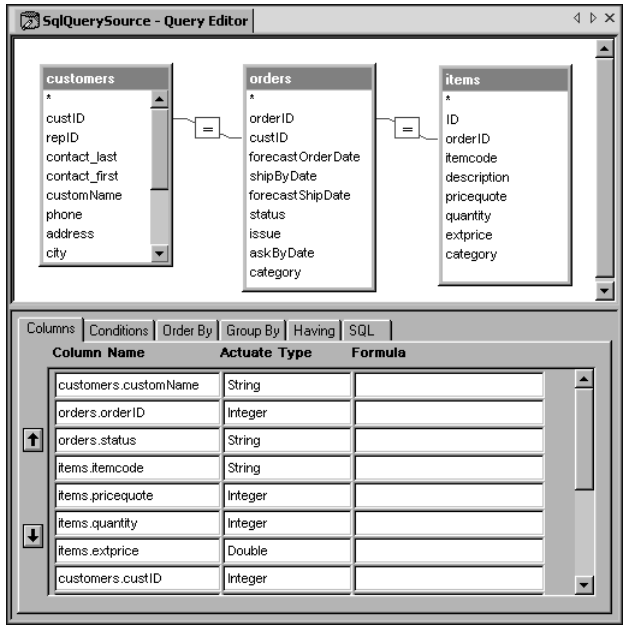


Figure 1-27 Query editor data perspective

4 Choose SQL to view the SQL query as shown in Figure 1-28.

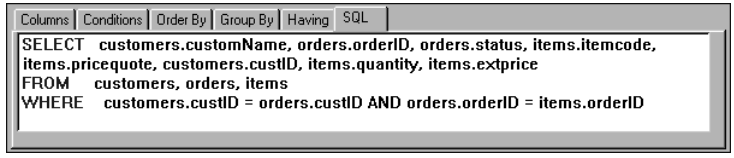


Figure 1-28 SQL Query data

As this example shows, Actuate e.Report Designer Professional creates the query for you as you work in the report wizard and the data perspective. It is not necessary to understand SQL to use Query Editor.

In this example, the SELECT statement lists the columns that you chose using the report wizard. The FROM statement lists the tables that you chose. The WHERE statement reflects the joins that appear in the upper pane of the design perspective.

5 To preview the data the SQL query retrieves from the Quikdata database, choose SQL➤Preview Data. Preview Query Data displays the data, as shown in Figure 1-29.

Choose Close.

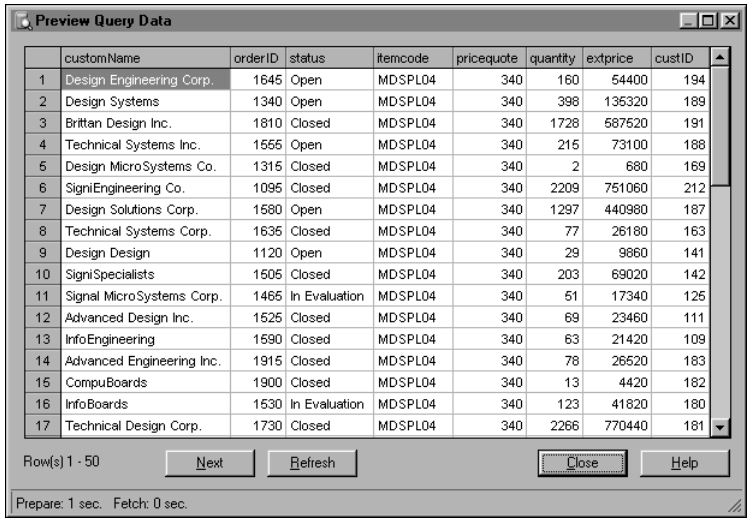


Figure 1-29 Preview Query Data

6 Choose View➛Design to return to the design perspective.

Understanding report structure

When you open the report, the design perspective is open and Report Structure appears at the left. If you do not see Report Structure, choose View➛Report Structure.

How to view the report structure

1 In Report Structure, expand Content—ReportSection if you have not already done so.

2 Expand the following components:

- Content—CustomersGroup
- Content—OrdersGroup
- Content—DetailFrame

Report Structure looks like the structured list shown in Figure 1-30.

The group sections in Report Structure correspond to the groups that you selected using the report wizard. Content—CustomersGroup contains the report content, which is organized into two groups, one for customer data, the other for order data. Content—OrdersGroup is a nested group, meaning that it appears within the other group section, Content—CustomersGroup.

Content—DetailFrame contains controls that display the details of each order. There are data controls that display prices, quantities, and extended prices.

There are labels for each data control. There is also a control for a line that separates one order from another.

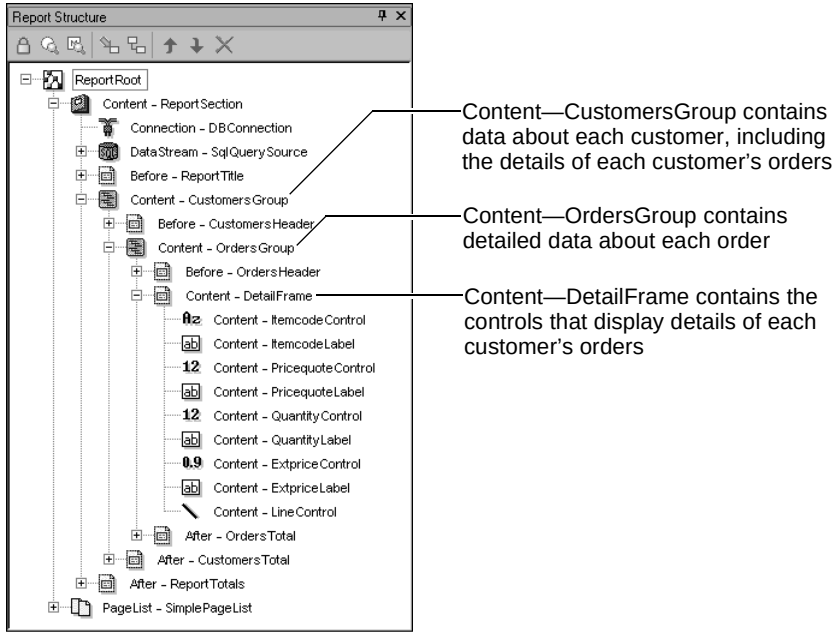


Figure 1-30 Report structure

3 Expand Before—OrdersHeader, as shown in Figure 1-31.

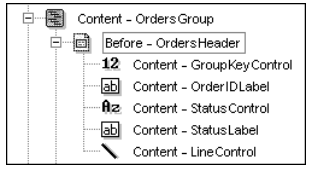


Figure 1-31 Expanded Before—OrdersHeader

The Before frame for a group section typically contains data, labels, and graphic images that appear before each group. In this case, the frame contains:

- Content—GroupKeyControl, a data control that displays the value for the group key
- Content—OrderIDLabel, a label that identifies the order ID
- Content—StatusControl, a data control that displays the value for the order status
- Content—StatusLabel, a label for the status data
- Content—LineControl, a line that separates one order from another

Enhancing a report design

The report design that you create in "Creating a report that groups data," earlier in this chapter, displays data that uses default naming and formatting. For example, the report title appears centered over the content. There are lines above and below the title. Components have default names. You can change any of the default formatting.

In this section, you:

- Rename a report component.
  e.Report Designer Professional assigns default names to a report component based on a number of factors, including the position of the component in the report and the type of control. You can change these default names to make the content or purpose of the component clearer to other users.
- Resize a control.
  Resizing a control ensures that the control's data value appears in the report in its entirety.
- Format a numeric data control to display currency values.
  You can format a numeric control to display currency, date and time, percentage, and other numeric values.

How to rename a control

The name that you assign to a control is a class name that must adhere to Actuate naming conventions. Component names are case sensitive. You cannot use spaces or special characters, such as a backslash (\), period (.), or question mark (?), in a component name.

1 In Before—OrdersHeader, right-click Content—GroupKeyControl. A context menu appears, as shown in Figure 1-32.

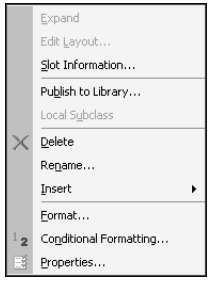


Figure 1-32 Context menu

2 Choose Rename.

3 In Rename, as shown in Figure 1-33, type:

```
OrderIDControl
```

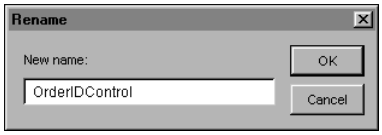


Figure 1-33 Renaming a report component

Choose OK. In Report Structure, the control name changes to Content—OrderIDControl.

How to resize a control

e.Report Designer Professional assigns a default size to a control. You can change the length, width, or height of a control to ensure that its entire contents appear in the report.

1 In the design perspective, select the control beside the Order ID label by clicking anywhere within the control's border. Handles appear on the control's border, as shown in Figure 1-34.

Figure 1-34 Resizing a control

2 Select the handle at the lower right of the control and drag it to the right until you can read the control label, as shown in Figure 1-35.

Figure 1-35 Widening a control label

How to format a currency control

When you set the format of a currency control, e.Report Designer Professional displays data using the currency conventions of the default locale of the client machine.

1 If the Properties window is minimized, expand it by clicking the Properties tab. Then, choose the push pin to lock the page in place.

2 In Report Structure, in Content—DetailFrame, select Content—ExtpriceControl. ExtpriceControl—Properties appears.

3 Click the field beside Format. Choose Currency from the list that appears, as shown in Figure 1-36. When the report runs, extended price data is formatted as currency.

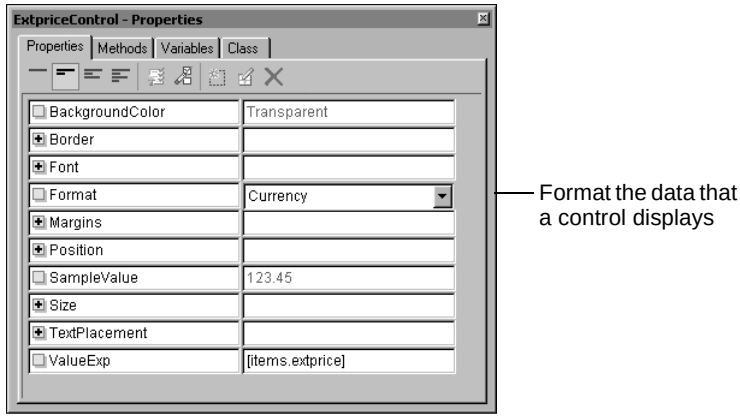


Figure 1-36 Formatting data from the Properties menu

4. Repeat steps 2 and 3 to format Content—ItemsExtpriceSum. This control appears three times in Report Structure, in the following frames:
   - After—OrdersTotal
   - After—CustomersTotal
   - After—ReportTotals
5. Minimize ItemsExtpriceSum—Properties.
6. Choose File➤Save to save the report design.

Running a report

After you finish the design and save it, run the report to compile a report object executable (.rox) file and generate a report object instance (.roi) file.

How to run the listing report

1. In e.Report Designer Professional, choose Report➤Build and Run.

   Status messages appear in the status line of the application window as e.Report Designer Professional compiles the report. As soon as the first page generates, it appears in the view perspective. e.Report Designer Professional continues generating pages until the report completes.

   The report includes the title that you typed in the report wizard and a default footer. The footer appears on every page. It includes the page number, page count, and the date. The report structure reflects the data grouping that you chose in the report wizard.

   The first page looks like the one in Figure 1-37.

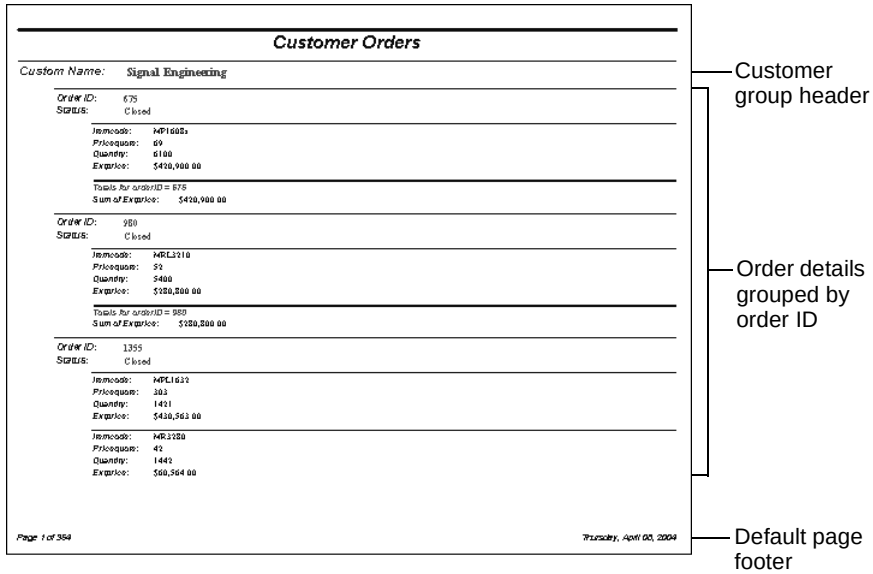


Figure 1-37 First page of a report

2 Navigate through the report to view additional data.

3 Choose View➛Last Page to view the end of the report. Note that the group totals and report total appear in Currency format, as shown in Figure 1-38.

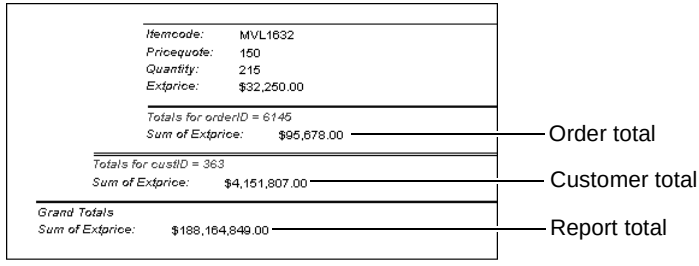


Figure 1-38 Last page of a report

At this point, you can return to the design perspective to make additional changes to the report. For example, you can create ad hoc parameters to permit users to restrict data when they run the report. You can add images to the report or make other enhancements to the report appearance. You can also set page breaks between groups so that each customer or order begins on a new page.

When you complete your work with the report, you can distribute it to give others access to the data.

Chapter

2

Understanding report design

This chapter contains the following topics:

- Overview of the design process
- Understanding report design concepts
- About the report design environment
- Performing common operations
- Going beyond the graphical tools

Overview of the design process

Designing a report involves the following tasks. You do not have to perform all the tasks in the order in which they are presented here. If you are new to e.Report Designer Professional or learning how to design reports, use this task list as a starting point.

- Plan the report.
- Start a new report design.
- Access data.
- Lay out the report.
- Format the contents of the report.
- Customize the page layout.
- Preview and test the report.

Planning the report

Before you create a report, you should identify the information that you want the report to provide and decide how to present the information. It is important to think through those details, then design a prototype on paper, which you can use to get feedback from your report users. Planning saves time in the long run because you do not waste time creating a polished report that contains the wrong information or layout.

Starting a new report design

Start e.Report Designer Professional and use one of the following options to create a new report design:

- Use a report creation wizard.
- Start with a report template.
- Start with a blank report.

Accessing data

A report can access data from a wide range of sources, including databases, enterprise resource planning (ERP) applications, and text files. To set up the report to access data, take the following steps:

- Install and configure the necessary third-party software on your computer. To access a database such as Oracle, for example, you must first install the database client software. To access a data source using Open Database Connectivity (ODBC), you must first configure it as an ODBC data source.

- In e.Report Designer Professional, choose the data source and connect to it.
- Build a query.

Data sources can store vast amounts of data. For most data sources, you extract the data that you need by writing a query. For text files and internal data sources, you can extract data without writing a query. For detailed information about accessing data from different types of data sources, see *Accessing Data using e.Report Designer Professional*.

Laying out the report

There are many ways to present information in a report. A report can display information in a simple list, a table, a chart, in multiple subreports, or any layout that you can imagine. Laying out a report entails placing data on the page and organizing it in a way that helps the report user easily grasp and analyze the information.

Formatting the report content

After you place data in a report, you format the report to give it a professional and polished appearance. Typical formatting tasks include highlighting certain data, specifying the presentation of dates, numbers, or currency values, aligning text, resizing components, and hiding sections. You can also apply conditional formatting to data, for example, to display numbers in different colors depending on their values. You can even use conditional formatting on entire sections, for example, to display different content depending on what report users choose to see when they run the report.

Customizing the page layout

When you create a new report, e.Report Designer Professional uses default values for page size, orientation, and margins. You do not have to set up the page layout unless you want a different layout. You can create highly formatted pages that use one layout for a title page and another for subsequent pages, or you can create different page layouts for left and right pages. The page layout is also where you add report elements that you want to appear in the same location on every page, such as a page number, a date, or a copyright statement.

Previewing and testing the report

You should preview and test the report as you design it. The most important thing to test is your query. After you build the query and insert fields onto your report, run the report and view the results. Verify that the data that appears is what you expect. Do not worry about the precise placement of the data or format at this stage. Once you validate the data, you can proceed with laying out the fields precisely and formatting them. As you do these tasks, check the report output periodically.

Understanding report design concepts

This section describes the basic report design concepts that e.Report Designer Professional uses. If you are using e.Report Designer Professional for the first time, understanding these concepts can help you quickly learn how to build reports.

About components

The building blocks of a report design are called components. Each piece of data, text label, graphical element, image, or chart in a report design is a component. e.Report Designer Professional provides a rich set of components, including:

- Components for displaying data, such as text, integer, date, and chart controls
- Drawing components, such as lines and rectangles
- Data access components, such as database connections and data sources
- Page layout components, such as pages and flows
- Structure components, such as frames, group sections, and conditional sections, that you use to organize other components

When you build a report without using a report creation wizard, you build it by placing components, one at a time, in a report design file. For example, to add a title to your report, place a label control to display the title. To add a company logo, place an image control to display the graphic.

Figure 2-1 and Figure 2-2 show components as they appear in the Layout window and as they appear when you view the finished report document.

Part of a report design

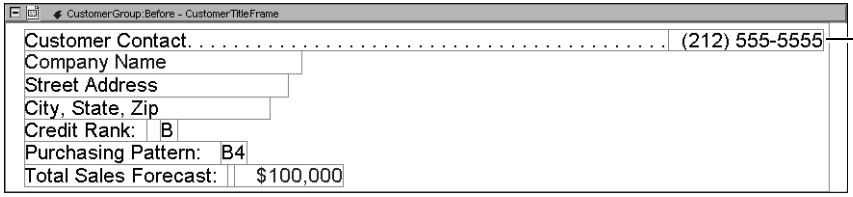


Figure 2-1 Report components in the Layout window

The report design you create is a template of sorts. It specifies the structure and instructions for generating the finished report but it is not the report itself. When you place a component in the report design to display data from a field, that single component can represent tens, hundreds, even thousands of values, depending on the number of records in the database.

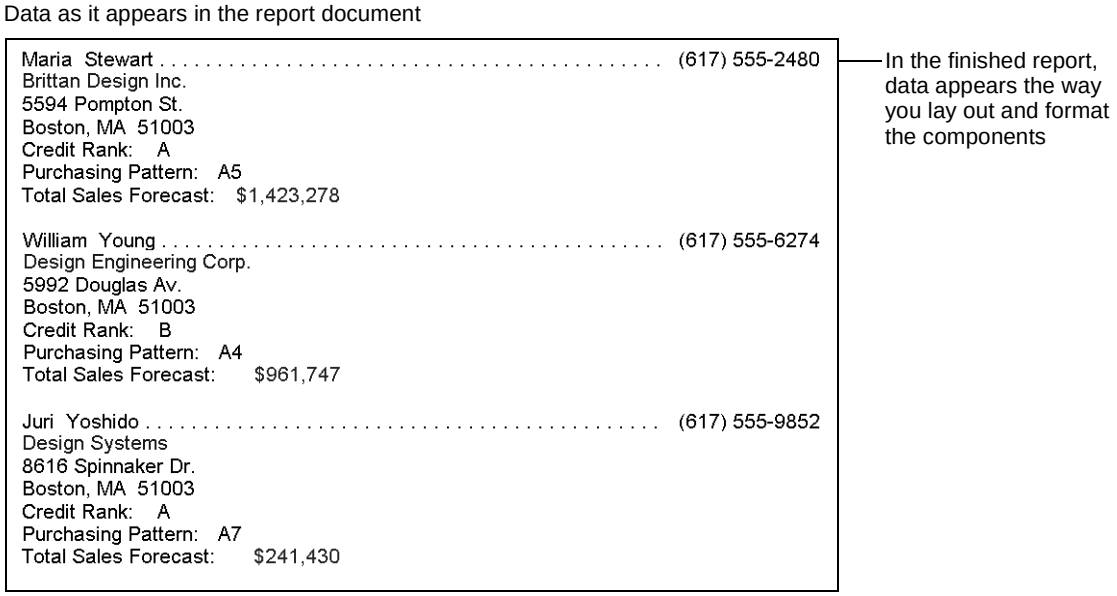


Figure 2-2 Report components in a finished report document

About component relationships

Building a report involves assembling components in a variety of ways, the most common of which is to place components within other components. Like most other documents, reports often present information in a hierarchical structure. Just as this manual organizes information into main topics and subtopics, a report can organize data into sections and subsections. For example, a sales report can contain these sections and subsections in this order:

- At the top-most level, a section for each regional office
- Within each office section, sales representatives at each office
- Within each sales representative section, all the customer accounts
- Within each customer account, all orders

Figure 2-3 shows the report design for this sales report in the Report Structure window. Notice the hierarchical structure. The main report component contains several components, including the office section. The office section, in turn, contains the sales representative section, which contains the customer section, which contains the orders section. To create this type of hierarchical structure, you use the group section component, a type of structure component.

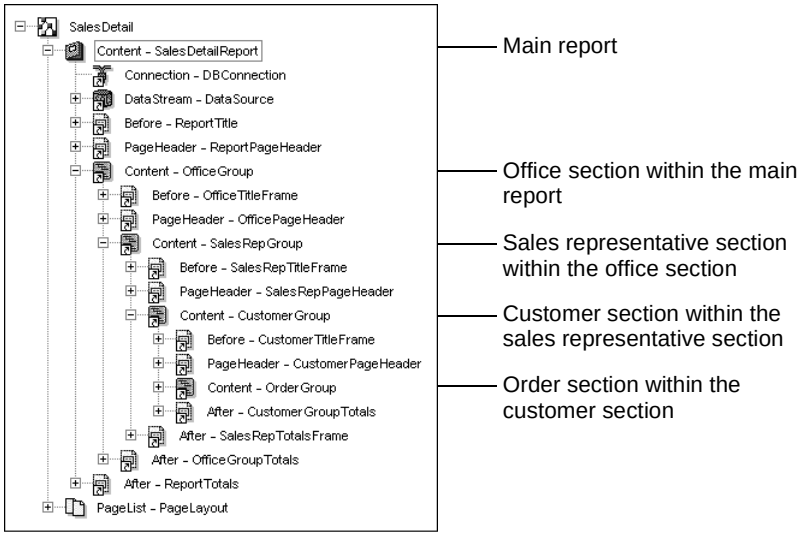


Figure 2-3 Report Structure window for a report design

About reusable components

Companies that create a large number of reports typically simplify the report design process by publishing common components to a library that all report developers in the company can share. The benefits are obvious:

- Standardization of report elements
- Ease of access to items that are frequently used
- Rapid and efficient development

The types of components that report developers typically publish to a library include the following:

- Page styles
- Report banners with company name and logo
- Connections to databases

When you create a report design, you can include a library and use its components in your report.

About the report design environment

e.Report Designer Professional provides a flexible environment that meets the needs of both beginning report developers and experienced report developers

who want the power of programming. It provides wizards to help report developers get a jump-start when they create reports, customizable options that enable report developers to tailor the environment to their style of working, and user-friendly tools for designing, debugging, and previewing reports, and publishing them on the web.

This section introduces the report design environment. If you are using e.Report Designer Professional for the first time, reviewing the topics in this section can help you learn how to use e.Report Designer Professional more effectively.

When you first start e.Report Designer Professional, a Welcome dialog asks if you want to create a new report or open an existing report.

If you choose to create a new report, e.Report Designer Professional presents a list of options, as shown in Figure 2-4. These options represent the wizards, libraries, and templates that are available for report creation.

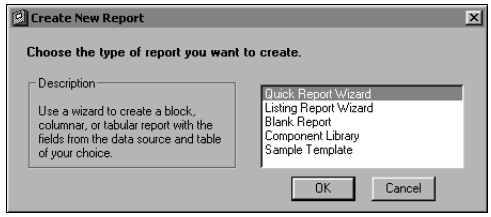


Figure 2-4 New report options list

The next time that you launch e.Report Designer Professional, the application starts with the last report that you saved, as shown in Figure 2-5. If you or another user customized the window, it might not look exactly like Figure 2-5.

Figure 2-5 Default design perspective windows

Windows application users will notice the standard user interface elements, such as pull-down menus, toolbars, and status bar. Integrated Development Environment (IDE) users will also notice the similarity in look and feel between e.Report Designer Professional and developer applications, such as Visual Studio.

About the application window elements

The application window consists of four main parts with which you should be familiar:

- Menus
- Toolbars
- Work area
- Status bar

Menus

The menus are standard Windows menus that you can pull down with your mouse. In some cases, you can also use shortcut key combinations to choose menu options. For example, Ctrl+O opens a new report.

Toolbars

When you first start e.Report Designer Professional, all available toolbars appear below the menu bar. The toolbars provide quick access to many of e.Report Designer Professional's functions. Many of the icons on the toolbars are self-explanatory. If, however, you need information about a toolbar button, hover the mouse over an icon. A ToolTip describing the toolbar button's function appears.

Work area

The work area occupies most of the application window. It is the large area between the toolbars and status bar. Here, you work with the tools that help you build, test, and preview reports. The set of windows that appear in the work area change, depending on the current report design task.

Status bar

The status bar appears at the bottom of the application window. It displays information about the actions that you perform as you build your reports. It also shows more detailed information about a menu option or toolbar button. You can hide the status bar by deselecting the Status Bar option on the View menu.

Managing workflow

e.Report Designer Professional provides an intuitive way to create a report. It provides tools following these primary phases of creating a report:

- Laying out the report
- Accessing data
- Debugging the report
- Viewing the report

When you work with e.Report Designer Professional, you work in one of these phases at a time. Each report creation phase is called a perspective. For each perspective, e.Report Designer Professional displays different menus and toolbars and a different set of windows, or tools, in the work area.

Perspectives model the way report developers work and they manage the work space effectively. When creating a report, you often perform each task multiple times and in different order. You might write a query, lay out data, preview the report, then modify the query, preview the report, debug the report, change the layout, preview the report again, and so on, until you are satisfied with the report's contents and appearance.

Because the process of creating a report is an iterative rather than a linear process, the use of perspectives makes it easy for you to switch from one main task to another and have only the relevant tools at your fingertips. To switch from one perspective to another, use one of the techniques shown in Figure 2-6 or Figure 2-7, or use the keyboard shortcuts outlined below:

- The perspective toolbar

Figure 2-6 Perspective toolbar

- The Design, Data, Debug, or View option on the View menu

Figure 2-7 View menu

- Keyboard shortcuts:
  - Alt+1 for the design perspective

- Alt+2 for the data perspective
- Alt+3 for the debug perspective
- Alt+4 for the view perspective

About the design perspective

Typically, you use the design perspective more than the other perspectives. Here, you lay out the report and format its contents. The design perspective appears when you start e.Report Designer Professional. By default, six windows appear in the work area, in the arrangement shown in Figure 2-8. You can minimize, close, or rearrange the windows.

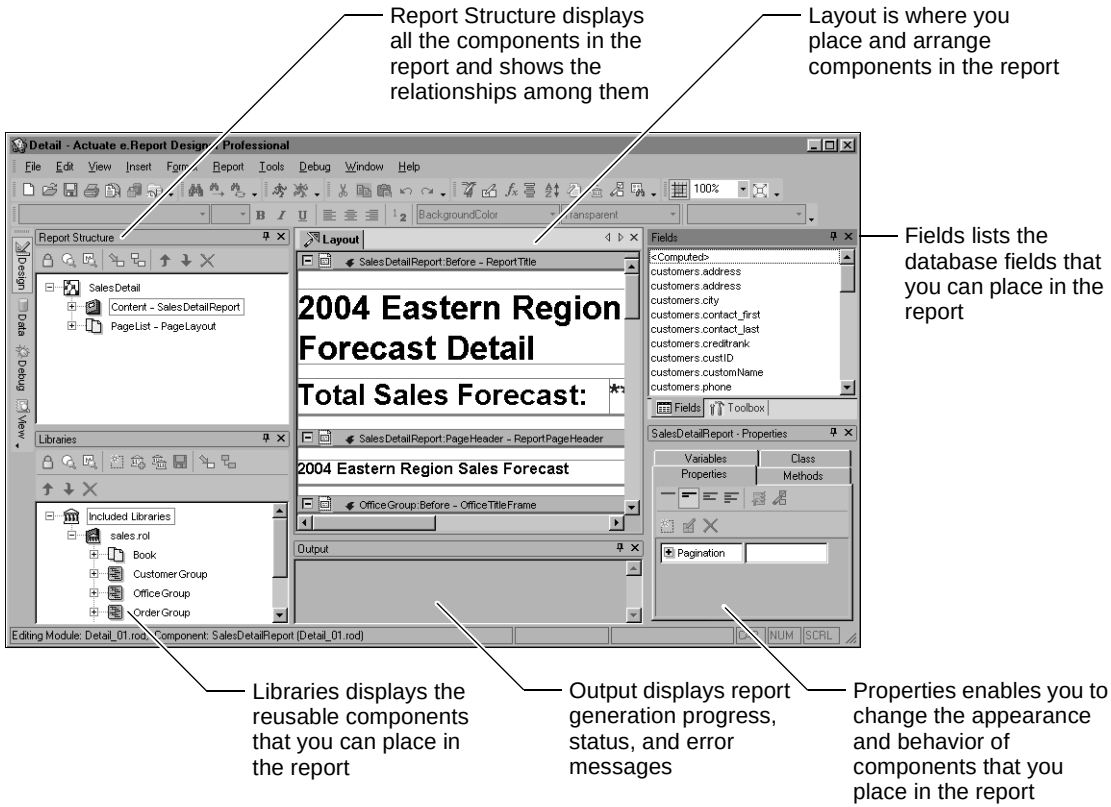


Figure 2-8 Default design perspective windows

About the data perspective

The data perspective is where you specify the data that you want the report to access. Figure 2-9 shows the default view of the data perspective. You can minimize, close, and rearrange windows.

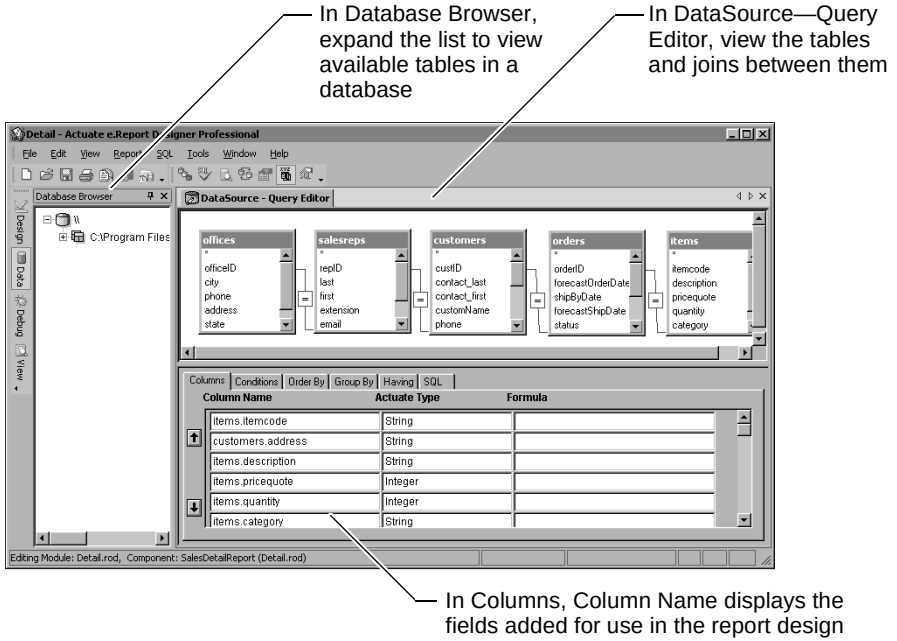


Figure 2-9 Default data perspective windows

About the debug perspective

The debug perspective is where you use e.Report Designer Professional's debugging tools to find and fix compile-time and run-time errors in the report. Typically, you use these tools if you need to debug code that you write to customize a report, or if you want to observe the code that e.Report Designer Professional executes to generate a report.

Figure 2-10 shows a view of the debug perspective. The debug perspective displays content only after you build and run a report.

About the view perspective

The view perspective is where you preview a report as you build it. Use this perspective to verify that a report appears and behaves the way that you intended. Figure 2-11 shows the default view of the view perspective. You can open additional windows, such as the Table of Contents and Search.

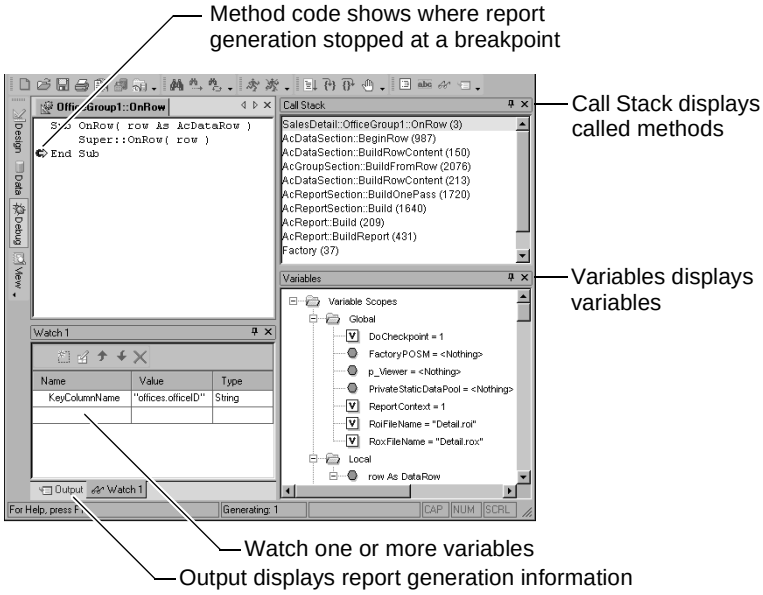


Figure 2-10 Default layout of debug perspective windows

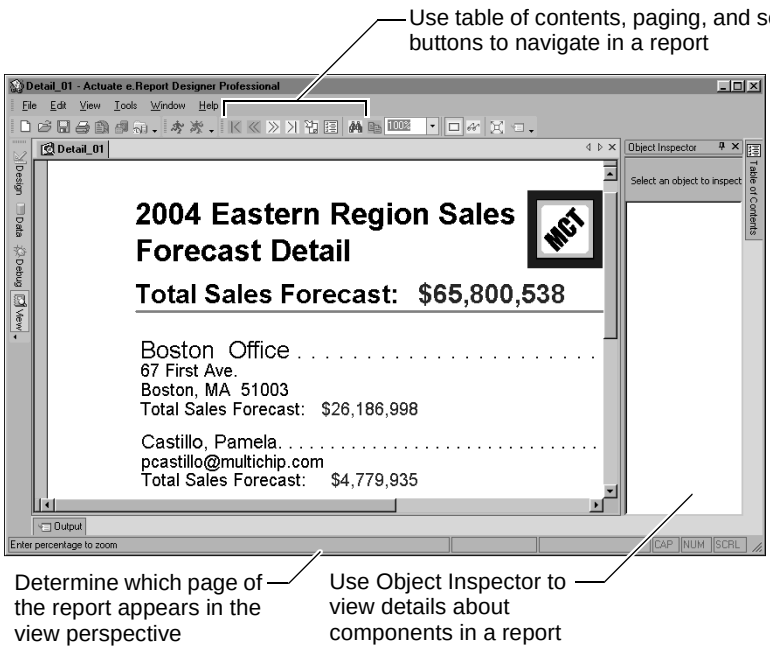


Figure 2-11 Default view perspective window

Personalizing the application window

The application window displays different menus, toolbars, and windows for each of the four perspectives. e.Report Designer Professional provides the capability to tailor each perspective individually to better fit the way that you work.

For each perspective, you can change or rearrange the set of windows that appear in the work area. You can also hide, rearrange, or create toolbars to meet your needs. For example, if you work mostly with Report Structure and Layout in the design perspective, you can close or hide the other windows when you are not using them.

e.Report Designer Professional saves the current window layout when you exit the program. The next time that you start e.Report Designer Professional, the perspective appears the way it did the previous time. You can, at any time, restore a perspective to its default state.

Moving, minimizing, and stacking windows

You can manipulate windows in the work area using any of the standard methods, including opening, closing, and resizing them. In addition, you can:

- Move windows from their default positions and place them anywhere that you choose, inside or outside the application window. To move a window, click its title bar and drag it to the desired location. A navigation pattern appears as you move the window. Drop the window at the top, bottom, left, or right corner of the navigation pattern to place the window in that position relative to the other window.

- Minimize, or hide, a window by choosing the pushpin icon.
- Drop the window on the center of the navigation pattern to stack one window on top of another so that the two windows appear as a single window with tabbed pages. To unstack windows, click a tab and drag the window to another location.

Figure 2-12 shows how to rearrange the positions of two windows.

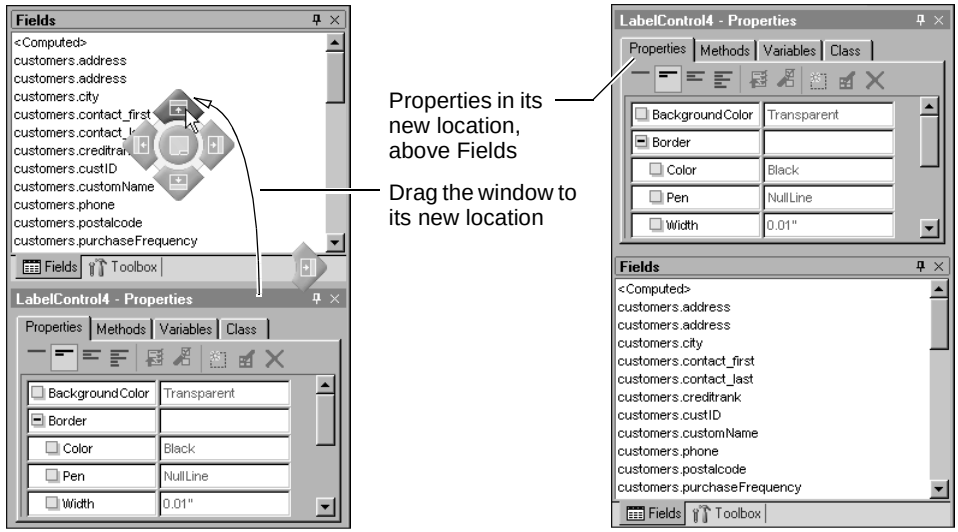


Figure 2-12 Rearranging window positions

Figure 2-13 shows how to minimize a window so that only the title bar appears at the edge of the work area.

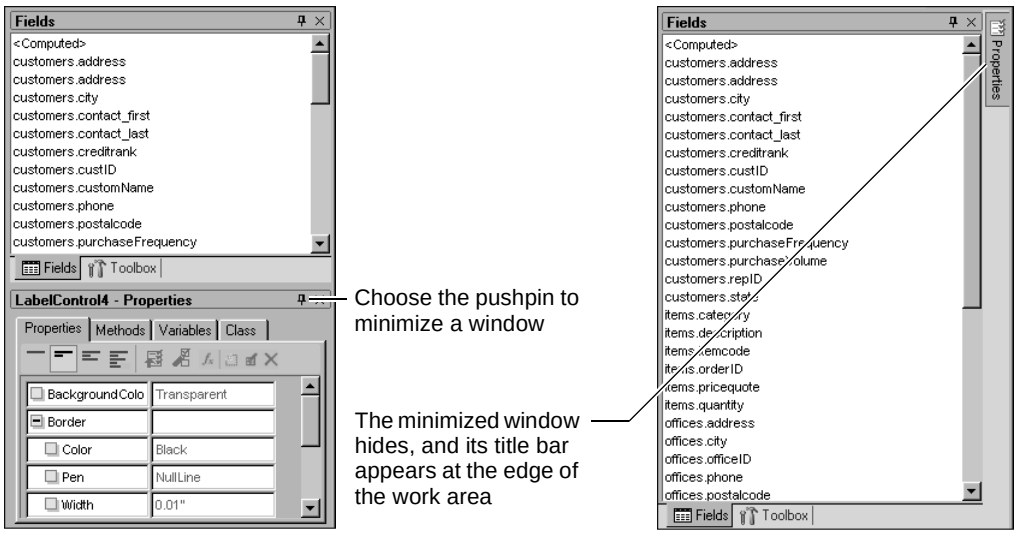


Figure 2-13 Minimizing windows

Figure 2-14 shows how to stack two windows so they appear as one window with tabbed pages.

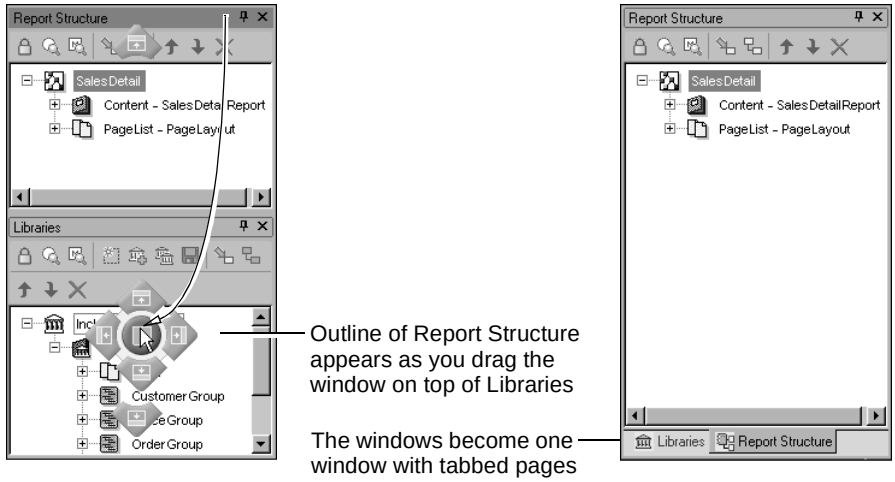


Figure 2-14 Stacking windows

Closing and restoring windows in a perspective

In each perspective, you can arrange the windows that display the tools that you require. If these windows cover parts of a design or view that you want to review, you can quickly close all those windows, then restore them to continue using the tools. In the design perspective, for example, you typically display Report Structure, Libraries, Properties, and Toolbox, which cover much of the layout window. To close the windows that cover the layout window, choose Close All Panes. To restore the windows, choose Restore All Panes.

Moving, hiding, and changing toolbars

You can change how and where toolbars appear:

- Display or hide individual toolbars using View➛Toolbars.
- Display or hide individual buttons on each toolbar by choosing the arrow on the right side of the toolbar and choosing Add or Remove Buttons.
- Move a toolbar from its default position by dragging the gray bar on the left side of the toolbar and placing it elsewhere, as indicated in Figure 2-15. If you move a toolbar away from the top of the screen, it becomes its own window.

Figure 2-15 Modifying toolbars

Restoring a perspective to its default layout

If you rearranged the windows in a perspective and do not like the way that you positioned them, you can rearrange the windows further or reset them to their default positions.

To restore the current perspective to its default layout, choose Tools➤Reset Workspace➤Active Perspective Only. To restore all the perspectives, choose All Perspectives.

The windows in the perspective appear in their original positions. Restoring a perspective has no effect on changes that you made to toolbars, only to windows.

About the main design tools

When you build a report, you spend most of your time in the design perspective and you use the layout and report structure windows more than the other windows. These windows, described in the following sections, provide two different views of a report design.

About the Layout window

The Layout window is where you do most of the initial work when building a report. It provides a visual way for you to place and format components in a report. Unlike Report Structure, Layout shows only components that are visible in the finished report. It does not show, for example, data connection or data source components.

The Layout window uses gray bars to label the different areas of a report design in which you can place report components. You place data or components in these different areas, called slots, to determine where and how often the data or component appears in the final report. For example, to create a report page header that displays the report title at the top of every page, you place a label component in a slot called PageHeader.

Figure 2-16 shows a report design in Layout.

About the Report Structure window

The Report Structure window displays the components that you use in a report design and shows the relationship among the components. It shows the hierarchy of components in a format similar to the outline view of a Word or PowerPoint document.

Unlike the Layout window, which shows only the visible components that appear in the final report, Report Structure shows all components. These components include data access components, such as connections and data sources, that do not appear in the final report.

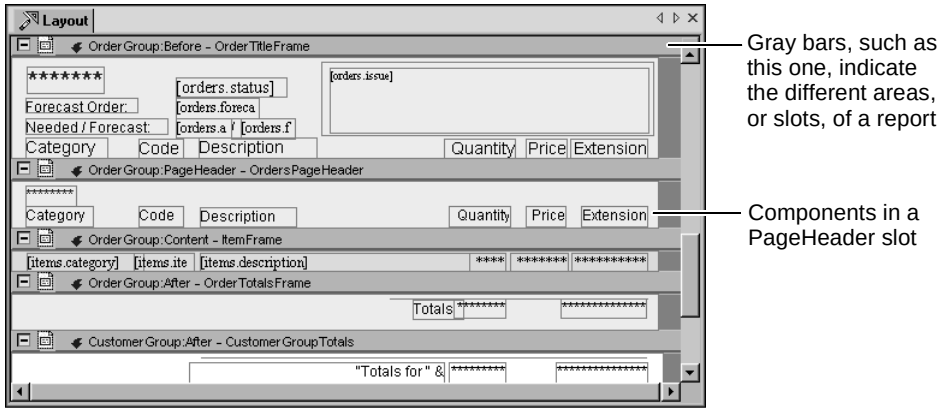


Figure 2-16 Report design in the Layout window

Use Report Structure to work with the organization of the report design. You can also place components in the report structure, but you have no control over where e.Report Designer Professional places them initially in the report layout. In fact, if you place multiple controls in a frame in Report Structure, the controls are stacked on top of each other in the Layout window.

Figure 2-17 shows a report design in Report Structure. Each icon represents a component. The top-level component represents the entire report and it appears in every report design. Components display either their default names or names that you specify.

As with the outline view of a Word document, you can expand or collapse parts of the report by clicking the plus (+) or minus (-) signs.

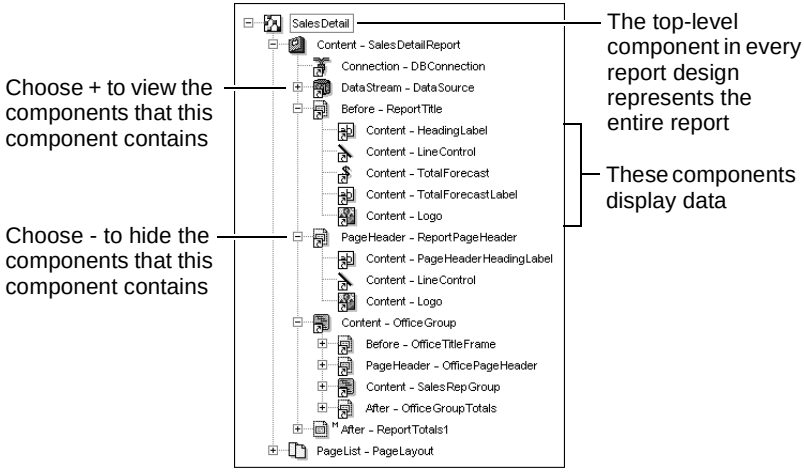


Figure 2-17 Report structure

Performing common operations

As you build a report, you perform certain operations frequently. These operations include:

- Placing components in a report design
- Naming a report component
- Searching for a component
- Examining class hierarchies
- Setting a component's properties
- Copying components
- Deleting components
- Saving, running, and previewing a report
- Closing a report

Placing components in a report design

You can place a component in a report design in one of the following ways:

- Use the Insert menu to select a component, then place it in the desired location.
- Drag a component from one of the following windows and drop it in the desired location:
  - Toolbox
    Toolbox contains every type of component that you can use in a report.
  - Fields
    Fields lists all the database fields that you can place in the report. The query you build determines the fields that are available. When you place a field in the report, e.Report Designer Professional creates the appropriate component for the field value. For example, if you place a field of integer data type, e.Report Designer Professional creates an integer control to display the field value.
  - Libraries
    Libraries contain components that you or other report developers publish for re-use in multiple reports.

Using any of these techniques, you can place components in either Report Structure or Layout. Place non-visible components, such as structure or data

source components, in Report Structure. Place visible components in the layout to control the location and appearance of the components.

The result of placing components using the Insert menu, Toolbox, or Fields is the same. In each case, e.Report Designer Professional creates a new component in the report design. When you place a component from a library, however, e.Report Designer Professional creates a reference to the library component rather than creating a new component. When you create a reference, the library component that you use in your report is automatically updated whenever the original author updates it. If, for example, corporate marketing decides to refresh the colors of the company logo, the image is changed in one place—the library—and all report designs that use the image are automatically updated.

In Report Structure, a reference to a library component displays a shortcut icon, as shown in Figure 2-18. You can hover the cursor over the shortcut to display a ToolTip with details about the reference. Also, you can double-click the icon to highlight the control in Layout, display its values in Properties, and display its class inheritance in Class Hierarchy. If you modify a referenced component, e.Report Designer Professional changes the component to a local copy and the consequences are that it remains only local.

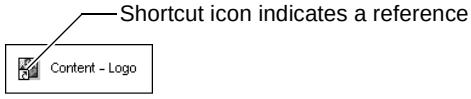


Figure 2-18 Shortcut to a reference

Naming a report component

When you place a component, you have the option of giving it a unique name. To use this option, you must set up e.Report Designer Professional to prompt you for the new name. Then, when you place a component using the Insert menu, the toolbox, or the Fields list, e.Report Designer Professional presents the Class Name dialog, as shown in Figure 2-19.

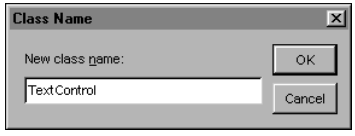


Figure 2-19 Class Name dialog box

This dialog contains a default name for the component that you are creating. If you accept this name, the name appears in Report Structure. If you change this name, be aware that e.Report Designer Professional imposes naming restrictions. Component names are case-sensitive. You cannot use spaces or punctuation in component names, nor can you begin a name with a numeral.

How to set up a name prompt

1 In e.Report Designer Professional, choose Tools➛Options. Options—Design Editor appears as shown in Figure 2-20.

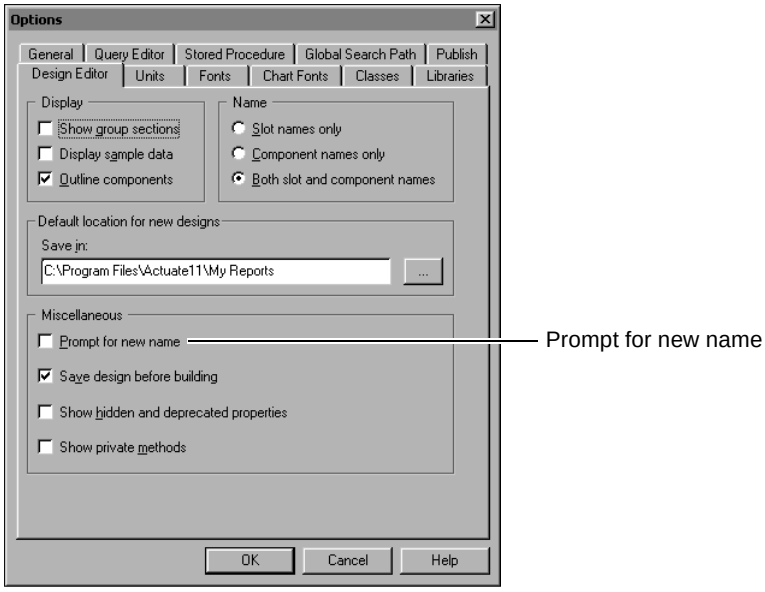


Figure 2-20 Name prompt setup in Options—Design Editor

2 Select Prompt for new name. Choose OK.

Searching for a component

e.Report Designer Professional supports searching for classes, properties, or methods within an open project. You perform simple operations on the results, or use them as a navigation point to look further into the report design.

Search criteria are specified in Component Search. To launch this tool, choose Tools➛Component Search. Component Search appears as shown in Figure 2-21.

The conditions combine using the AND operator. For example, if you specify both a class name and a method name, the search finds all classes or methods that match both conditions. Table 2-1 describes the fields in Component Search.

Table 2-1 Component Search fields

Group	Item	Description
Search in	Entire Project	The search encompasses the main report design as well as all of the included libraries.

Table 2-1 Component Search fields

Group	Item	Description
Search in *(continued)*	Report Design	Searches only the main report design. Included libraries are not searched.
	Included Libraries List	Use this choice to narrow the search to a particular library.
Classes	Name	Searches for this class name. Can include regular expressions.
	Include Subclasses	Includes all subclasses that match the name condition. This choice does not combine with other search conditions using the AND operator.
Properties	Property	Searches for this property name. This field must match the property name exactly. For a structure property the structure name and the member name must both be specified. The search is for a class with a property of this name, with the value specified in the Value field.
	Value	The value to match in the specified property. This value can be an Exact match, a Partial string, or a Regular expression.
Methods	Method	Searches for this method name. To match, a class must have at least one method that meets the conditions specified.
	Text	Searches for this text in the source code of each method, including the signature.
List	Classes	Display matching classes.
	Methods	Display matching methods.
Match	Exact match	Requires an exact match of search conditions.
	Partial string	Matches on a partial string.
	Regular expressions	Matches based on regular expressions.
	Match case	Matches on a case sensitive basis.

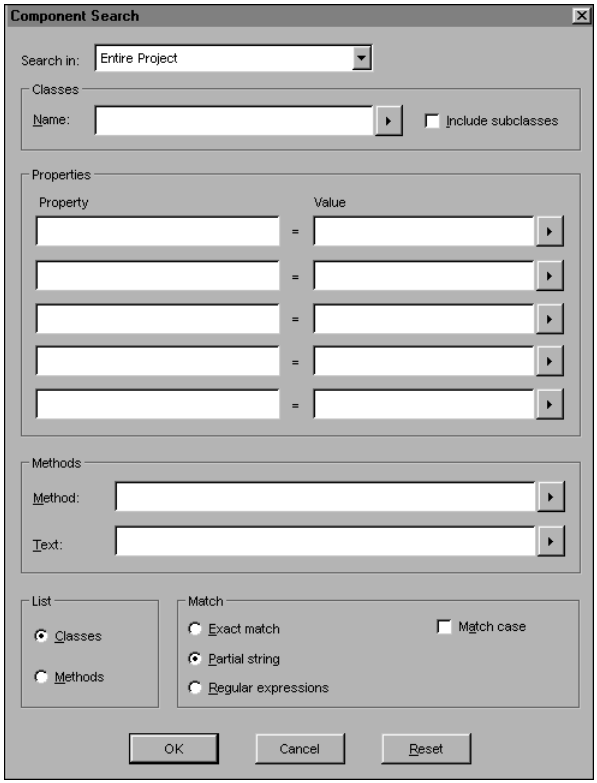


Figure 2-21 Component Search

Understanding a component search regular expression

A regular expression is a tool that supports matching a string to a pattern. These expressions are very powerful and can become complex. Regular expressions are made up of normal characters and metacharacters. Normal characters include upper and lower case letters and digits. Metacharacters have special meanings to the regular expression interpreter. The metacharacters are shown in Table 2-2.

Table 2-2 Regular expression metacharacters

Metacharacter	Description
+	Matches one or more occurrences.
*	Matches zero or more occurrences.
?	Matches zero or one occurrence only.
.	Matches any single character except newline.

Table 2-2 Regular expression metacharacters

Metacharacter	Description
[ ]	Matches a single character within the set specified. If the first character in the set is a caret (^), the match is of any character not in the set.
^	When used with brackets ([ ]), matches any character except those specified.
\	Escape character that supports matching on a metacharacter and other special characters.

Any character that is not a metacharacter matches itself. If you need to match on a metacharacter, a backslash (\) followed by the metacharacter causes the search to interpret the metacharacter as a normal character. Escape codes can also be used for certain special characters, as described in Table 2-3.

Table 2-3 Escape codes

Escape Code	Value
\b	Backspace
\e	ESC (escape)
\f	Formfeed
\n	Newline
\r	Carriage return
\t	Tab
\xddd	Hex number 0xddd
\ddd	Octal number ddd
\^C	Control code

A regular expression can consist of one or more characters. You can combine regular expressions to create more specific search patterns. See Table 2-4 for some typical combinations.

Table 2-4 Regular expression examples

Regular Expression	Description	Example Match
[a-z]*	Zero or more lowercase characters	a, aaa, asdfaklj, ajdfk
[A-Z][a-z]*	Any capitalized string	A, Abc, Bdksf, Zfjska
.*	Any string of characters	fjdDkls, 29823, aks888

(continues)

Table 2-4 Regular expression examples (continued)

Regular Expression	Description	Example Match
T.*	Any string that Starts with capital T, and ends with zero or more other characters	TabcdEFG, T12345, T
ABC?D	A string that has an optional single character	ABCD, ABD
.*Data.*	The string Data surrounded by any other characters	MyDatabase
Test\[+	The string Test followed by at least one left bracket ([)	Test[
X[abc]Z	The letters X and Z with either a, b, or c between them	XaZ, XbZ, or XcZ
X[abc]*Z	The letters X and Z with zero or more letters a, b, or c between them	XZ, XaZ, XaaaaZ, XababccZ
X[^abc]Z	The letters X and Z with any letter except a, b, or c between them	XAZ, XkZ, X4Z

About the results of a component search

The results of a component search appear in Component Search Results. This list appears in the same location as the Toolbox window by default. You view these results by choosing OK on Component Search or by choosing View➛Component Search Results. The window appears as shown in Figure 2-22.

Selecting an item in the result list shows the item's class information in the Properties window. Selecting multiple items shows the common properties of the selected classes. If methods are listed, selecting one opens the editor for that method. The window has several toolbar buttons, as described in Table 2-5.

Table 2-5 Component Results window toolbar

Item	Description
New Search	Opens Component Search with the current search conditions.
Refresh	Runs the current search again, without opening Component Search.
Show Full Scope	Shows a fully scoped name when checked, and non-scoped when unchecked. For example, ColumnHeadings::LabelControl::Finish() is a fully scoped method name and LabelControl::Finish() is non-scoped name.

Table 2-5 Component Results window toolbar

Item	Description
Go to Definition	Opens the definition of the selected item.
Edit	Opens the editor for the selected class or method.
Delete	Deletes the selected class or method.

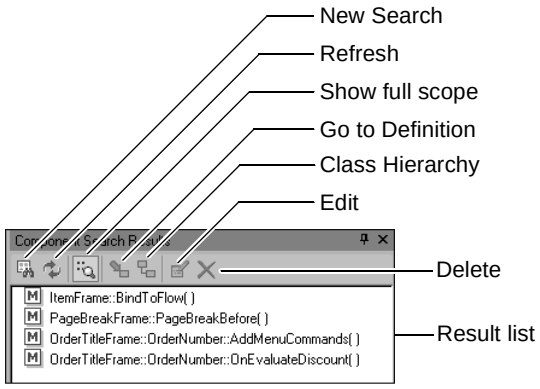


Figure 2-22 Component Search Results showing all overridden methods

How to search for a component

To search for a particular component, enter the information about that component in Component Search. In this example, Component Search and the Component Search Results window are used to search one of the example report designs for classes that match specific conditions.

1. Open <eRDPro_HOME>\Examples\DesignAndLayout\Detail\Detail.rod.
2. Choose Tools➛Component Search. Component Search appears.
3. In Classes—Name, type Forecast and then choose OK. The Component Search Results window appears as in Figure 2-23. Note that there are several components with the word Forecast in their name.

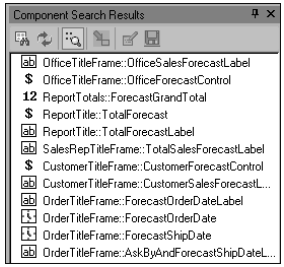


Figure 2-23 Search results

4 Choose View➛Component Search. Component Search appears with the conditions you provided for the previous search.

5 In Classes—Name, change the value to .*Forecast.*. In Properties—Property, type font.color. In Properties—Value type [mM].*. In Match, select Regular expressions. Component Search appears as shown in Figure 2-24. Choose OK. Component Search Results appears with a shorter list. This list contains those components of the report with Forecast in the title and with a Font.Color property with any color starting with M (Magenta, Maroon, MintGreen).

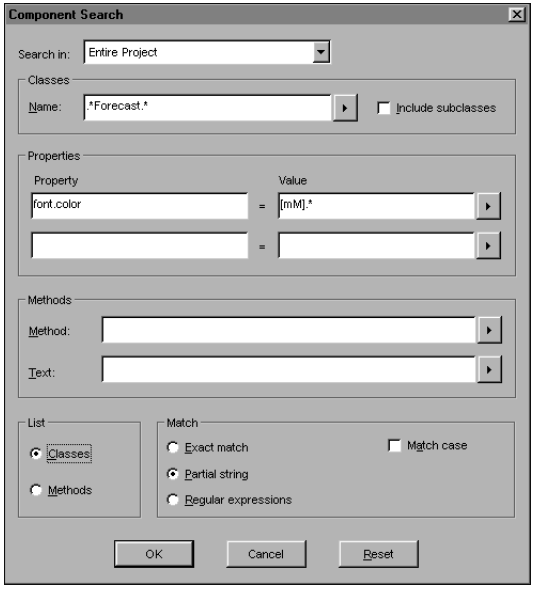


Figure 2-24 Component search with new criteria

6 Choose Tools➛Component Search. Component Search appears. Choose Reset.

7 In Classes—Name type Frame and choose OK. Component Search Results appears similar to Figure 2-25. Note that all frames in this report design are globally scoped.

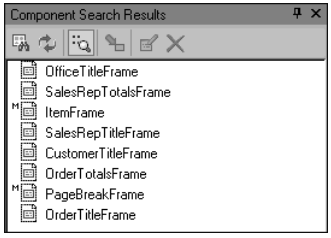


Figure 2-25 Frame search result

8 Choose View➛Component Search. In Methods—Method, type BindToFlow and choose OK. Component Search Results appears with the class that matches all conditions.

9 Choose New Search. In List, select Methods. Choose OK. Component Search Results appears shows the matching method. Select the method. The editor opens for the method.

Examining class hierarchies

To understand the behavior of a component in a complex report design, you typically need to know the customization of a superclass or a subclass of the component. The class hierarchy window provides you the ability to examine the chain of class inheritance that describes a component. This tool shows the list of superclasses that contribute to a given class. It can also show all the subclasses of a given component. It also shows the overridden or defined members for each class.

To open the class hierarchy window, choose View➛Class Hierarchy. The window appears similar to Figure 2-26. To see the hierarchy of any report component, select the component in any window in the design perspective.

The base class from AFC is at the top of the tree and the component is at the bottom. The AFC component that serves as the root in this window represents the closest AFC ancestor to the selected class. If there is no AFC base class, then the top-most ancestor class is displayed. The members of a class appear beneath its node in the same manner as the structure and project windows.

The properties window updates to show any class that you select in the class hierarchy. The contents of the class hierarchy window are updated automatically every time a new class is selected in another window. The Lock View toolbar button disables this behavior, so that you can examine other classes while you keep the current class hierarchy visible.

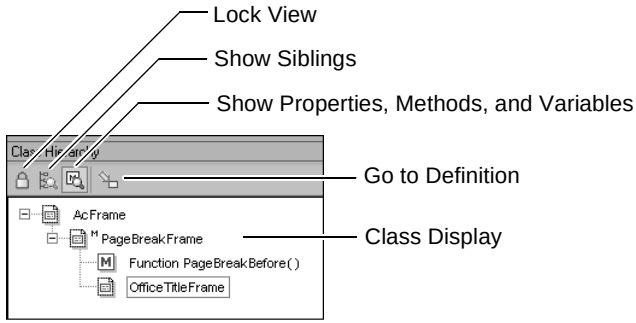


Figure 2-26 Class hierarchy window

Use the Show Siblings toolbar button to show any sibling classes. For example, to see the siblings of a component in the report structure shown in Figure 2-27, select

the component then choose View➛Class Hierarchy. In Class Hierarchy, choose the Show Siblings button. The siblings for the selected component appear in the Class Hierarchy window as shown in Figure 2-28.

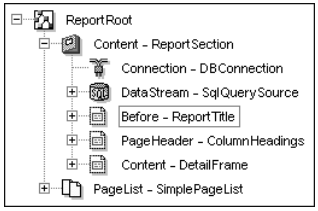


Figure 2-27 Report structure with siblings

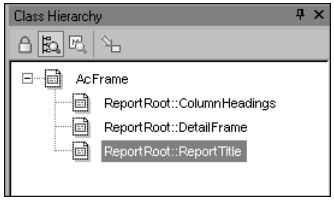


Figure 2-28 Class hierarchy window with siblings displayed

The Show Properties, Methods, and Variables button toggles their display. Choosing Go to Definition selects the component in the appropriate structure or project window.

Setting a component's properties

After you place a component in a report design, you typically set its properties to change its appearance or behavior. For a data control, for example, you can set its display format and style, specify if the report user can search on the data, or control its visibility on different platforms.

To view a component's properties, right-click the component and choose Properties. To set or change a property value, type a value. Some properties, such as background color, provide values that you can select.

The Properties page, shown in Figure 2-29, provides filters that enable you to view the following sets of properties:

- Overridden properties, which are properties for which you specify a value.
- Most commonly used properties. This set includes overridden properties and adds properties that report designers typically set.
- Advanced properties. This set includes commonly used properties from the previous sets and adds properties for more specialized formatting or modifications.

- Expert properties. This set includes properties from the previous sets and adds properties that require programming methods.

When you open the Properties page the first time, the set of most commonly used properties appears. If you select a different filter, that filter is maintained—even after you close e.Report Designer Professional or open a different report—until you select a different filter.

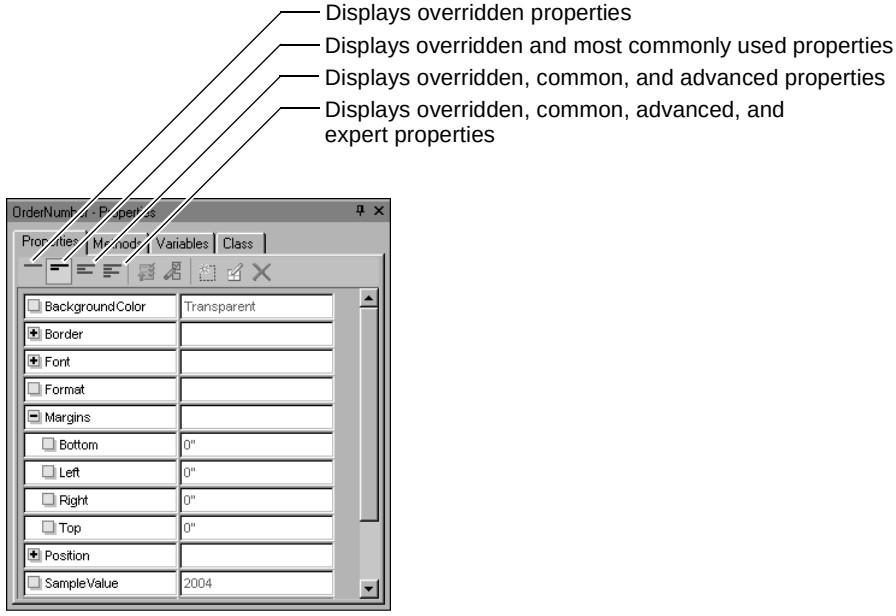


Figure 2-29 Report design component properties

Some properties initially appear only in the advanced properties or all properties set. If you modify the value of one of these properties, it subsequently appears when you choose the overridden properties or most commonly used properties filters, because it becomes an overridden property.

Some properties appear in a property group, for example the Bottom property in the Margins group as shown in Figure 2-29. To indicate the property group and the property name, this property appears as Margins➤Bottom in this manual.

If a deprecated, obsolete, or hidden property has been overridden it appears red in all filter settings.

Copying components

A report design that presents a lot of data can contain many components. Often, you format different pieces of data the same way to achieve a consistent look. For example, you can set column headings to one font style and text data to another. In this situation, you can create and format one component, then copy it and use

it as a template for creating similar components. When you copy a component, all the property settings are also copied.

To copy a component, use one of the following techniques:

- Select the component and press Ctrl as you drag the component to another location.
- Select the component, choose Edit➤Copy, then choose Edit➤Paste. A copy of the component appears on top of the original component.

Deleting components

Some components can have nested components appearing within them in the Report Structure window. If you delete a component containing other nested components, the nested components that are not used elsewhere in the design are also deleted.

For example, if you delete the PageList from a report design, the PageStyle component and its contents are also deleted.

Saving, running, and previewing a report

You can save a report design by using the standard File➤Save command. e.Report Designer Professional saves it in a report object design (.rod) file. To view the results of your report design, you build and run the report. e.Report Designer Professional connects to the data source, retrieves data to populate the report, and generates the report. After e.Report Designer Professional generates the report, it displays the report in the view perspective.

Figure 2-30 shows each step of the internal report-building process and the files that e.Report Designer Professional creates after each action. Understanding the process and the resulting files helps you to decide how to control the report-building process and how to organize and use the different files.

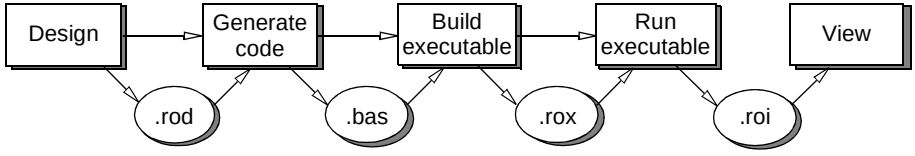


Figure 2-30 Internal report-building process

Each step, except for generating code, corresponds to an option on the Report menu. Most of the time, you choose Build and Run to execute all the steps shown in the illustration, from generate code to view.

Building the report

You can build a report without running it if you want to update the code that e.Report Designer Professional generates but you do not want to view the

finished report. To build a report without running it, choose Report➤Build. e.Report Designer Professional performs the following actions:

- Generates Actuate Basic code and saves it in an Actuate Basic source (.bas) file. The code in this file corresponds to the information in the report object design (.rod) file. You can view a BAS file in any word processor or text editor. Looking at the generated code is a good way to learn Actuate Basic and understand how e.Report Designer Professional creates reports.
- Compiles the BAS file and creates a report object executable (.rox) file. The ROX is the file that you run to create the finished report for viewing or printing.

Running the report

Run the report object executable (.rox) file when you want to generate and view the report. To run a report, choose Report➤Build and Run. e.Report Designer Professional performs the following actions:

- Builds the ROX, as described in "Building the report."
- Retrieves data from specified sources.
- Creates a report object instance (.roi) file. This file contains objects that store the finished report's structure and data.
- Opens the view perspective, which displays the report.

Viewing the report

You view a report in one of the following ways:

- Choose Report➤Build and Run. e.Report Designer Professional executes a series of actions, including retrieving data from specified sources and displaying the report in the view perspective. Use this option to view a report with the most current data, or to preview changes that you make to a report design.
- Choose View➤View. e.Report Designer Professional displays the previously generated version of the report in the view perspective. Use this option if it is not important to view a report with the most current data or if you did not make changes to the report design. This option displays the report more quickly because e.Report Designer Professional does not have to connect to the data source, retrieve data, and generate the report.

Closing a report

You can close a report object design (.rod) file in one of the following ways:

- Choose File➤Close Design.

- In the design perspective, click the X in Layout, as shown in Figure 2-31. If you have several windows open in Layout, you must first select Layout, then click the X.

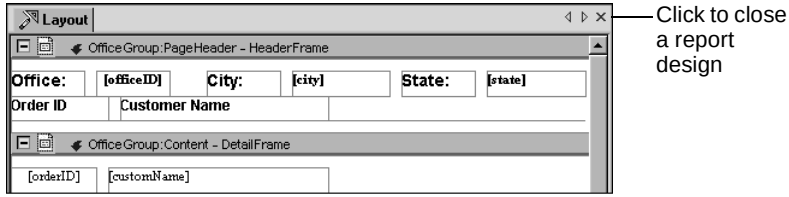


Figure 2-31 Closing a report design

When you close a report design, e.Report Designer Professional closes all the report's windows. When you reopen the file, the design perspective appears and any standard design windows, such as Report Structure, Properties, or Libraries, reopen as well. Windows such as an Actuate Basic source (.bas) file or the method editor do not reopen. If you open a report design in a new version of e.Report Designer Professional and then save it, it is unreadable in older versions. e.Report Designer Professional saves the original report design with a .bak extension.

Going beyond the graphical tools

To create most reports, you use e.Report Designer Professional's graphical tools without needing to understand how they process your actions. As you develop expertise with the product, however, you might want to control the report-generation process or build highly customized reports that use your own components.

This section describes the information you need to know to go beyond e.Report Designer Professional's graphical tools.

About object-oriented design

Using e.Report Designer Professional's interactive tools, you create reports by combining and manipulating a variety of components. Behind the scenes, e.Report Designer Professional translates your actions into an object-oriented program written in Actuate Basic.

If you are familiar with object-oriented languages or applications, you already know about classes, properties, methods, and the concepts of subclassing and inheritance. e.Report Designer Professional applies all these concepts to report components.

The components that you work with in the graphical user interface are classes. When you place a component in a report, you create a subclass that inherits the properties and behavior of an Actuate class. To change the appearance of a

component, such as its color or font, you set its properties. To customize a component's behavior, you add code to its methods.

The library afc.rol defines the Actuate Foundation Classes (AFC). e.Report Designer Professional implements these classes and their methods to promote a flexible environment in which report developers can build highly customized reports. If, for example, your report requires a connection to an obscure data source, or the data requires special processing, you can customize or extend the report-building infrastructure by modifying methods or creating new methods and classes.

For more information about e.Report Designer Professional's object-oriented architecture and details about the AFC, see *Programming with Actuate Foundation Classes.*

Customizing reports through programming

With e.Report Designer Professional's suite of tools and wizards, and its rich set of components, you can build a wide range of reports without writing code. If, however, you discover an operation that you cannot accomplish using the graphical tools, you can write procedures or functions, or extend the functionality of component methods to accomplish the task. The programming language that you use is Actuate Basic, which is based on Visual Basic.

The extent of programming required depends on what you want to do. It can be as simple as defining a global variable to store a value, or as complex as creating a new class that extends the report-building architecture. Other programming tasks can include verifying parameter values that a report user provides to control the contents of a report, writing a special data filter to select certain records, or writing report-generation statistics to a log file.

Most of the time, programming involves modifying, or overriding, a component method to extend its functionality. The following example overrides a Start method to check whether a report user typed a value for a parameter and to display a message if the user did not type a value:

```
Sub Start( )
   Super::Start( )

   If DataSource::OfficeState = "" Then
      MsgBox "You must type a value for Office State"
   End If

End Sub
```

As you can see, the code can be very simple. The challenging part is determining which component method to override to add your code. To accomplish this part of the programming task, you need to know:

- The report-generation process

- How the AFC classes work
- The methods of each class, what they do, and how they interact with other methods

To learn about the classes and their methods, and how to override methods, see *Programming with Actuate Foundation Classes.* For more information about Actuate Basic, see *Programming with Actuate Basic.*

Part Two

Designing a report

Chapter

3

Planning and creating a report

This chapter contains the following topics:

- Planning the report
- Deciding how to start a new report
- Configuring report development resources
- Creating a report without a wizard

Planning the report

Before you start to create a report, you should identify the information that you want the report to provide and decide how to present the information. After you make these decisions, it is helpful to design a prototype on paper. It often takes several iterations before you produce a report design that satisfies you and your report users. It is easy to make those changes on a paper prototype.

Deciding the content of the report

This step is the most important. To get started, answer the following questions:

- What is the purpose of the report?

  A purpose statement helps you determine the information that you need. It also gives the report a starting point. The following sentence is an example of a purpose statement:

  ```
  The purpose of this report is to show monthly sales by region,
     then by sales representatives, and to flag the
     representatives whose sales figures fall below a certain
     amount.
  ```

- Who is going to read the report?

  A report can be used by different types of users. A sales report, for example, can be used by sales representatives, sales managers, and the vice president of sales. Each user is interested in different aspects of the report and probably needs access to different parts of a report. Knowing the users of your report helps you plan the report accordingly. With e.Report Designer Professional, you can apply page-level security to ensure that data is available only to those individuals authorized to view the data.

- What information should appear in the report and where is it coming from?

  Much of the information in a typical report is taken directly from data fields in a database, application, or text file. First, you need to know the source of data for the report. Second, you need to understand how the data is structured. If, for example, the data source is a database, it helps to know what tables are in it, the relationship among tables, the columns in each table, the data types, and so on. If necessary, ask your database administrator for this information.

- Does any of the data need to be calculated?

  Some report data comes directly from data fields, such as sales representative names or addresses. Some information must be calculated from other data field values, such as the percentage by which sales figures exceed or fall below a certain amount.

Determining how the report will be viewed

When designing a report, you need to consider and test the environment in which the report will be viewed because how a report appears and prints depends on the user's hardware and software.

You should always design for the final delivery environment. These design decisions include choosing the right fonts and colors, selecting the appropriate page size, fine-tuning the size of report fields and the amount of space between them, and so on.

Consider the following questions:

- Will the report be viewed on the web, as a PDF file, or in the native file format? The appearance of the report differs slightly depending on the viewing environment.
- What printers will be used? The appearance of a printed report can vary, depending on the fonts installed on the printer. Even fonts with the same name supplied by different vendors can affect output.
- Will the report be viewed on UNIX or Windows? A few features are supported on Windows only, such as TIFF images.
- Which is more important, viewing or printing the report? Recognize that there are differences between online and printed reports, decide which is more important, and design for that platform. If, for example, a report will primarily be viewed online, you can add interactive viewing features, such as hyperlinks and search capabilities, and you can specify that the report be displayed on a single scrollable page.
- Is precise printed output required? Mailing labels, for example, need to be printed precisely on sheets of labels.

Deciding if the report will be used by international users

e.Report Designer Professional supports creating reports for a wide range of locales, from Albanian to Ukrainian. In e.Report Designer Professional, a locale defines a set of conventions for typing, displaying, and sorting data. If your report will be used by worldwide users, you need to design the report to support the appropriate locale. If your report will be used by users in multiple countries, you need to go through the process of internationalizing the report so that it works correctly in multiple locales.

For information about designing reports for locales other than the U.S., see *Working in Multiple Locales using Actuate Basic Technology.*

Deciding the layout and format of the report

Once you identify the report's purpose and content, you should have a good idea how to organize and present the information. Consider the following questions:

- Do you need to organize information into groups? If yes, how?
  For example, a monthly sales report may display sales figures by region or by sales representative, or by both. To display both, you would group the information by region first, then list the sales representatives for each region.
- Do you need to summarize the data? If yes, how?
  Reports that present numerical data, such as expense reports, financial statements, and earnings reports, always contain summary sections for totals, averages, or percentages. Decide if this summary information should appear in a list, a cross tab, a chart, or a combination of those options.
- Do you want to compare and analyze summarized data?
  For example, a single page can display the sales figures by region and by product type. The best way to present this data for easy analysis is in a cross tab, which is a row-and-column layout that looks similar to a spreadsheet.
- What information do you need in page headers and footers?
  Reports typically display information in the page header to help users navigate multipage reports. For example, you can display the region name in the header so users know that sales representatives on page *n* are part of region *x*. In the page footer, you can show the page number and the date the report is generated.
- Are there corporate standards that you need to follow?
  If your company produces reports for external use, such as financial statements for clients, it is likely that a report that you create needs to use corporate styles. Corporate styles typically dictate the fonts, colors, and page layouts that you can use.
- Are there corporate resources that you can use?
  Often, a company that produces many reports maintains a library or libraries of common report objects, such as company logo, page styles, standard headers or footers. e.Report Designer Professional makes it easy for a report developer to publish shared objects to a library and for others to include this library in their report designs.

Designing a prototype on paper

After you make all the decisions described in the previous sections, you should design a prototype. Use any tool with which you are comfortable, such as a word processor, graphics program, or pen and paper.

A prototype should show approximately what the finished report will look like, including the report title, page header and footer information, and all the fields in the body of the report. Using a prototype to get feedback and approval from your primary users can save you time. You do not waste time creating a polished report that contains the wrong information or layout.

A paper prototype is especially useful when you are first learning e.Report Designer Professional. With a blueprint in hand, you can focus on learning and using the tool, rather than trying to design and learn at the same time.

Deciding how to start a new report

When you start e.Report Designer Professional, you have several options for creating a new report. You can:

- Use a report creation wizard.
- Start with a report template.
- Start with a blank report design.

The option that you choose depends on your preferred way of working and the availability of reports or templates that resemble your report type. Some users find it easier to start with a wizard or a template, then add, delete, or change items. Others prefer to start with a clean slate.

Using a wizard

e.Report Designer Professional provides two wizards that help you create basic reports quickly. They are helpful when you want to create a simple report or the basic structure for a more complex report. All you have to do is choose File➤New, and choose the wizard that most closely matches the type of report that you want to create. The wizard walks you through the steps of creating the report, from selecting the data source to selecting a basic layout.

Once you finish creating a report with a wizard, you cannot return to the wizard to make changes to the report. You must use the Layout window to make the changes manually. If the changes are major, such as changing a tabular report to a columnar report, then it is easier and faster to start over with a wizard.

About the Quick Report wizard

The Quick Report wizard creates a report that lists all the records and field data from a single table of your choice. For example, if an employee table contains 100 employee records that each contain four fields, employee name, ID, phone number, and e-mail address, the Quick Report wizard creates a report that displays all the fields for all 100 records. The report displays the records in the order in which they were typed into the table.

Use this wizard to start with a report that displays all the data from a table. The Quick Report wizard displays all fields from the table. If your report requires data from multiple tables or if you want more control to select specific fields or sort the data, use the Listing Report wizard.

Figure 3-1 shows examples of some of the different report formats that you can create with this wizard.

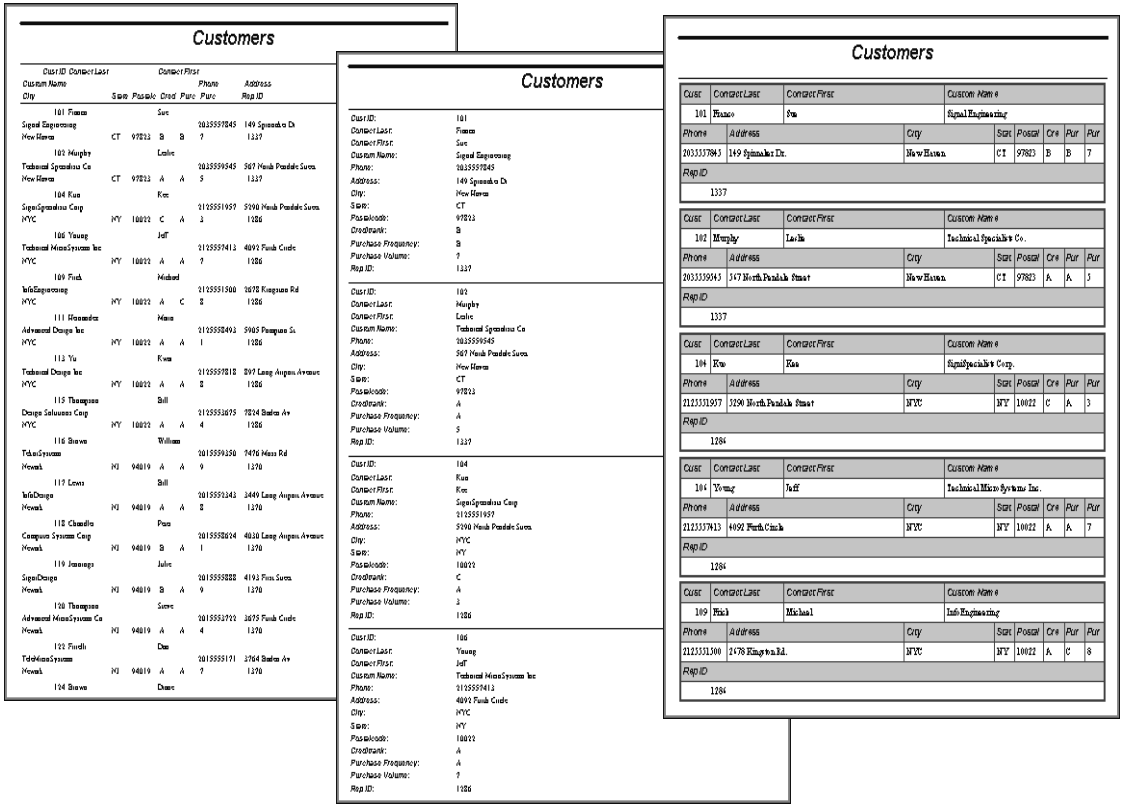


Figure 3-1 Report format examples

About the Listing Report wizard

The Listing Report wizard creates a report from fields that you select from one or more tables of your choice. It also enables you to specify how to group and sort the data. The Listing Report wizard gives you flexibility in selecting specific data from multiple tables and in organizing the data.

Figure 3-2 shows examples of the report formats that you can create using the Listing Report wizard.

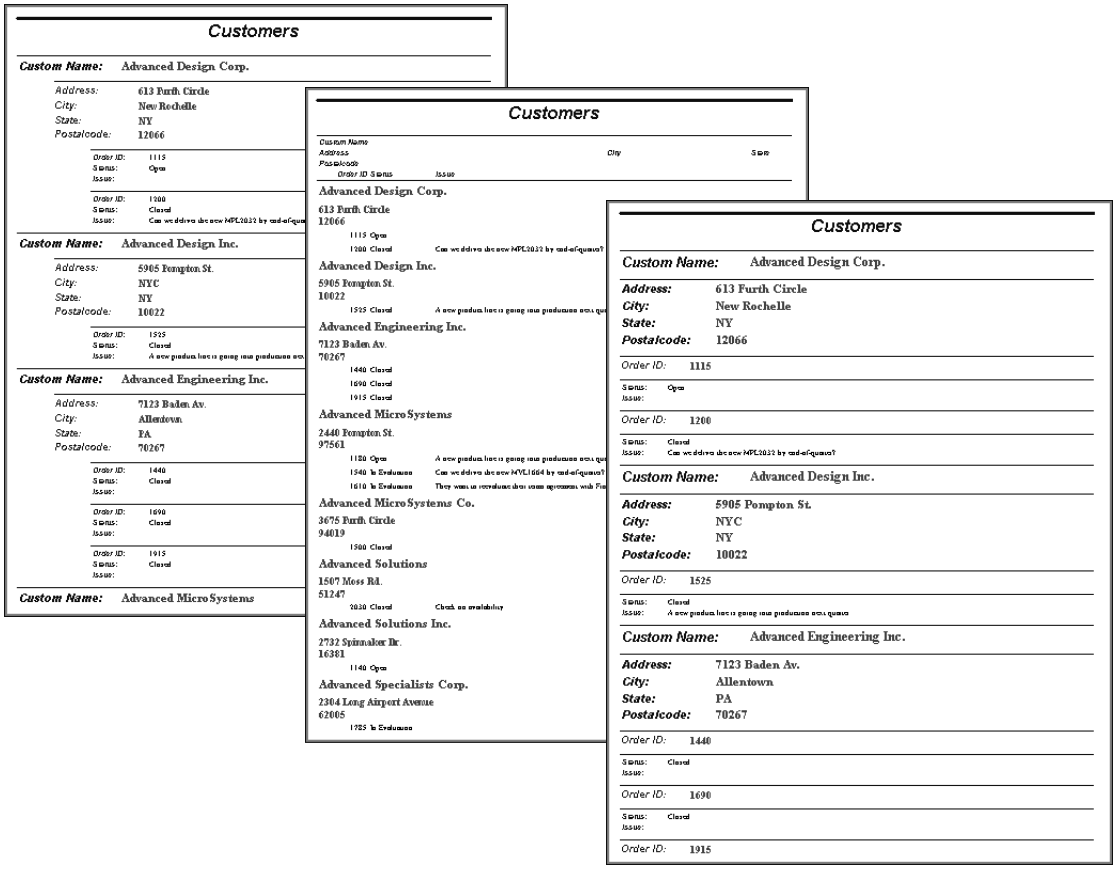


Figure 3-2 Report format examples created using the Listing Report wizard

Starting with a template

If you work in an organization that creates and maintains templates for different types of reports, start with a template and build your report from there. This approach is also efficient for managers responsible for report development who may have relatively junior staff members. Using templates gives beginners a leg up on the report development process. Templates are typically skeleton reports that contain the commonly used elements, such as company name and logo, standard page design, data connections, and so on.

Report designers who create templates make the templates available to other designers by specifying them in a configuration file. If you load this configuration file, the templates appear as report creation choices when you choose File➤New, as shown in Figure 3-3.

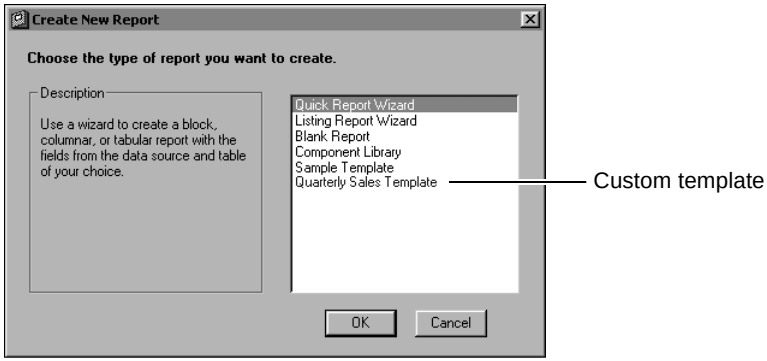


Figure 3-3 Selecting a report template

Starting with a blank report design

Start with a blank report design for these situations:

- The report that you want to create is different from the report types available using the wizards or templates.
- You are well versed with e.Report Designer Professional and want full control of creating a report from the start.
- You want to understand the potential of the tool.

To start with a blank report design, choose File➝New, then choose Blank Report.

Configuring report development resources

To help new users get started quickly, e.Report Designer Professional's default report development environment includes a configuration file that specifies paths to sample component libraries, report templates, data sources, and other resources for use in report designs. Before you begin developing production reports, you should specify a configuration file appropriate for your development environment.

If you do not replace or disable the default configuration file, all reports that you create include some components, such as a default connection component, and sample libraries.

Creating a report without a wizard

Although the wizards simplify the report creation process by presenting a step-by-step approach, they limit what you can do. You must accept the fonts and

colors the wizard uses and you are limited to a few basic layouts. It is likely that you have to refine the reports that the wizards create. Similarly, if you start a new report from a template, you need to modify the report.

It is important to learn the steps involved with creating a report without a wizard because you need this information to refine or modify reports created from wizards or templates.

This section describes the typical steps to create a basic report without using a wizard. Use this information as a quick reference when you build a report manually. Each step is described in more detail in later chapters.

Starting a new report design

1. Choose File➤New.

   Create New Report appears.

2. Choose Blank Report.

   A new report design appears.

When you start with the blank report option, the report design that appears is not completely blank. The report structure displays a few components. Some of these components—the connection and data stream components, for example—are required by all reports. Others are components for displaying common report elements, such as a report title, page numbers, and a report generation date.

Accessing data

After you start a new report design, the next step is to set up the report to access data. To accomplish this task, you need to set up a connection to the data source, then build a query to specify the data to use in the report.

These steps assume that you have already configured your computer to access the data source. For example, if the data source is a database such as Oracle, you should have installed and configured the Oracle client software on your computer. Similarly, an ODBC data source requires that you first configure the data source using an external tool, such as the Windows ODBC Data Source Administrator.

For a complete discussion of how to configure, use, and access data sources in your report, see *Accessing Data using e.Report Designer Professional.*

Setting up the connection

To set up the connection between the report and the data source, do the following:

- Choose Tools➤Database Connection.

- Depending on whether a predefined connection is available, choose one of these options:
  - Choose from a list of databases
  - Build a new database connection
- Provide the connection information your database server requires.

Specifying the data to use in the report

After you set up the connection to the database, you can access the database to specify which data to include in the report. This step consists of the following substeps:

- Selecting a database table or tables
- Selecting table columns
- Filtering records

Selecting tables

1. Choose View➛Data, then provide the necessary connection information.
2. In Database Browser, click the + icon next to the database to display all the tables.
3. Drag the required tables from Database Browser and drop them in DataSource—Query Editor.

Selecting columns

After you select the tables, you can select the fields whose values that you want to use in the report.

1. Select the Columns tab in the lower pane.
2. Select the columns to use in the report, using one of the following methods:
   - Drag the columns from the upper page and drop them in Column Name. To select all the columns, drag and drop the asterisk (*).
   - In Column Name, use the drop-down list to select the column.

Filtering records

You have the option of selecting only certain records to use in your report. For example, rather than get information about all customers, you can limit the information to customers in a certain state.

1. Select the Conditions tab.
2. In Column Name, select the column to which you want to apply a condition.

3 In Query Expression, type the condition to apply.

For example, if you selected a column called customer.state, and you want your report to include only customers from California, type the following in Query Expression:

```
= CA
```

Grouping and sorting data

The next step after you build your query is to organize the data by grouping related information and sorting individual records.

1 Choose View➛Design.

2 Choose Tools➛Sorting and Grouping.

Sorting and Grouping—Grouping displays the fields available for grouping, as determined by your query.

3 Select a field on which to group.

You can select multiple fields on which to group to create a hierarchy of groups. For example, you can group customers by state, then group orders by customer.

4 Sort individual records. For example, at the detail level, each order may list all the item records. You can sort these item records by a field, such as item code or purchase quantity.

1 Choose Sorting.

2 In Fields to Sort By, click a field.

3 Select a field from the drop-down list.

4 Specify the sort order by selecting Ascending or Descending.

Placing data

After you specify the data to use in your report and how to group and sort it, you are ready to place data in the report design to display it.

1 Choose View➛Fields to display the list of fields.

2 Drag each field whose values that you want to display in the report and drop it in the desired frame.

3 To insert calculated data, choose Insert➛Computed Field.

Summarizing data

You can include aggregate information, such as counts, averages, and totals, at the beginning and end of the report and before and after each grouped section.

1. Choose Insert➤Total and place the component in the desired Before or After frame.
2. Select the field whose values to summarize and select one of the aggregate functions.

Formatting data

When you place data components, e.Report Designer Professional uses default font styles, sizes, and colors that are optimal for display and printing on multiple platforms. To change the format of data or a component, use one of these techniques:

- Apply text formatting using the Format toolbar shown in Figure 3-4.

Figure 3-4 Format toolbar

- Reposition and align components using options on the Format menu.
- Change the component's properties using the Properties window.

Specifying a report title

One of the default components in every new report design is a label component that displays Report Title at the top of every page in a report. All you have to do is edit the text to display an appropriate title for your report.

1. In Report Structure, expand the PageList and PageStyle components.
2. Select Content—ReportTitle.
3. Change the text to your report's title, using one of these methods:
   - Set the component's Text property to the report title.
   - In Layout, double-click the component, then edit the text. If the Properties window appears, wait a bit longer between the first and second clicks.

Setting page size, orientation, and printable area

e.Report Designer Professional uses default paper size, orientation, and margins. To change these settings:

1. Choose Tools➤Page Setup. Page Setup appears as shown in Figure 3-5.

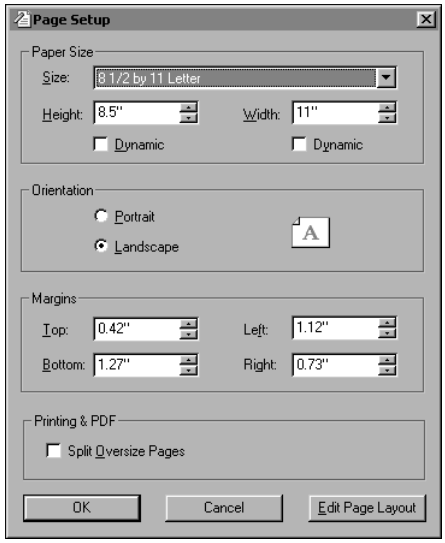


Figure 3-5 Changing settings in the Page Setup window

2 Make the desired changes. Choose OK.

Naming an Actuate report

When you finish creating a report, save it and give it a file name that describes the purpose or contents of the report. It is important to avoid special characters in the file name, such as /, \, ;, *, angle < >, or ?.

If you use disallowed special characters in a report file name, Actuate Basic cannot correctly interpret the name. When you run the report, the run fails and the Output window displays an error message similar to the one in Figure 3-6.

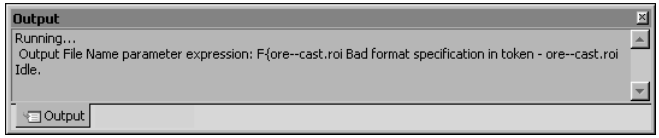


Figure 3-6 Output error message

Report file names support using spaces and the -, _, [ ], and { } characters. Use { } only under certain circumstances. For more information about using curly braces, see *Using Information Console.*

Performing other report design tasks

There are many more tasks that you can accomplish using e.Report Designer Professional to build more versatile and sophisticated reports. Some of these tasks include:

- Creating charts
- Creating cross tabs reports
- Creating report parameters for user input
- Building subreports
- Adding web functionality into your report
- Adding hyperlinks to link different reports or parts of a report
- Using page-level security to manage user access

Chapter

4

Laying out a report

This chapter contains the following topics:

- About laying out a report
- Grouping report data
- Placing data in the report
- Sorting report data
- Enhancing the appearance of a report
- Testing a report design

About laying out a report

You can present information in a report in many ways. You can display information in a simple list, a table, a chart, a cross-tabulation report, in multiple subreports, or in any layout that you can imagine. Laying out a report entails placing data on the page and organizing the data in a way that helps the report user easily read and understand the report's content.

Figure 4-1 shows a few examples of report layouts that you can create using e.Report Designer Professional.

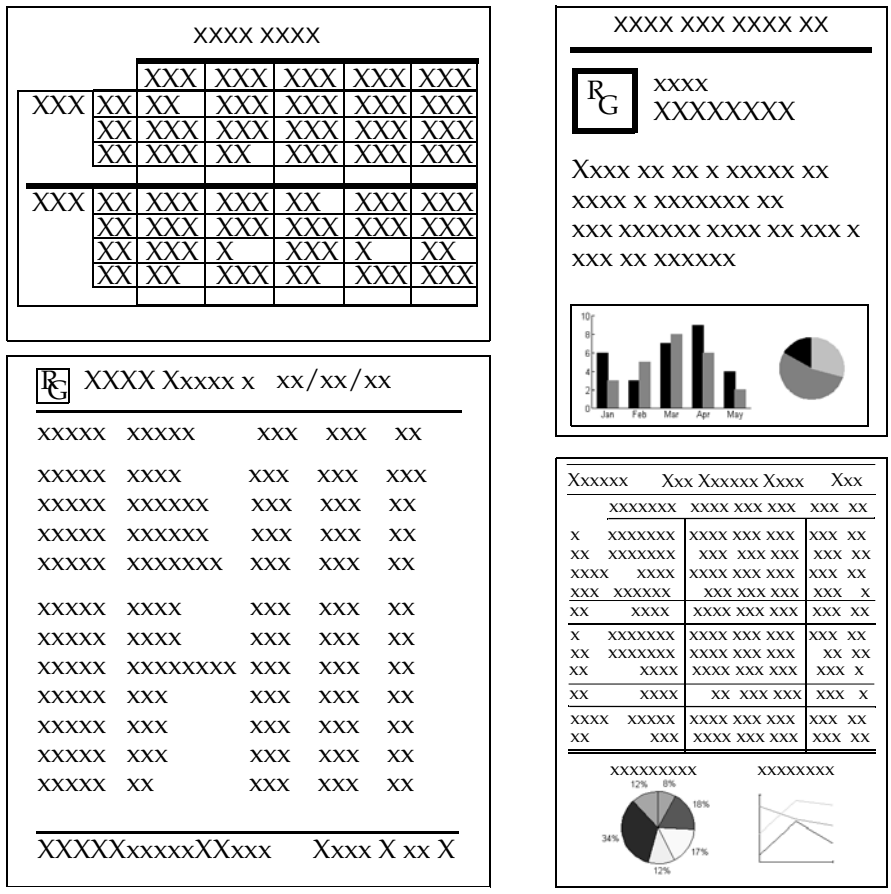


Figure 4-1 Report layout examples

Because there are infinite ways to lay out a report, it helps to work from a paper design. If you try to design while you create the report layout, you risk losing

track of the data to place in the report or you can finish laying out one part of the report before realizing that you can better present the data using another layout.

Preparing to lay out a report

Before you lay out a report, you must first set up the data source for your report. If the report uses data from a relational database, for example, you must first complete the following tasks:

- Set up a connection to the database.
- Define a query, which specifies what data to retrieve from the database.

For detailed information about these tasks, see *Accessing Data using e.Report Designer Professional.*

Understanding the layout process

After you set up the data source for your report, you are ready to lay out the report. The following is an overview of the process:

- Create the basic structure of the report. To create a simple report that groups and lists data, you can use the Grouping and Sorting tool, which creates the report structure for you. To create a more complex report that contains subreports or conditional sections, you create the basic structure by placing section components in the report design.

  This chapter describes the procedures for laying out a listing report, which is a typical and basic report.
- Define how to group data, if grouping is required. Most reports use groups as a way to sort data. A sales report, for example, can display orders in groups by customer, and display customers in groups by state.
- Place data in the report.
- Sort data in the detail section of the report, if sorting is required.
- Enhance the appearance of the report by adding text labels, pictures, boxes, lines, and so on.

Grouping report data

It is common for reports to present data that is sorted and organized into meaningful groups. A customer orders report, for example, can organize orders by customers, then organize customers by state. Reports that organize data into logical groups are called grouped reports or grouped listing reports.

A grouped report presents data that is structured to be more meaningful to the user than a simple list of the data would be. Grouped reports make data easier to locate, read, and use. When you group data, you can:

- Add titles at the beginning of each group to make each group readily identifiable.
- Sort data by one or many fields. For example, you can sort alphabetically by customer name, then sort numerically by order ID within each customer group.
- Add subtotals, counts, averages, or other summary information at the beginning or end of each group. This feature makes complex data more useful to report users.
- Insert page breaks before or after each group. This feature supports controlling access to data on a page-by-page basis.

Compare the following reports shown in Figure 4-2.

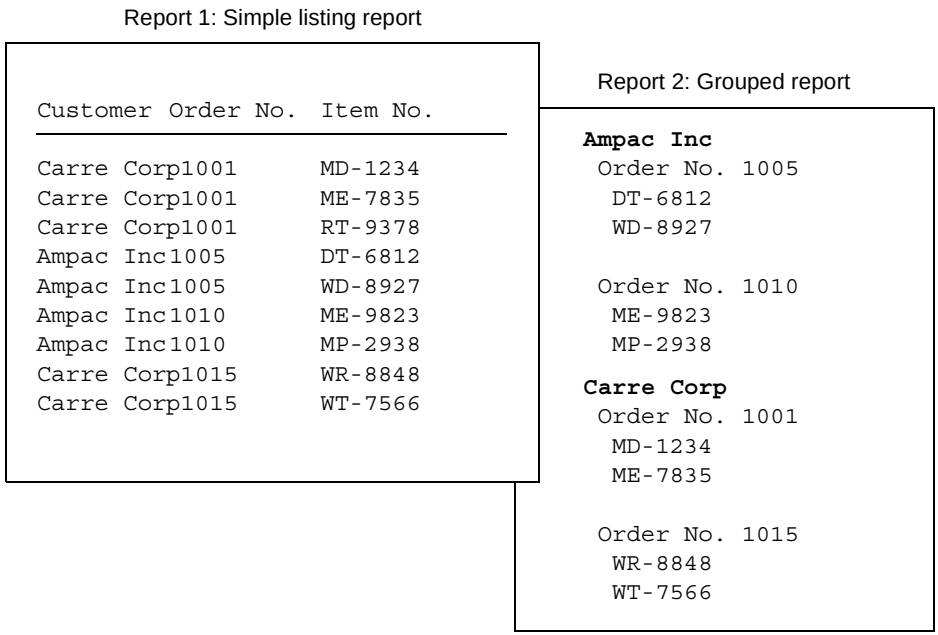


Figure 4-2 Report sample comparison

The first report displays customer order information in a simple list. It sorts records, also called data rows, by order number, which is the way the database stores them. Notice the repeated customer name and order number information.

The second report presents the same customer order information in a layout that is more effective for viewing and analysis. It sorts data first by customer, then by orders. To sort on these two fields, you create groups of customers and orders.

The customer group is the outer, or topmost, group. The order groups are within each customer group, and the order items are within each order group.

About sorting and sort keys

Figure 4-3 shows how grouping affects the sorting of data rows.

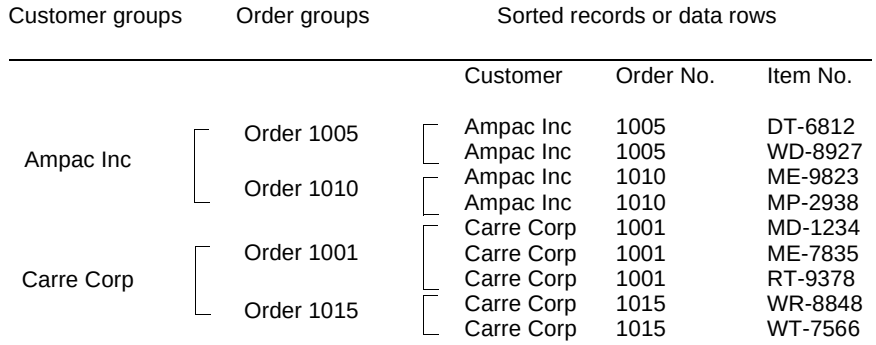


Figure 4-3 Grouping and sorting of data rows

Each data row represents an item in a sales order. The rows are sorted according to order number. The set of rows that contains all the items for one order is called one order group. The order groups, in turn, are sorted according to customer. The set of rows that contains all the orders for one customer is called one customer group.

The field on which data rows are sorted is called a sort key. Each group in a report has a sort key. For a customer group, for example, the sort key can be customers.customerName, where:

- customers is the table name.
- customerName is the field name.

Creating a group

You can create as many groups as you want in a report. The example report in "Grouping report data," earlier in this chapter contained two groups, orders and customers. You can create a third group, state, to sort customers by state. As a practical matter, however, a report that contains too many groups makes the report difficult to read.

To create a group, use the Grouping and Sorting tool to select the field or fields on which to group. The instructions in this section include an example of creating a report that groups orders by customer, then by order number. The example report uses the sample database, sfdata.mdb, as its data source.

How to create the CustomerOrders example report and build the query

1. Create a new report with a connection to the Sfdata sample database:
   1. Choose File➤New.
   2. In Create New Report, select Blank Report. Choose OK. A new report design appears.
   3. Choose Tools➤Database Connection to connect to sfdata.mdb.
   4. In Database Connection, select Choose from a list of predefined databases. Choose Next.
   5. In Database Connection—Choose Database, select SFData Sample Database. Choose Next.
   6. In Database Connection—Properties, choose Finish. The report design appears.
2. Build a query:
   1. Choose View➤Data.
   2. In Database Login, choose OK. You do not need a user name or password to connect to sfdata.mdb.
   3. From Database Browser, drag the customers table to DataSource—Query Editor as shown in Figure 4-4.

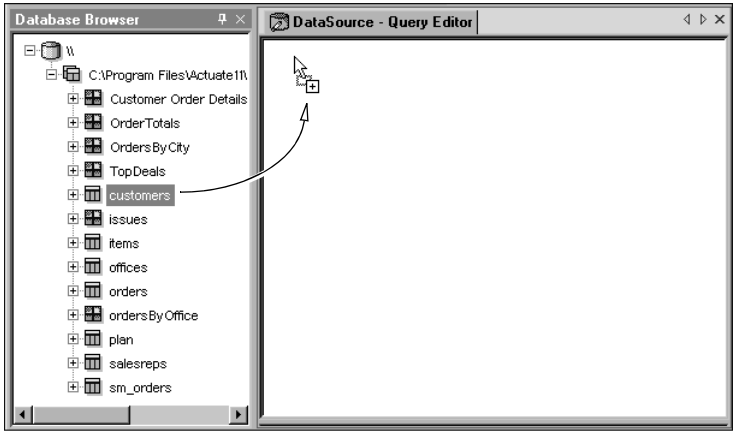


Figure 4-4 Building a query

The columns from the customers table appear in DataSource—Query Editor.

4 Drag the orders and items tables to DataSource—Query Editor. The columns from the orders and items tables appear in DataSource—Query Editor. Drag the bottom edge of each table down to see all the fields as seen in Figure 4-5.

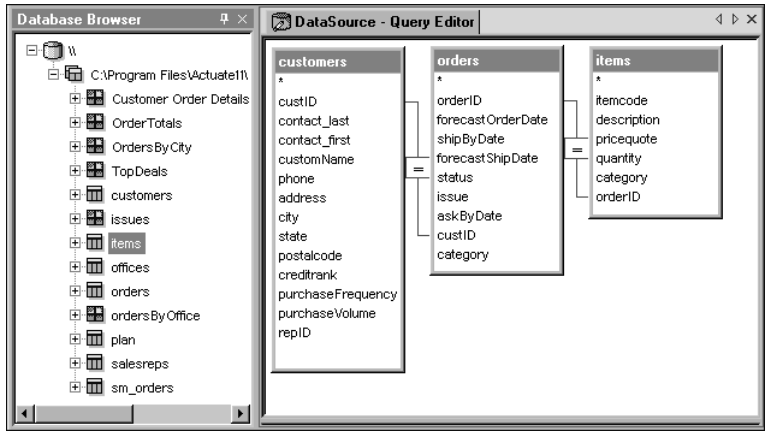


Figure 4-5 Showing entire field within a table in Query Editor

5 Add all the columns from the customers table by dragging the asterisk (*) at the top of the customers list and dropping it into Column Name as indicated in Figure 4-6.

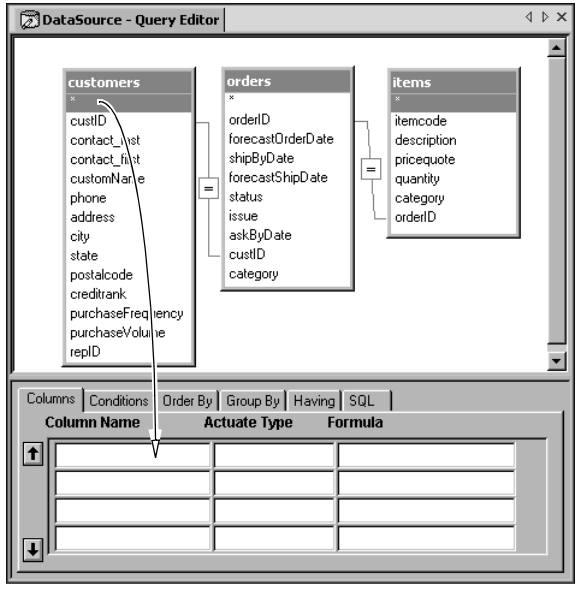


Figure 4-6 Moving data to a new report

The column names from the customers table appear in Column Name.

6 Add all the columns from the orders and items tables to Column Name by dragging the asterisk (*) from each table and dropping it into Column Name.

You created a query that selects all the columns from the customers, orders, and items tables.

3 Choose View➛Design to return to the design perspective. The design perspective appears.

4 Save the report design as CustomerOrders.rod.

How to create a group

This procedure assumes that you use the CustomerOrders example report, CustomerOrders.rod, you created in "How to create the CustomerOrders example report and build the query," earlier in this chapter.

1 Choose Tools➛Grouping. Grouping and Sorting—Grouping displays the fields available for grouping, as determined by your query and shown in Figure 4-7.

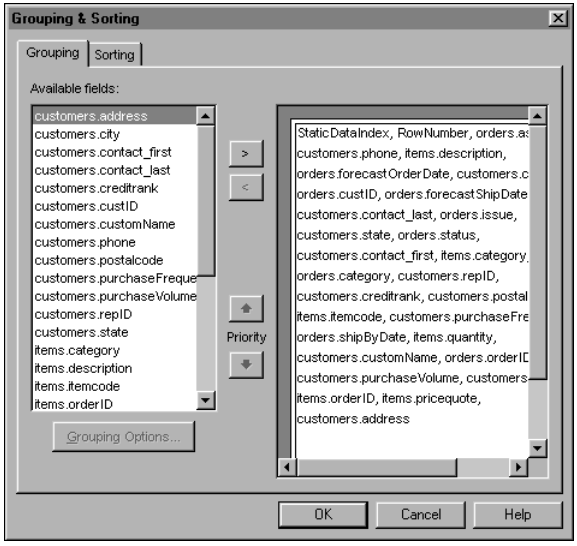


Figure 4-7 Grouping and sorting

2 In Available fields, select a field on which to group, then choose the single right arrow. For the CustomerOrders example report, select customers.customName.

3 Select the next field on which to group, then choose the single right arrow. For the CustomerOrders example report, select orders.orderID as is shown in

Figure 4-8. The right side shows the hierarchy of the groups, which is the order of groups within groups.

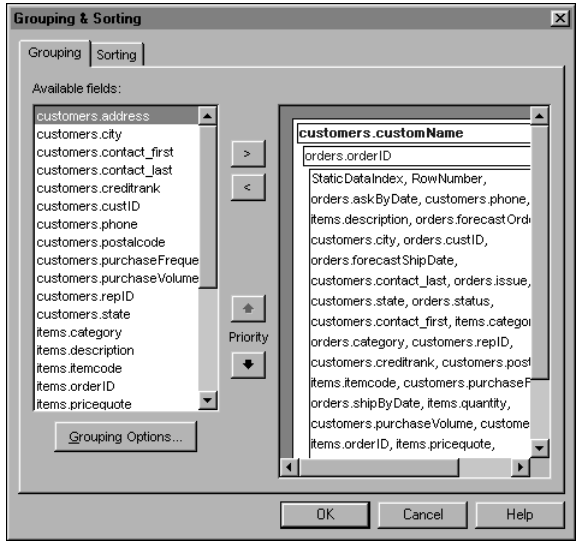


Figure 4-8 Hierarchy of groups

To change the order of groups, use the up and down arrows.

4 When you finish creating groups, choose OK.

Examining how groups appear in the report structure

Look at the structure of the report that you created in the CustomerOrders example report design. Based on your grouping choices, e.Report Designer Professional added group section components to the report design. Figure 4-9 and Figure 4-10 show an example of a report's structure, showing empty slots, before and after creating groups.

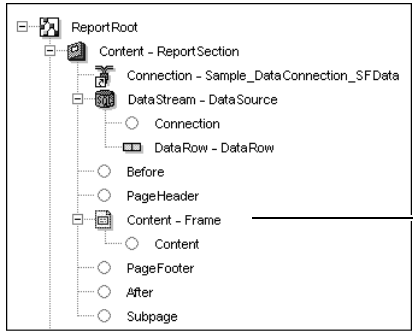


Figure 4-9 Report structure before creating groups

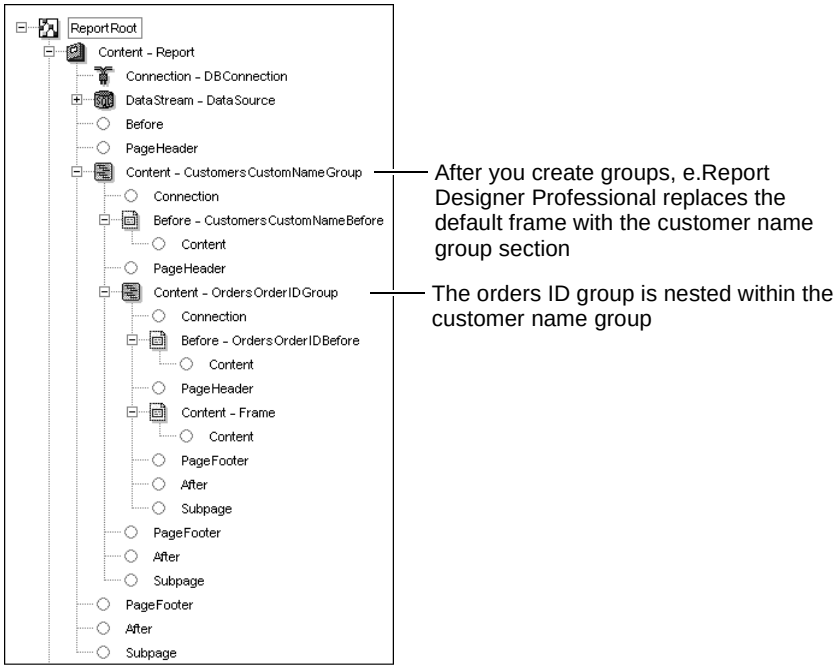


Figure 4-10 Report structure after creating groups

Now, look at the properties of a group section. Note the value for the Key property. e.Report Designer Professional sets the value of this property based on the field on which you choose to group. The property value is the group's sort key. Figure 4-11 shows an example of a group section's properties.

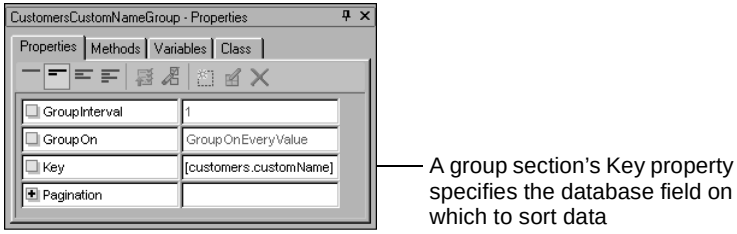


Figure 4-11 Key property

Grouping report data by intervals

When you create a group, e.Report Designer Professional's default behavior is to sort data by a single value, such as a customer name, an order ID, or a date. In the CustomerOrders example report, each customer name starts a new group, and each order ID starts a new group within the customer group.

Sometimes, however, it is more useful to group data by a specific interval. A sales report, for example, can present sales by quarter rather than by a certain date.

Similarly, you can group data in a shipping report by weeks or months rather than by a date. Compare the reports shown in Figure 4-12 and Figure 4-13.

Shipping Schedule

Ship by	Order ID	Customer name
04/05/2004	1440	Advanced Engineering
04/05/2004	1445	Technical Design Corp.
04/06/2004	1355	Signal MicroSystems
04/07/2004	1400	InfoSpecialists Inc.
04/08/2004	1320	CompuDesign Co.
04/12/2004	1410	Brittan Design Inc.
04/13/2004	1420	Exosoft Corp
04/14/2004	1250	Technical Solutions Inc.
04/14/2004	1325	SigniMicro Systems
04/15/2004	1500	CompuBoards
04/16/2004	1275	Technical Design Corp
04/19/2004	1515	TeleMicro Systems
04/21/2004	1550	InfoBoards Inc

Figure 4-12 Report 1: Grouped by single date values

Weekly Shipping Schedule

Week of 04/05/2004

Ship by	Order ID	Customer name
04/05/2004	1440	Advanced Engineering
04/05/2004	1445	Technical Design Corp.
04/06/2004	1355	Signal MicroSystems
04/07/2004	1400	InfoSpecialists Inc.
04/08/2004	1320	CompuDesign Co.

Week of 04/12/2004

Ship by	Order ID	Customer name
04/12/2004	1410	Brittan Design Inc.
04/13/2004	1420	Exosoft Corp
04/14/2004	1250	Technical Solutions Inc.
04/14/2004	1325	SigniMicro Systems
04/15/2004	1500	CompuBoards
04/16/2004	1275	Technical Design Corp

Week of 04/19/2004

Ship by	Order ID	Customer name
04/19/2004	1515	TeleMicro Systems
04/21/2004	1550	InfoBoards Inc

Figure 4-13 Report 2: Grouped by weekly intervals

The following sections describe in more detail how grouping by interval works for string, numeric, and date and time data.

As the reports show, grouping by interval provides the following benefits:

- Organizes a long report into shorter, more readable pieces
- Summarizes data further for more effective analysis

Grouping string data by intervals

When you group string data by interval, the interval that you specify is a prefix of a particular length. For example, if a customer group sorts customers by name, you can group customers by the first letter of their names, or the first two letters, or the first three letters, and so on.

You typically group by the first letter to group names by letters of the alphabet. In a customer list, for example, you might want to group all customers whose names begin with A under the heading A, all customers whose names begin with B under the heading B, and so on.

You typically group by multiple letters to group items whose names contain special prefixes as a way to classify or categorize items. A computer parts vendor, for example, might use the prefix ME for all memory chips, CP for CPU boards, MO for monitors, and so on. In this case, creating a computer parts list that groups names by the first two letters lends itself to logical groupings by part type.

Figure 4-14 show the results of grouping names by the first letter, two letters, and three letters. The lines separate the groups.

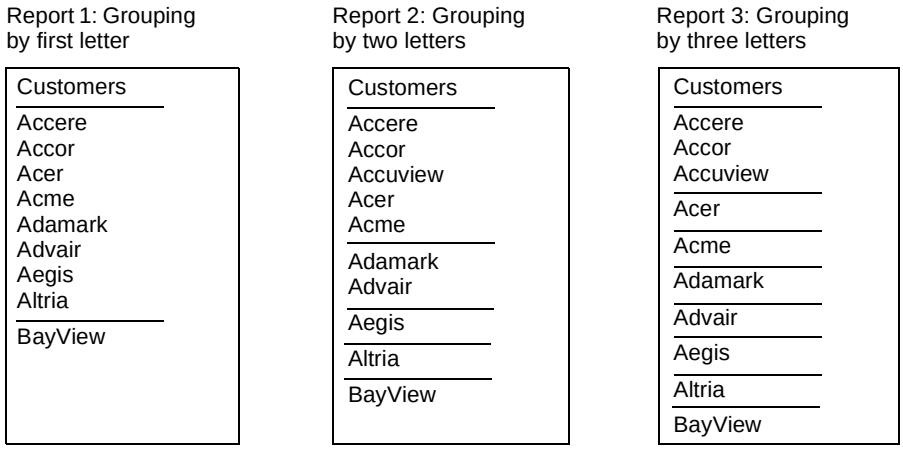


Figure 4-14 Report examples grouped by letters

Grouping numeric data by intervals

When you group numeric data by intervals, you group by a range of numbers. For example, if an order group sorts orders by numeric ID, you can group the orders by intervals of 10, 50, 100, 1000, and so on.

The interval that is best for any set of numeric data depends on the range of numeric values. If the numbers range from 100 to 200, it makes sense to group in intervals of 10s. If the numbers range from 100 to 1000, you might want to group in intervals of 100s.

Figure 4-15 show the results of grouping numbers by intervals of 10s, 100s, and 1000s. The lines in the illustration separate the groups.

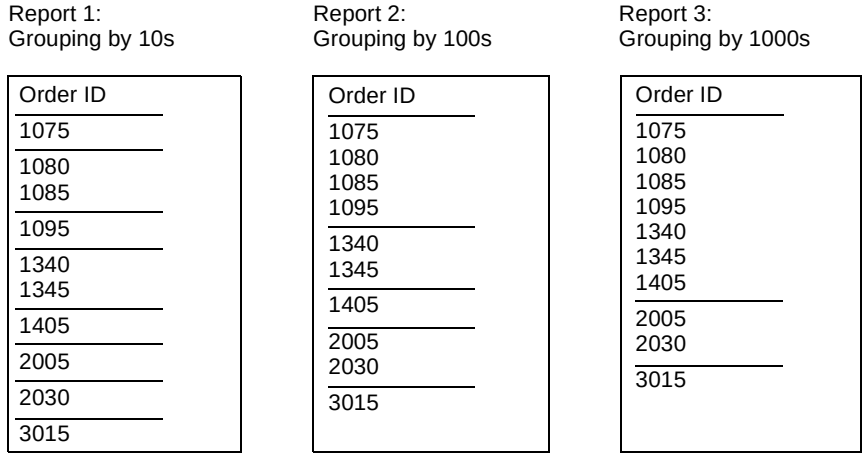


Figure 4-15 Report examples grouped by numeric intervals

Grouping date and time data by intervals

When you group date and time data by intervals, you group data by time periods, such as hours, days, weeks, months, and so on. Grouping by time periods is useful for reports that display information with a time or schedule focus, such as weekly shipping schedules or sales figures by quarters. For an example of such a report, see "Grouping report data by intervals," earlier in this chapter.

Deciding which tool to use to group data by interval

You can group data by intervals in one of the following ways:

- Select a grouping interval through grouping options in the Grouping and Sorting tool. Use this option to select from a list of common intervals.
- Set the group section's GroupOn and GroupInterval properties. Use this option to specify an interval that is not available with the Grouping and Sorting tool.

Figure 4-16 compares the two ways of grouping data by interval. The grouping interval selection, 2 Initial Letters, in Grouping Options is equivalent to setting the group section's GroupOn property to GroupOnPrefix and GroupInterval to 2.

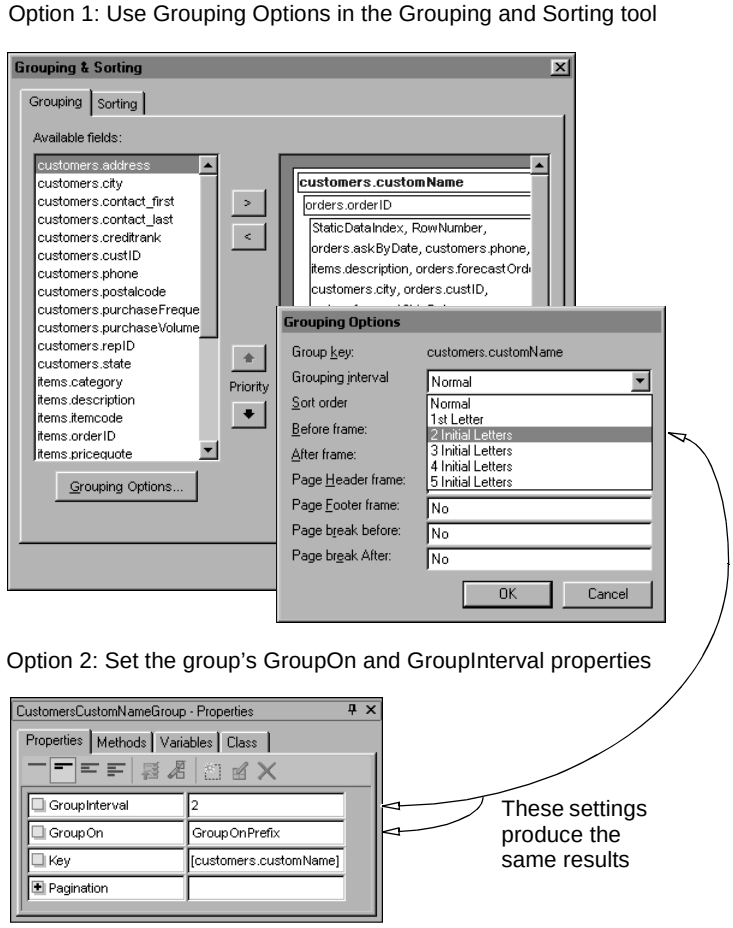


Figure 4-16 Grouping options one and two

How to group data by intervals using the Grouping and Sorting tool

The following instructions assume that you already created groups:

1. Choose Tools➛Grouping. Grouping and Sorting appears.
2. In Grouping and Sorting—Grouping, select the group whose data you want to group by interval.
3. Choose Grouping Options.

4 In Grouping Options, for Grouping Interval, select one of the interval options. Choose OK.

5 In Grouping and Sorting—Grouping, choose OK.

How to group data by intervals using the grouping properties

The following instructions assume that you already created groups:

1 In Report Structure, right-click the group whose data you want to group by interval, then choose Properties.

2 In the Properties page for the group section, set the GroupOn property to a value that specifies the grouping condition that you want. Select one of the following values:

- For string data, select GroupOnPrefix.
- For numeric data, select GroupOnInterval.
- For date data, select one of the GroupOn<time period> values.

3 Set the GroupInterval property to a number that represents the number of units to include in each group. For example, if you selected GroupOnWeek for the GroupOn Property, specify 2 for GroupInterval to group data in two-week periods.

Grouping on a custom key

Typically, you group data based on field values as they appear in the database. Sometimes, however, you want to group data by values that do not correspond to values in a database field. For example, you can group on values from a combination of two fields or on values that are parsed from one field.

Consider, for example, an accounting report that you want to group by customers' main account numbers. The database, however, stores account numbers in the form of XXXXXYY, where XXXXX is the main account and YY is the subaccount. To group by the main account only, create a custom group key to group on only the XXXXX values.

Use the following procedure for grouping on a custom key:

- Create or select the group on which to apply custom grouping.
- Set the group's GroupOn property to GroupOnCustom.
- Write code that specifies the custom key. The place to write this code is in the GetKeyValue( ) method of the group. This method returns the value that e.Report Designer Professional uses as the group key for the group. For more information about GetKeyValue( ), see *Programming with Actuate Foundation Classes*.

The following code example extracts a part of each account number to use as a group's key value:

```
Function GetKeyValue( row As AcDataRow ) As Variant
   GetKeyValue = Int( row.GetValue( "customerAccounts.AccountID"
      ) / 100 )
End Function
```

How to group on a custom key

1. Drag a group section from Toolbox—Structure and drop it into the appropriate slot. Component Properties appears.
2. For the GroupOn property, select GroupOnCustom, then choose OK. The group section appears in the report structure.
3. Right-click the group section that you just created, then choose Properties. The Properties page for the group section appears.
4. Choose Methods. The Methods page appears, displaying a list of methods for the group section.
5. Select Function GetKeyValue(row As AcDataRow) As Variant.

If you do not see the method in the list, choose Methods You Can Override. Select Function GetKeyValue(row As AcDataRow) As Variant.

6. Choose Override. The method editor displays the required lines of code for overriding the method as shown in Figure 4-17.

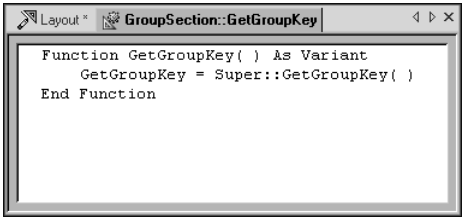


Figure 4-17 Method editor

7. Add the group key code to the method.

Placing data in the report

When you finish creating groups for a report, you essentially finish creating the report structure. Then, you are ready to place data in the report. If you run the report before placing data, it displays nothing.

e.Report Designer Professional divides a report and its group sections into several parts, as shown in Figure 4-18. These parts, called slots, correspond to familiar document parts, such as headers and footers.

You place data in different slots to determine where and how often the information appears in the report.

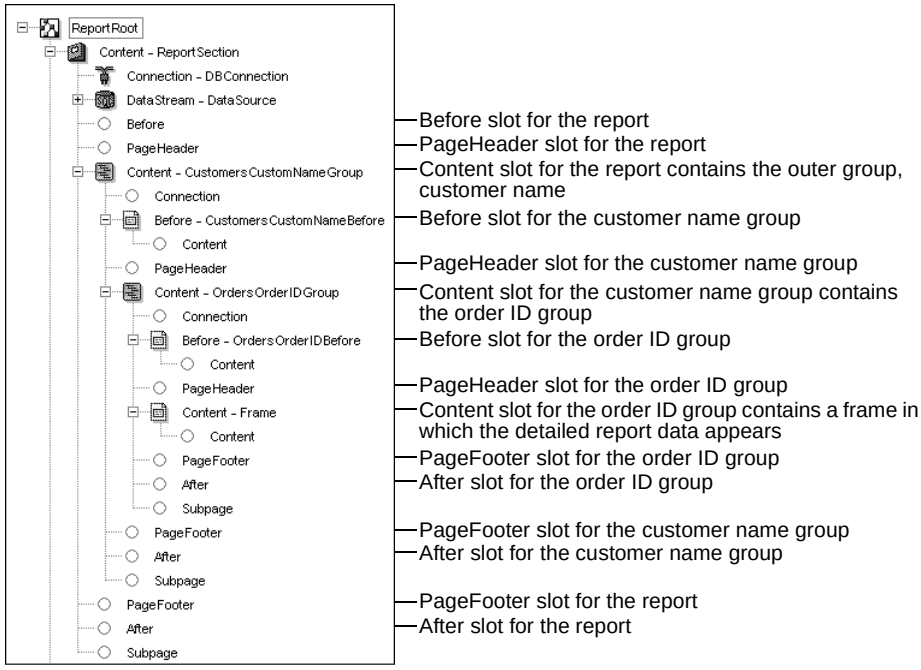


Figure 4-18 Overview of sections of a report

If you create a complex report with subreports or multiple levels of groups, the structure and layout views of the report are equally complex because e.Report Designer Professional divides each report and each group into multiple parts. Using + or -, expand or collapse components to see as much or as little of the report structure as you need at any time.

You can also hide empty slots to reduce the amount of information the report structure displays. To do so, right click in the white space in Report Structure. Then, in the context menu that appears, deselect Show Empty Slots.

Deciding where to place data

Table 4-1 describes the types of data that you place in each of the slots in a report or a group. You do not have to place data in all the slots.

Table 4-1 Data placement in a report

Slot	For a report	For a group
Before	Text or data appears once at the beginning of a report. Examples: ■ Display the report title. ■ Display a chart that summarizes data for the entire report.	Text or data appears once at the beginning of a group. Examples: ■ Display the group name field, such as an order number for an orders group. ■ Display column headings, such as Item, Quantity, and Price, above the fields in an order group.
PageHeader	Text or data appears at the top of each page, except the first. Example: Repeat the report title in a different format from the format of the report title on the first page.	Text or data appears at the top of each page, except the first, as the group continues. Examples: ■ Display a customer name at the top of every page of a multipage customer group section. ■ For the innermost group that displays the main content, repeat the column headings, such as Item, Quantity, and Price, to keep readers oriented.
Content	The Content slot contains the topmost group. Example: For a report that groups orders by customer, then by orders, the Content slot contains the customer group.	The Content slot contains another group, or the detailed report data. Example: For a customer group, the Content slot contains the orders group. The slot of the order group, in turn, displays all the items in each order.
PageFooter	Text or data appears at the bottom of every page, except the last. Example: Display text, such as: Report continues on the next page	Text or data appears at the bottom of every page, except the last, of a group that spans multiple pages. Example: Display text, such as: Order continues on the next page

Table 4-1 Data placement in a report

Slot	For a report	For a group
After	Text or data appears once at the end of the report. Example: Display aggregate information, such as grand total of all customer orders.	Text or data appears once at the end of the group. Example: Display aggregate information about the group, such as total order amount.

Figure 4-19 and Figure 4-20 show the first and second pages of the generated report, Detail.roi, and details about the slots in which data was placed to achieve the results.

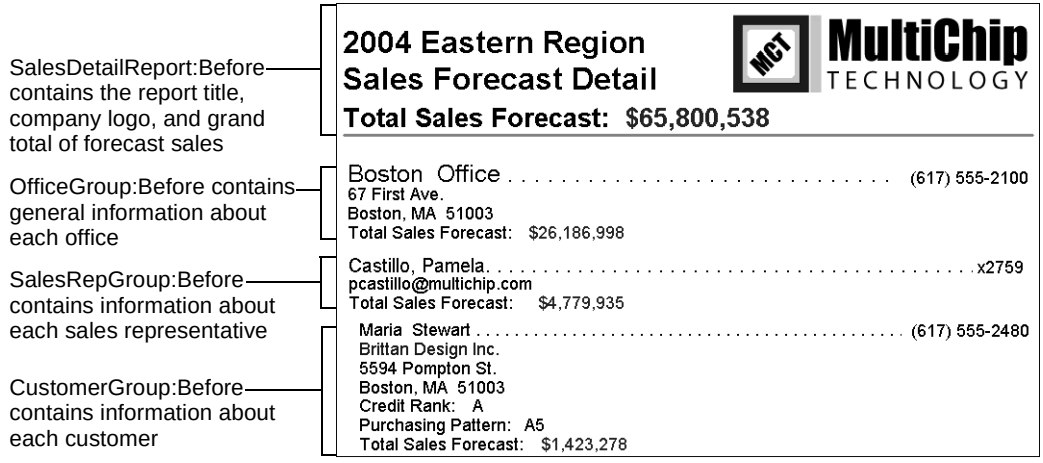


Figure 4-19 First page of the report, Detail.roi

Placing data in a report

To place data in a slot, you use a frame and controls. A control is a component that displays data, graphics, charts, or other information. Controls are the visible building blocks in a report. When you view a report, controls that you place in the report design display all the information that you see.

There are two general types of controls, data controls and static controls. Data controls typically display data from database fields. For example, you use a text control to display a data string, an integer control to display numbers, a date and time control to display dates and times, and a chart control to present data in a chart. Static controls display elements that do not come from database fields, such as lines, images, and text labels. Use static controls to enhance the appearance of a report or identify the data.

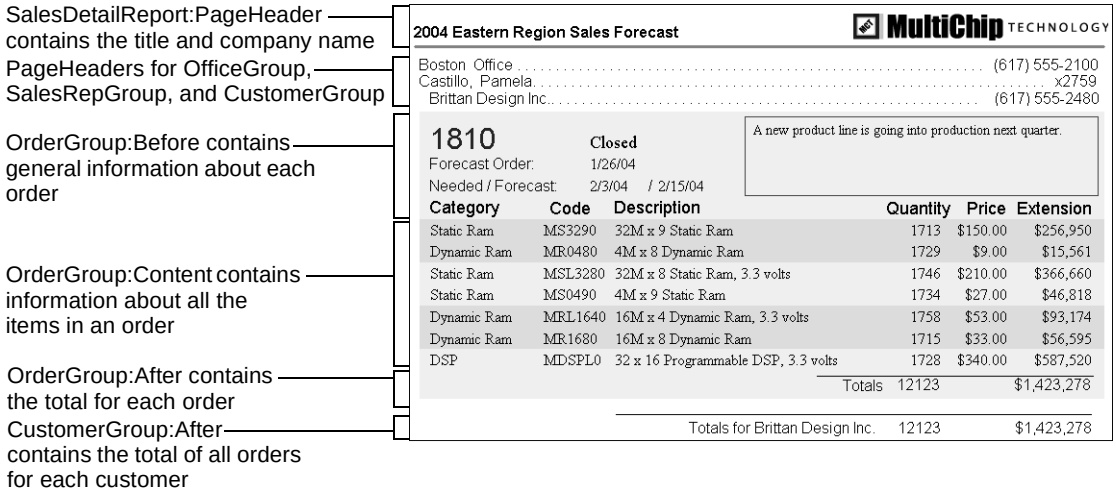


Figure 4-20 Second page of the report, Detail.roi

A frame is a rectangular container that groups related controls. If you use data controls to display data from database fields, you must place them in a frame. Figure 4-21 shows a report design with a frame that contains eleven controls.

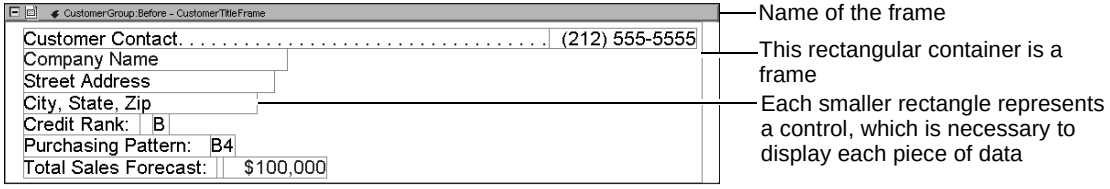


Figure 4-21 Part of a report design showing a frame that contains controls

Figure 4-22 shows how the data appears in the finished report.

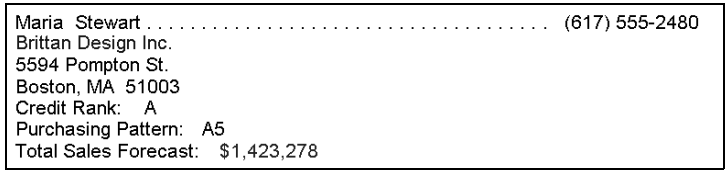


Figure 4-22 The same design displaying data as it appears in the finished report

When you use Tools➛Grouping and Sorting to create groups, e.Report Designer Professional automatically creates the group sections. It also creates frames in the Before and Content slots of the innermost group because grouped reports almost always contain data in these slots.

If you want to add data to the PageHeader or PageFooter slot, place a frame in the slot first. To do so, drag a frame from Toolbox—Structure and drop it in the slot.

Once you place a frame, you are ready to place data in it. The following sections describe how you place different types of data in the report.

Placing database fields

Most of the data that you place in a report are database fields. The report displays this data exactly as the database stores it. For example, in a customer orders report, you place customer name, order number, item, quantity, and price fields in the report.

Typically, you place database fields in a frame in the Content slot but you also can place database fields in other slots. For example, to use a customer name as a group heading, you place it in a frame in the customer group's Before slot. To repeat a customer name at the top of each page, you place it in the customer group's PageHeader slot.

You can place database fields in three ways:

- Use Insert➤Database Field.
- Select fields from Fields.
- Drag a control from Toolbox—Controls, then specify the database field for the ValueExp property.

Using the first and second options, you choose a field and e.Report Designer Professional creates a control that is most appropriate for the field's data type. For example, e.Report Designer Professional creates a text control for a field of string type, or an integer control for a field of integer type. The third option requires you to choose both the control and the field.

The following instructions describe how to place database fields in a report, using the CustomerOrders report that you created in the previous section as an example.

How to place database fields in a report

1 Choose View➤Fields to display the list of database fields as shown in Figure 4-23.

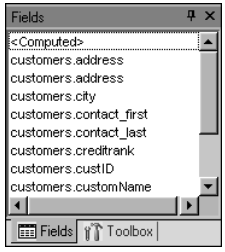


Figure 4-23 List of database fields

2 Drag a field from Fields and drop it in a frame. For the CustomerOrders example report, place items.itemcode in the Content frame in OrdersOrderIDGroup. A control displays the [<table.column>] name or Sample Value, depending on your Display Sample Data setting in the Options dialog.

3 Repeat step 2 to insert the rest of the database fields in a frame. For the CustomerOrders example report, insert items.quantity and items.pricequote to the right of the items.itemcode field.

Figure 4-24 shows how the report design should look.

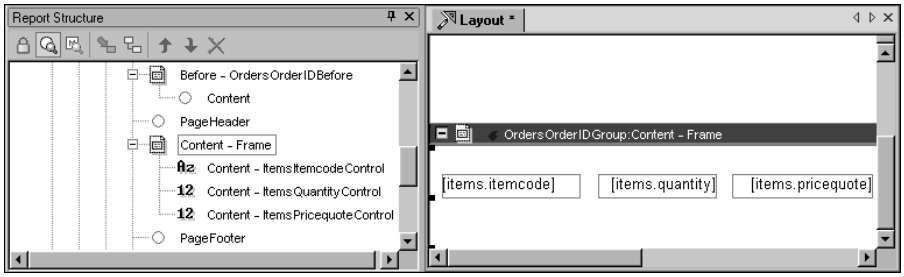


Figure 4-24 Report design layout

4 To see what the report displays, run the report by choosing Report➛Build and Run.

Figure 4-25 shows the first page of the CustomerOrders example report. The report displays rows of data that correspond to the database fields that you placed. The white space beneath the title represents the frame that contains no data. Later, you place data in this frame. Notice that each page displays Report Title, the page number, and the current date. When you create a new report with the Blank Report option, e.Report Designer Professional includes these report elements, which are defined in the Page component.

5 Choose View➛Design to return to the report design.

Placing computed fields

A computed, or calculated, field displays the result of an expression rather than stored data. For example, the database stores the price of an order item and the quantity ordered. To display the extended price, you insert a computed field. The expression for this computed field is:

```
[items.quantity] * [items.pricequote]
```

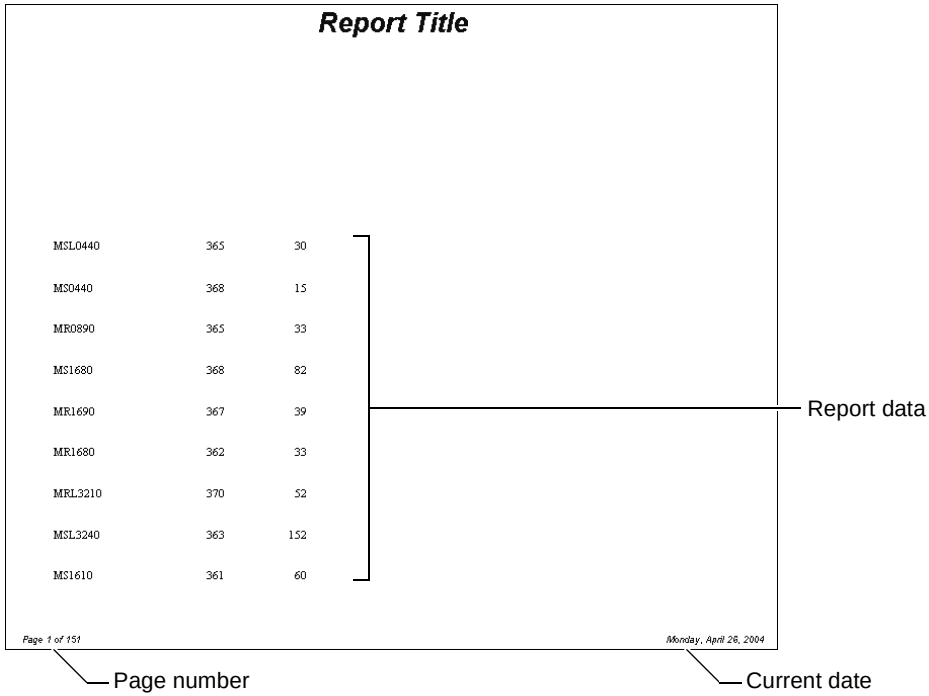


Figure 4-25 First page of CustomOrder examples report

Table 4-2 shows examples of when to use computed fields and the types of expressions that you can specify.

Table 4-2 Computed fields and types of expressions

Uses for computed fields	Examples of expressions

Display data from several fields	`[customer.fname] & " " & [customer.lname]`
	This expression displays a customer's first and last names, which the database stores separately.
	`[customer.limit] - [customer.balance]`
	This expression calculates a customer's available credit.

Display text that includes values from a database field	`"Order " & [orders.orderID]` This expression displays, for example, Order 4123.

(continues)

Table 4-2 Computed fields and types of expressions (continued)

Uses for computed fields	Examples of expressions

Display information using an Actuate Basic function, Actuate Foundation Class (AFC) method, or user-defined method	`Date$( )`
	This Actuate Basic function displays the current date.
	`Page No$( )`
	This AFC method displays the page number.
	`ComputeDiscount([customer.id], [order.total_amount])`
	This user-defined method calculates a discount.

Format data	`Format([items.pricequote], "Currency")`
	This expression displays numeric data as a currency value.
	`Format$([orders.forecastShipDate], "Long date")`
	This expression displays dates in long date format, such as Wednesday, March 17, 2004.
	`Format$([customers.customName], ">")`
	This expression displays customer names in all uppercase.

Parse data	`Format$([orders.shipByDate], "mmmm")`
	This expression takes a date, such as 01/19/04, and returns the name of the month, January.
	`Year([orders.shipByDate])`
	This expression takes a date and returns the year.

How to place a computed field in a report

The following instructions describe how to place a computed field in a report, using the CustomerOrders example report that you created earlier. For more information about the CustomerOrders example report, see "How to create the CustomerOrders example report and build the query," earlier in this chapter.

1. Choose Insert➛Computed Field.
2. Click the frame in which you want to insert the computed field. For the CustomerOrders example report, click the Content frame in OrdersOrderIDGroup and place the computed field next to the items.pricequote field. Expression Builder appears.
3. Type an expression. You can use choose elements from Expression Builder to construct the expression. For the CustomerOrders example report, type the following expression as shown in Figure 4-26 to calculate the extended price for the order item:

   ```
   [items.quantity]*[items.pricequote]
   ```

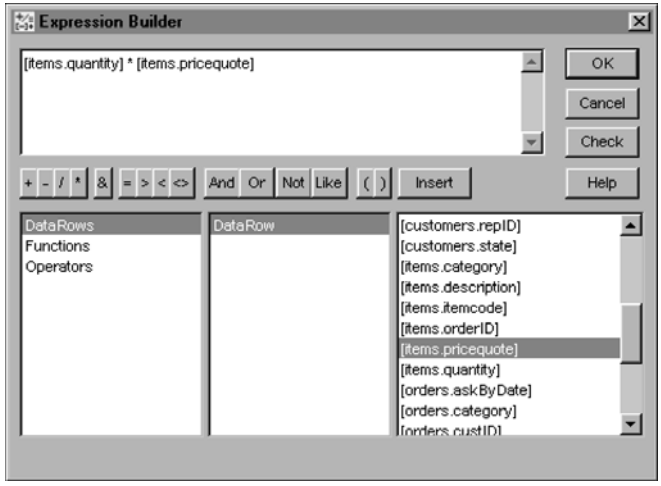


Figure 4-26 Expression Builder

4 Choose OK. Choose Control Type appears.

5 Select a control type that matches the data type of the computed field's return value. For example, if the computed field returns a string value, select the text control. If the computed field returns a date value, select the Date and Time control.

   For the CustomerOrders example report, select the Currency control to display the computed extended price. Choose OK.

6 To check that the report displays the correct values for the computed field, run and view the report by choosing Report➛Build and Run.

   For more information, if e.Report Designer Professional displays error messages, or the computed values are not what you expect, see "Verifying values of computed fields," later in this chapter.

   Figure 4-27 shows the first page of the CustomerOrders example report.

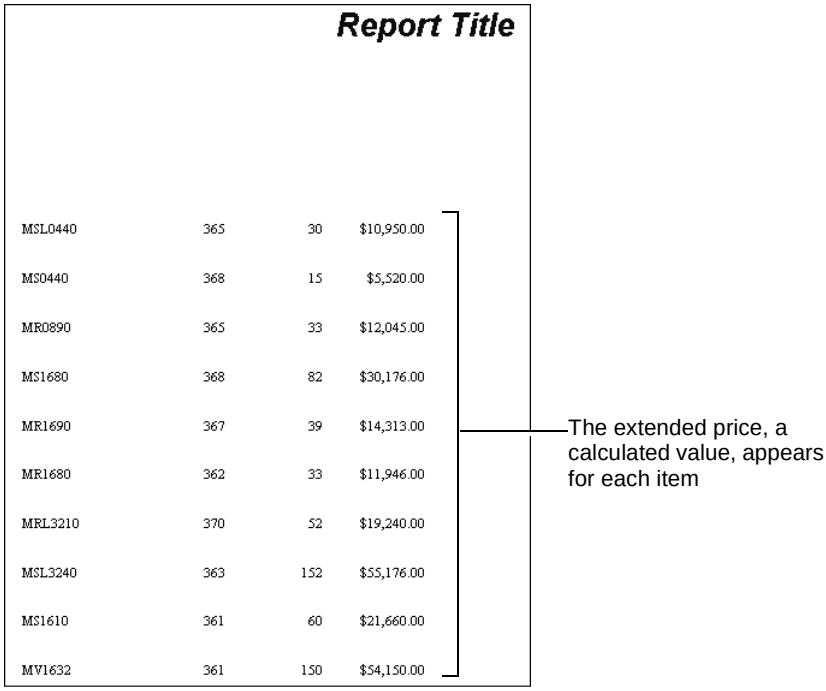


Figure 4-27 Customer order example report

Associating data with controls

When you place a field in the report using Insert➤Database Field or View➤Fields, e.Report Designer Professional automatically creates a control that is appropriate to display the data type of the database field. When you place a computed field using Insert➤Computed Field, e.Report Designer Professional prompts you to select the type of control suitable for displaying the computed value.

After you place a database or computed field on the report, e.Report Designer Professional binds, or associates, the field value with the control through the control's ValueExp property. For example, if you place the database field, items.itemcode, e.Report Designer Professional sets the control's ValueExp value to [items.itemcode]. Similarly, if you place a computed field that calculates an item's extended price, the control's ValueExp value is [items.quantity]*[items.pricequote].

If you place a control in a report by dragging it from Toolbox—Controls, you must set the value of ValueExp yourself as shown in Figure 4-28. Otherwise, the control does not display data.

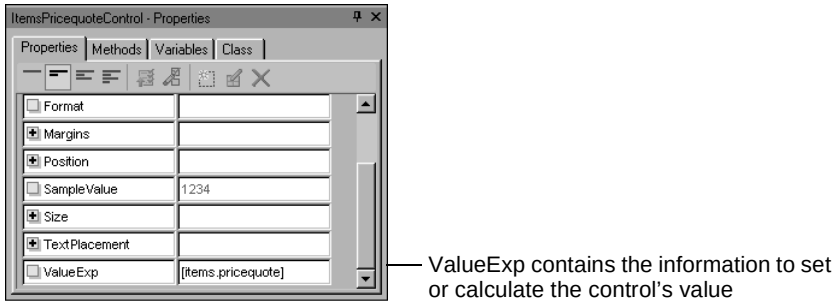


Figure 4-28 Setting ValueExp when using Toolbox—Controls

After you place a database field or computed field in the report, you can:

- Change the control's value by changing its ValueExp value.
- Change the control type, for example, change an integer control to a currency control.

Changing a control's value

You change a control's ValueExp value to display different data. The following situations are typical cases for changing a control's ValueExp value after placing data in a report:

- You inadvertently place the customers.city field twice instead of placing customers.city and customers.state. To correct this mistake, change ValueExp from [customers.city] to [customers.state].
- You place a database field and format the data and size the control the way that you want it. To save time on formatting each control the same way, you copy and paste the control multiple times to create controls with the same format. Then, you change the ValueExp of each control to display data from different fields.

Changing a control's class

You change a control class to display data in a different way. The following are some typical cases for changing a control class after placing data in a report:

- The database stores prices or sales amounts as integers. When you use Insert➤Database Field or View➤Fields to place the data, e.Report Designer Professional creates an integer control to match the field's data type. Although you can use the format option to display a currency symbol in front of the number, you prefer to use a currency control instead.
- You use a label control to display literal text, such as Total. Later, you decide to change the label to include a database value, such as Total For Order xxxx, where xxxx is a value from the order ID field. To change what the label displays, take the following steps:

- Because a label control can only display literal text, you need to change the label control to a text control.
- Set the text control's ValueExp property to the following value:

```
"Total For Order " & [orders.orderID]
```

Although you can delete a control and re-insert the database or computed field, it is easier to change the control's class, especially if you have already positioned and formatted the control the way that you want it.

How to change the class of a control

1. Right-click the control whose class you want to change, then choose Properties. The Properties page for the control appears.
2. Choose Class. The Class page appears as shown in Figure 4-29.

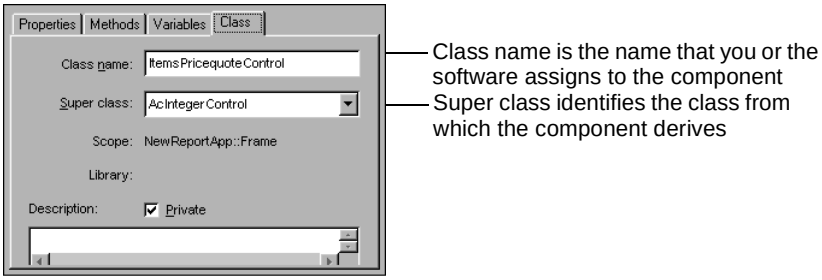


Figure 4-29 Class page

3. In Super class, change the control's super class name. For example, to change an integer control to a currency control, change AcIntegerControl to AcCurrencyControl.

   For more information about classes, see *Programming with Actuate Foundation Classes*.

Adding column headings

Column headings are static labels that appear above data columns. You typically add column headings to describe data fields if a report displays detail data in a columnar layout. The CustomerOrders example report, for example, displays all the items in an order, and each data row contains the item code, quantity ordered, price per unit, and extended price. With column headings, shown in Figure 4-30, the report user knows what the numbers in the data rows represent.

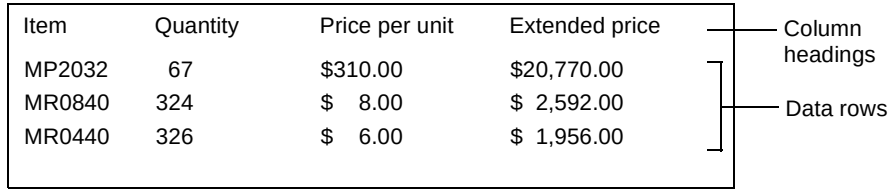


Figure 4-30 Column headings

To create column headings, you use literal text, such as Item and Quantity. For column headings to appear above the data fields, you place the column headings in the Before slot of the group that contains the data fields. Text or data that you place in a Before slot appears only once, before the group's data rows begin. If the group's data exceeds one page, the column headings for the group do not repeat at the top of the subsequent pages. To repeat group column headings on subsequent pages, you must also place the column headings in the group's PageHeader slot.

How to add column headings

The following instructions describe how to add column headings to a report, using the CustomerOrders example report that you created earlier:

1. Choose Insert➛Label and click in the frame where you want to place the column heading. For the CustomerOrders example report, place the label in the Before frame in OrdersOrderIDGroup. A label control appears in the frame.
2. Change the default text, Label, to the column heading that you want to display.
    1. Right-click the label control, then choose Properties. The Properties page for the label control appears.
    2. Type a new value for the Text property.
3. Repeat steps 1 through 2 to create all the column headings for the selected frame. For the CustomerOrders example report, create these column headings:
    - Item
    - Quantity
    - Price per unit
    - Extended price
4. Choose Report➛Build and Run. The report generates and appears in CustomerOrders.roi.

    Figure 4-31 shows the first and second pages of the CustomerOrders example report. The column headings appear at the top of the first page but not at the

top of the second page, although the order data continues on the second page. Column headings appear again when a new order begins.

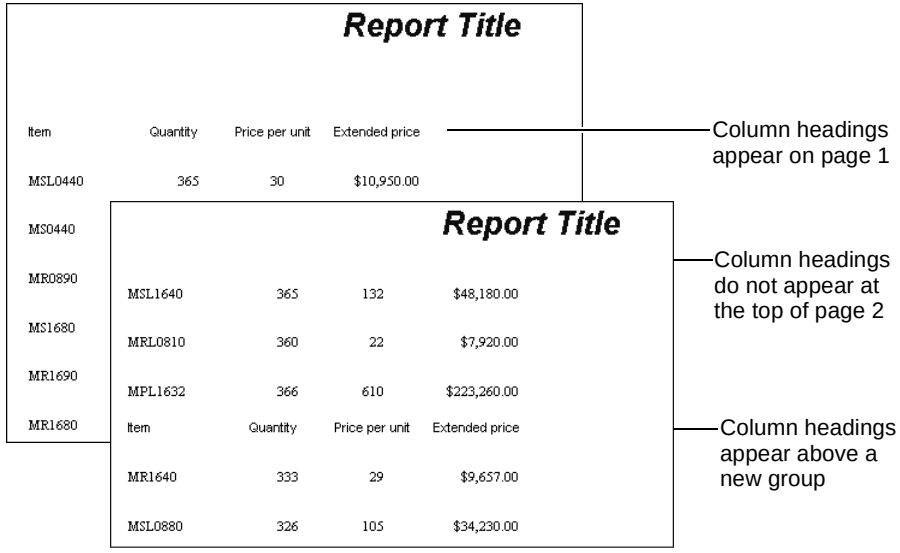


Figure 4-31 First two pages of CustomerOrders example

5 To display the column headings on subsequent pages when the data goes beyond one page, take the following steps:

1. Create a frame in the PageHeader slot of the OrdersOrderIDGroup.
2. Place the same column headings in the new frame.

Figure 4-32 shows the column headings in the report design.

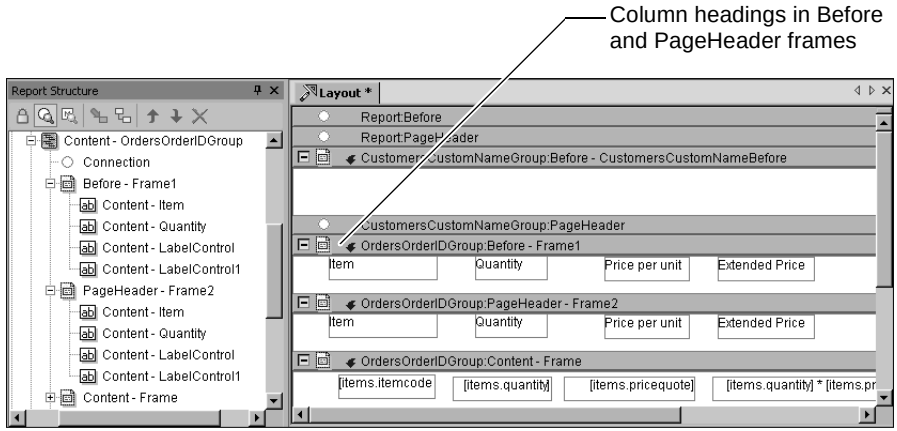


Figure 4-32 Column headings within a report design

6 Run and view the report by choosing Report➤Build and Run. Figure 4-33 shows the first and second pages of the CustomerOrders example report. Now, the column headings appear at the top of every page that displays data for this order and at the beginning of each group of orders.

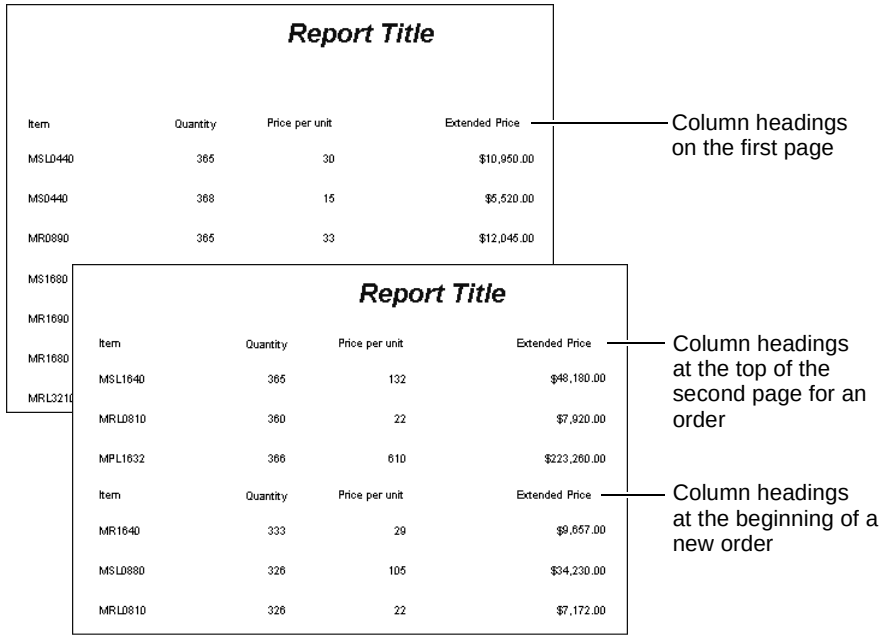


Figure 4-33 Column headings in a generated report

Adding group headings

If your report contains groups, you typically add a heading that appears at the beginning of each group to identify the data within the group. Unlike column headings, which are static, group headings change based on the content of the group.

For data that is grouped by customers, for example, you can add the customer name at the beginning of each group. For a customer list that groups names by the first letter, for example, the title preceding each group is A, B, C, and so on. For a shipping schedule that groups data by weeks, the title preceding each group is Week of 03/01/04, Week of 03/08/04, and so on.

Figure 4-34 shows examples of headings that change with each group. Group headings are highlighted in bold text.

If you do not use group headings, it can be unclear where one group ends and another begins. Remove the A and B headings from the Customers report, for example, and the report looks like a simple list sorted alphabetically. You can add

a line or space between groups to indicate a change in group but it is better to display a descriptive heading.

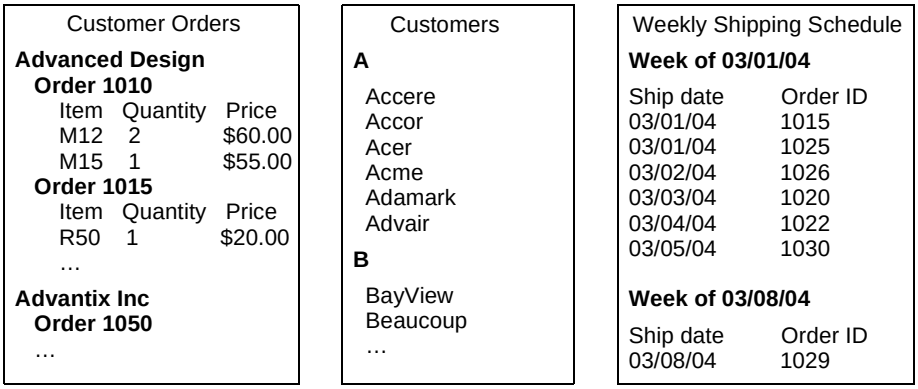


Figure 4-34 Report examples of headings

Placing group headings

To create a group heading, you can use a database field or a computed field, depending on the value that you want to display. For a heading to appear at the beginning of each group, you place the database or computed field in the Before slot of the group. Text or data that you place in a Before slot appears only once, before the group data begins. If the data in the group exceeds one page, the group headings are not repeated on the subsequent pages.

To repeat group headings on subsequent pages, you must place the group headings in the group's PageHeader slot as well.

Specifying expressions for group headings

Unlike column headings, which you create using static text, you specify expressions for group headings because the heading values change based on the group's content. You can use a database field or a computed field, depending on the value that you want to display. Typically, you use a computed field to combine a field value and literal text. Sometimes, you use an Actuate Basic function to extract the values that you need. The following examples show how to use a database field, a computed field, and an Actuate Basic function to create group headings.

To display the customer name as the group heading for the CustomerOrders example report, use the customer field as the expression:

```
[customers.customName]
```

To display the order number group heading for the CustomerOrders example report, combine literal text with the order ID field, as follows:

```
"Order " & [orders.orderID]
```

To create headings, A, B, and so on, for a customer list grouped alphabetically, use the Actuate Basic String$( ) function to get the first letter of the name in each group, as follows:

```
String$(1, [customers.customName])
```

How to add a group heading

The following instructions describe how to add a group heading to a report, using the CustomerOrders example report that you created earlier:

1. Depending on the value that you want to display as the group heading, insert either a database field or a computed field in the group's Before frame.
2. Take the following steps to add group headings to the CustomerOrders example report:
   1. Choose Insert➛Database Field, then drop the component in the Before—CustomersCustomNameBefore frame. Choose Field appears.
   2. Select [customers.customName]. Choose OK. A text control appears in the CustomersCustomNameBefore frame.
   3. Choose Insert➛Computed Field, then drop the component above the column headings in the Before—OrdersOrderIDBefore frame. Expression Builder appears.
   4. Type the following expression:

      ```
      "Order " & [orders.orderID]
      ```

      Choose OK. Choose Control Type appears.
   5. Select Text as the control type. Choose OK. A text control appears in the Before—OrdersOrderIDBefore frame. You can expand the control to display all the text. The report design looks like the one in Figure 4-35.

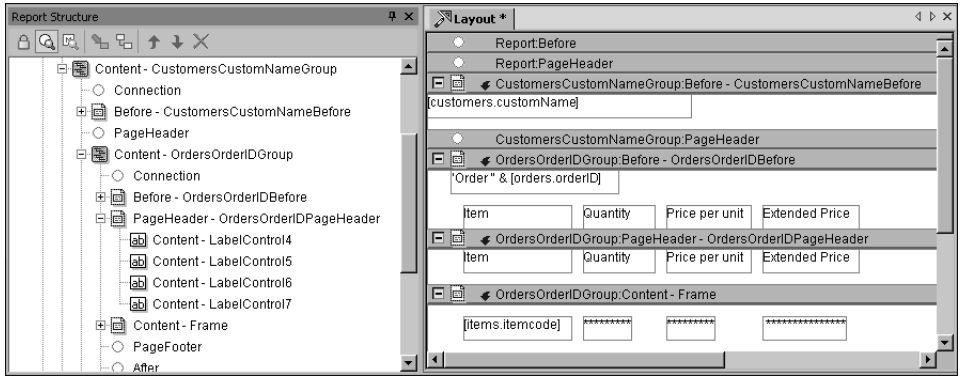


Figure 4-35 Expand text control to display all text

3 Choose Report➛Build and Run. Figure 4-36 shows the first and second pages of the CustomerOrders example report.

The [customer.customName] and "Order" & [orders.orderID] values appear as group headings only on the first page, although the customer and order data continue on the second page.

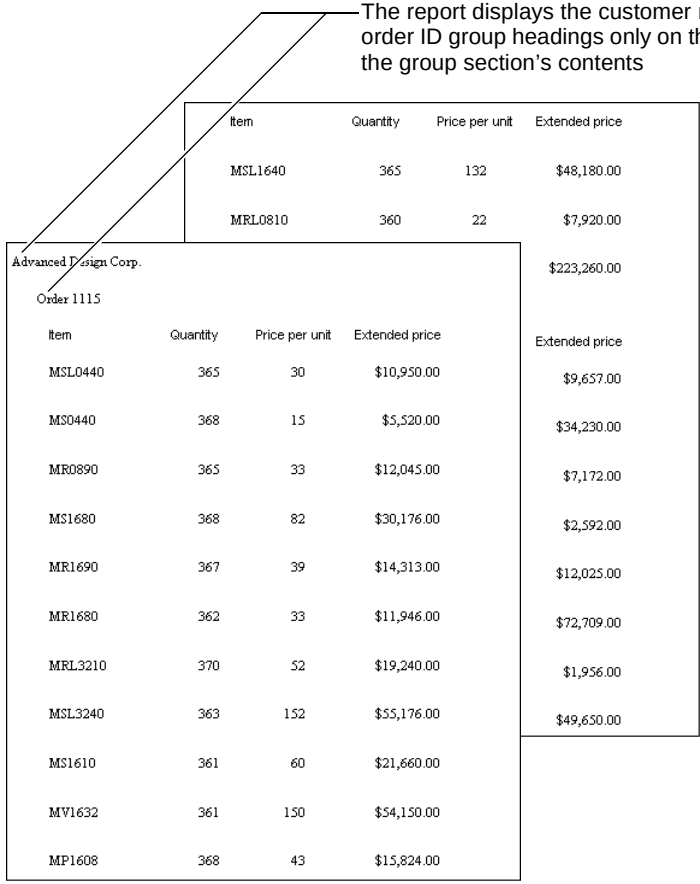


Figure 4-36 Example of customer name and order ID headings

4 To display the customer and order ID group headings on subsequent pages when the data extends beyond one page:

1 Create a frame in the PageHeader slot of each group.

2 Place the same group headings in each of the new frames. Expand the headings to display all the text.

Figure 4-37 shows the group headings as they should appear in the report design.

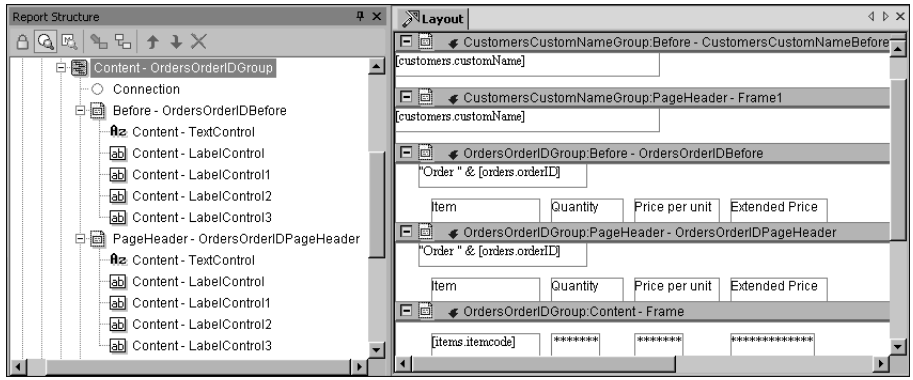


Figure 4-37 Displaying group headings to appear on multiple pages of a report

5 Run and view the report by choosing Report➤Build and Run. Figure 4-38 shows the first and second pages of the CustomerOrders example report. Now, the group headings appear on both pages.

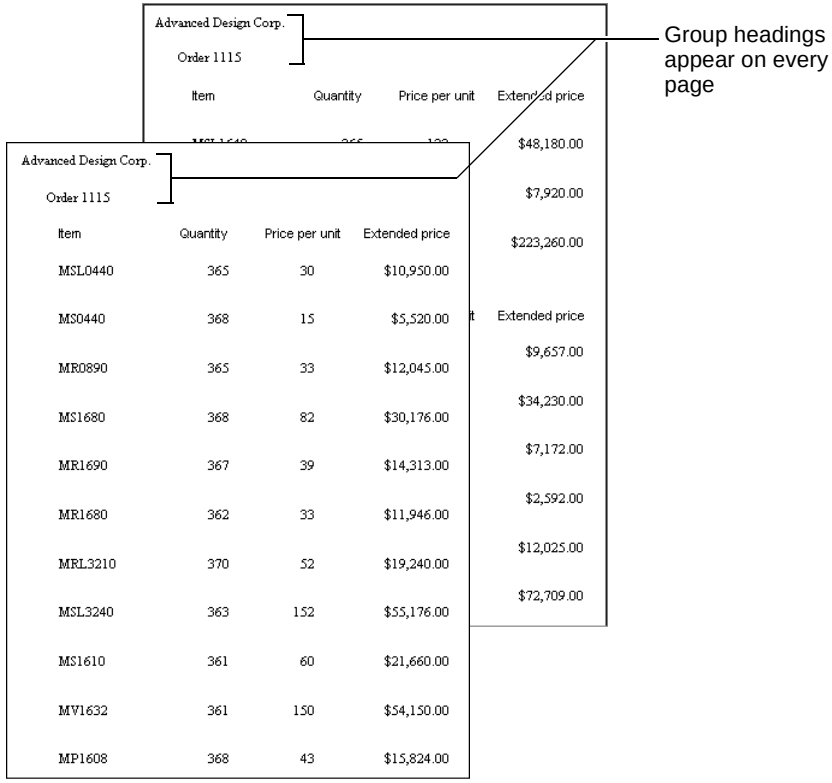


Figure 4-38 Group headings on each page of a report

Adding aggregate information

One of the primary purposes for grouping data is to calculate and display aggregate, or summary, information for each group of records. For example, when you group order data by sales representative, you can display the highest order total in each sales representative group. You can also display aggregate data for an entire report. For example, a monthly sales report typically displays data such as total sales for the month. Aggregate functions do not include null values in their calculations.

Table 4-3 lists the calculations that you can perform, depending on the data in a group or a report.

Table 4-3 Calculation types for data in a group or report

Type of calculation	Aggregate function
Sum the values of a specified numeric field	Sum( )
Count the number of data rows	Count( )
Get the average value of a specified field	Ave( )
Get the highest value of a specified field	Max( )
Get the lowest value of a specified field	Min( )

You can use one or more aggregate functions in an expression. You can also use a clause, such as GROUP BY, to refine an aggregate function.

You can place aggregate information only in the following places in a report:

- At the beginning of a group or a report, in the Before slot
- At the end of a group or a report, in the After slot

Examining sample aggregates

Figure 4-39 shows a report that displays totals for each order, totals for each customer, and a grand total of all sales.

The report uses the following Sum( ) expression to calculate the cumulative totals for the groups and the report:

```
Sum([items.quantity] * [items.pricequote])
```

To display the totals as shown in the report, take the following steps:

- To display the grand total at the beginning of the report, place the aggregate data in the Before slot of the report.
- To display the customer total at the beginning of each customer group, place the aggregate data in the Before slot of the customer group.

- To display the order total at the end of each order group, place the aggregate data in the After slot of the order group.

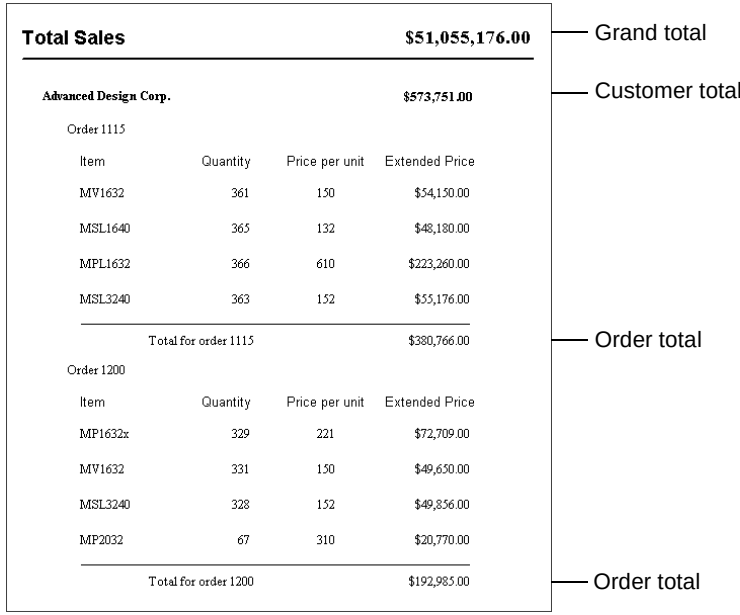


Figure 4-39 Report showing aggregates of various groups

How to add aggregate information

1 In Report Structure or Layout, select the Before or After frame of the group in which to place the aggregate information. In the CustomerOrders example report, insert a frame in the After slot in Content—OrdersOrderIDGroup, as shown in Figure 4-40.

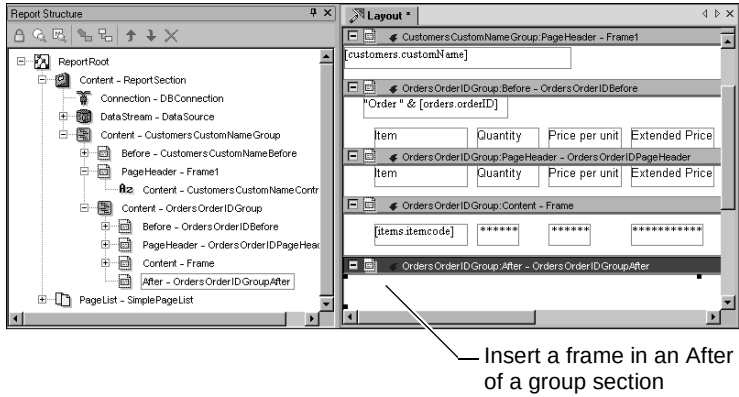


Figure 4-40 Insert a frame to add aggregate information

2 Choose Insert➤Total.

3 Click the frame where you want to insert the aggregate information. Total appears.

4 Specify the following information:

   1 Select the database field on which to perform an aggregate calculation. For the CustomerOrders example report, select items.pricequote from the drop-down list.

     Notice that you can select only database fields, not computed fields. To perform an aggregate calculation on a computed field, such as a field that computes the extended price, items.quantity * items.pricequote, of an order item, you modify the control's ValueExp property later in this procedure.

   2 Select the aggregate function to use. For the CustomerOrders example report, select Sum from the drop-down list. Choose OK. Choose Control Type appears.

   3 Select an appropriate control type from the drop-down list. For the CustomerOrders example report, select Currency from the drop-down list. Choose OK. e.Report Designer Professional creates a control in the selected frame.

5 To perform an aggregate calculation on a computed field, complete the following steps:

   1 Right-click the control, then choose Properties. The Properties page for the control appears as shown in Figure 4-41.

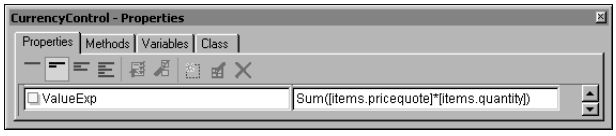


Figure 4-41 Setting up reports to show aggregate calculations

   2 Change the control's ValueExp property. For the CustomerOrders example report, e.Report Designer Professional set the ValueExp property to Sum([items.pricequote]) based on what you did in step 4. To calculate the total amount for each order, change ValueExp to:

```
Sum([items.pricequote] * [items.quantity])
```

6 Run the report by choosing Report➤Build and Run. The report appears in the view perspective.

Figure 4-42 shows the aggregate information in the CustomerOrders example report for a group section that spans more than one page.

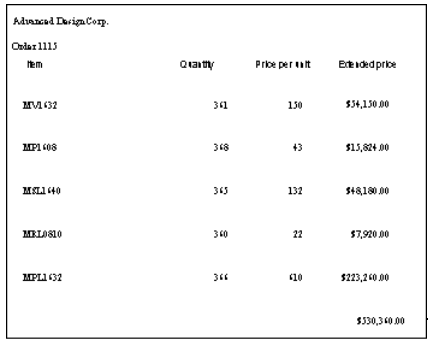


Figure 4-42 Aggregate total for a multipage report

7 Choose View➤Design. The report design appears.

8 Identify and highlight the aggregate data in the report. For example, add a text control and a line control.

In the CustomerOrders example report:

- Insert a text control and set its ValueExp property to display text that describes the aggregate data:

```
"Total for order " & [orders.orderID]
```

- Set the text control's TextPlacement➤Horizontal property to TextAlignRight.
- Insert a line control to separate the report content in the After frame from the content in the preceding frame as shown in the report design in Figure 4-43.

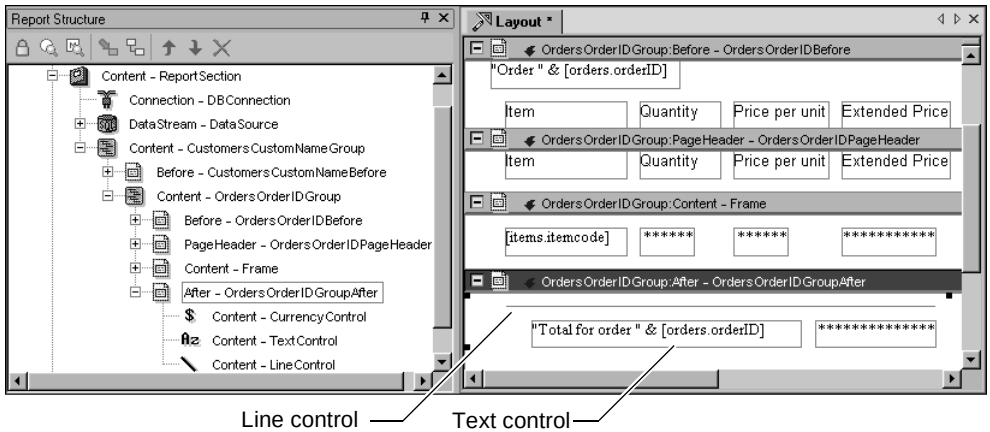


Figure 4-43 Formatting a report to present aggregate information

9 Run the report by choosing Report➤Build and Run. The report appears in the view perspective.

Figure 4-44 shows the aggregate information in the CustomerOrders example report for a group section that spans more than one page.

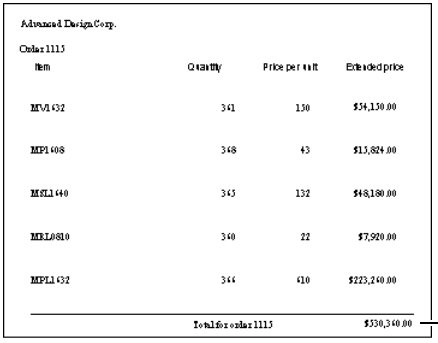


Figure 4-44 Generated and formatted report result

Refining aggregate functions

The example report in "Examining sample aggregates," earlier in this chapter, uses an aggregate function, sum, with an extended price expression, [items.quantity] * [items.pricequote]. To create the group section totals, e.Report Designer Professional sums all the extended prices in each group. To create the grand total, e.Report Designer Professional sums all the extended prices in the report.

To display more complicated aggregate information, you can refine an aggregate function. For example, you can display a group section total as a percentage of the report total. To refine an aggregate function, you can add a clause to the aggregate expression. A clause can perform the following tasks:

- To filter the data that the aggregate function summarizes, use the WHERE clause. For example, you can add values to a sum total only when a particular condition is true.
- To filter all duplicate values from an aggregate calculation, use the DISTINCT clause. For example, in a listing report that repeats company names in data rows, you can use DISTINCT to count distinct company names. You cannot use DISTINCT in a lookahead aggregate.
- To compare a value to a report total, use the OVERALL clause. To compare a value to a group total, use the GROUP BY clause. These types of calculations are called lookahead aggregates, because e.Report Designer Professional must calculate the report total or group total before calculating the aggregate value for an individual row or a group section. For example, you can show a group section total as a percentage of the report total.

The following sections describe how to use WHERE, OVERALL, GROUP BY, and DISTINCT.

Using WHERE to filter aggregate data

The WHERE clause enables you to filter the data that an aggregate calculation returns. You can use the WHERE clause with other aggregate clauses, such as GROUP BY and DISTINCT. Use the following syntax:

```
WHERE (Condition)
```

where condition is an expression such as a comparison or an aggregate function.

For example, the following expression calculates the average transaction amount for sales transactions:

```
AVE([transaction.amount]) WHERE([transaction.type]="Sales")
```

Using DISTINCT to filter aggregate data

The DISTINCT clause enables you to filter duplicate values from an aggregate calculation. Use the following syntax:

```
DISTINCT(Expression)
```

where Expression is the field name or expression on which to filter. e.Report Designer Professional includes data in an aggregate calculation only if the Expression value is different from the previous row's Expression value.

For example, the following expression uses DISTINCT to count unique company names:

```
COUNT([Customer.Name]) DISTINCT([Customer.Name])
```

You cannot use DISTINCT with OVERALL or GROUP BY. To correctly calculate unique rows, ensure that you sort a report on the same expression that you use in the DISTINCT clause. For example, the following report shown in Figure 4-45 sorts data by state, then by order number.

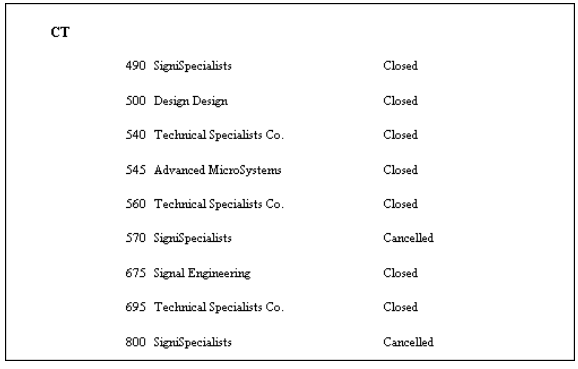


Figure 4-45 Using DISTINCT clause to sort unique rows

If you use DISTINCT in an aggregate expression that calculates the total number of customers in each state section, the aggregate returns inaccurate data. For example, the aggregate function counts three instances of SigniSpecialists in CT because the three instances do not appear in sequential rows in the report. To correctly calculate the number of customers in CT, you must first sort the detail rows by customer, as shown in Figure 4-46.

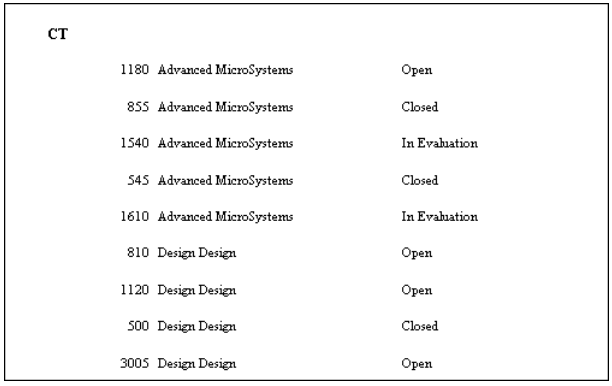


Figure 4-46 Report sorting rows by customer

Because all the instances of one customer name appear sequentially in the customer name column, DISTINCT can eliminate duplicate instances of each name and count only unique customer names.

Using a lookahead aggregate in a conditional section or sequential section

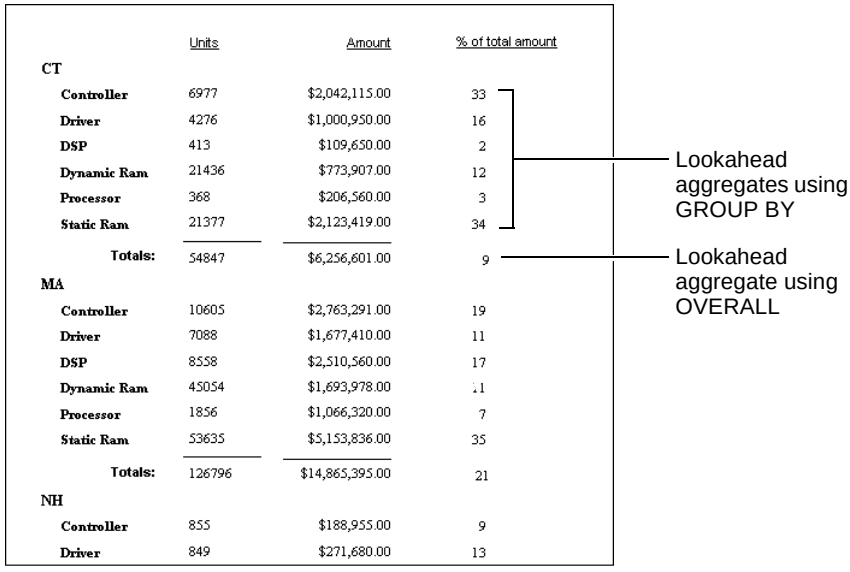
Actuate e.Report Designer Professional does not support using an aggregate function in a conditional section or a sequential section. If you use an aggregate function in a conditional section or a sequential section, the aggregate function retrieves only the data value from the first data row. You can use a lookahead aggregate in a control that is in a sequential section or in a conditional section.

About lookahead aggregates

The OVERALL and GROUP BY clauses enable you to create lookahead aggregates. A lookahead aggregate is a calculation that uses data that is not available until e.Report Designer Professional processes an entire group or report. For example, the report shown in Figure 4-47 uses two lookahead aggregates.

The first aggregate uses GROUP BY to compare a product category total to a state group total. Within each group section, the aggregate calculates the percentage of a group total each product category represents. For example, in the report in the illustration, the Controller sales amount represents 33% of the CT sales amount.

The second aggregate uses OVERALL to compare a state group total to the report total. The aggregate calculates the percentage of the report total each state group total represents. For example, the report shows that the CT sales amount represents 9% of the total sales amount.



	Units	Amount	% of total amount
CT			
Controller	6977	$2,042,115.00	33
Driver	4276	$1,000,950.00	16
DSP	413	$109,650.00	2
Dynamic Ram	21436	$773,907.00	12
Processor	368	$206,560.00	3
Static Ram	21377	$2,123,419.00	34
Totals:	54847	$6,256,601.00	9
MA			
Controller	10605	$2,763,291.00	19
Driver	7088	$1,677,410.00	11
DSP	8558	$2,510,560.00	17
Dynamic Ram	45054	$1,693,978.00	11
Processor	1856	$1,066,320.00	7
Static Ram	53635	$5,153,836.00	35
Totals:	126796	$14,865,395.00	21
NH			
Controller	855	$188,955.00	9
Driver	849	$271,680.00	13

Figure 4-47 Report with two lookahead aggregates

Using GROUP BY to create a lookahead aggregate

You use GROUP BY to create an aggregate that compares a row total to a group section total. Append a GROUP BY expression to an aggregate function expression using the following syntax:

```
GROUP BY(Group key)
```

where Group key is the field on which the section is grouped.

For example, the following example calculates the number of orders for a customer:

```
COUNT([order.orderID]) GROUP BY([customers.ID])
```

You can use GROUP BY with WHERE. You can also use GROUP BY in an expression with other expressions. For example, in the report in Figure 4-47, the GROUP BY aggregate uses the following expression series:

```
SUM([items.pricequote]*[items.quantity])/
   SUM([items.pricequote]*[items.quantity]) GROUP BY
   ([customers.state]) * 100
```

The first sum expression calculates the extended price total for each product category row. The second sum expression calculates the sum of the extended

price for each state section. To calculate the percentage of state sales that each product category row represents, the expression divides the row total by the state total and multiplies the result by 100.

Using OVERALL to create a lookahead aggregate

You use OVERALL to create an aggregate that compares a row or group section total to a report total. Append OVERALL to an aggregate function expression. For example, the following example calculates the sum of employee salaries in a report:

```
SUM([employee.salary]) OVERALL
```

You can use OVERALL with WHERE or in an expression with other aggregate expressions. For example, in the report in the previous illustration, the OVERALL aggregate uses the following expression:

```
SUM([items.pricequote]*[items.quantity])/
   SUM([items.pricequote]*[items.quantity]) OVERALL*100
```

The first sum expression calculates the extended price total for each product category row. The second sum expression calculates the extended price total for the entire report. To calculate the percentage of total sales that each state total represents, the expression divides the state total by the report total and multiplies the result by 100.

Inserting a page break before or after each group

e.Report Designer Professional's default behavior is to display a report's contents continuously on every page. You can, however, specify page breaks so that each group starts on a new page. For example, in a customer orders report that groups orders by customer, you can start each customer on a new page.

You use these two properties to specify where page breaks occur within a group:

- PageBreakAfter. If you set this property to True, e.Report Designer Professional inserts a page break after a group finishes, including after the last group. Using this approach can result in a page that contains only headers and footers on the page after the last group appears.
- PageBreakBefore. If you set this property True, e.Report Designer Professional inserts a page break before a group begins, including before the first group. Using this approach can result in a near-blank page before the first group appears.

How to insert a page break before or after each group

1. In Report Structure, right-click the group section for which you want to specify page breaks, then choose Properties. The Properties page for the group section appears.

2 Set Pagination➛PageBreakBefore to True to insert a page break before each group.

3 Set Pagination➛PageBreakAfter to True to insert a page break after each group.

Changing the placement of the report title

When you create a new report with the Blank Report option, e.Report Designer Professional automatically includes the following report elements as a convenience:

- Report Title, which appears in the same location at the top of every page
- Page number, which appears in the same location at the bottom left of every page
- Current date, which appears in the same location at the bottom right of every page

These components do not appear in the report layout because they are part of the page layout. Typically, you place data in the report layout and place decorative elements that you want to display on every page in the page layout.

To view and modify a report's default page layout, expand PageList—SimplePageList and PageStyle—Page in Report Structure.

To change the report title location and appearance, complete the following tasks:

- If you do not want the report title to appear in the same format on every page, delete it from the page layout and place it in the report layout instead.
- To display the report title only on the first page, place it in a frame in the report's Before slot.
- To display the report title in one format on the first page and in a different format on subsequent pages, place the title for the first page in the report component's Before slot. For subsequent pages, place the title in the report component's PageHeader slot.

Figure 4-48 shows the report title as it appears in the Detail example report design.

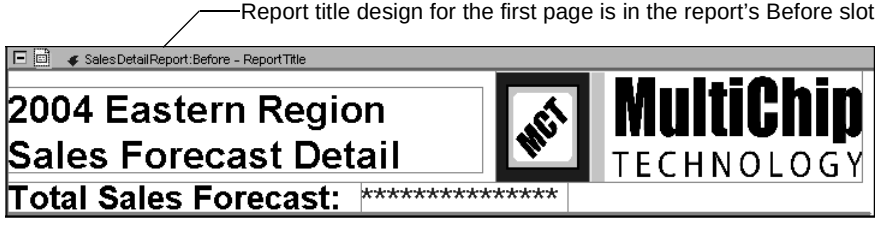


Figure 4-48 Report title design for the first page of report

Figure 4-49 shows the report title as it appears in the Detail example report document. On the first page, the report title spans multiple lines.

Figure 4-49 Report title appearance on first page of the report document

On subsequent pages, the title appears on a single line as shown in the report design in Figure 4-50 and in the report document in Figure 4-51.

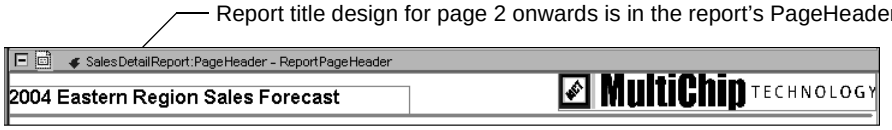


Figure 4-50 Report title design on subsequent pages of report

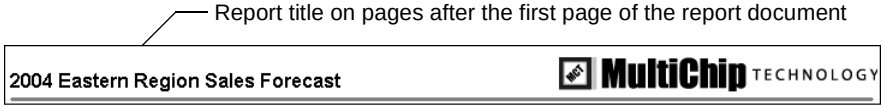


Figure 4-51 Report title appearance on subsequent pages of report

Sorting report data

For a report that contains groups, data sorting occurs at two levels:

- Group level
- Detail, or data row, level

Sorting data at the group level

Grouping data sorts data by the field that you specify. If you create multiple groups, you sort data by multiple fields.

When you create groups, e.Report Designer Professional automatically sorts the data within each group in ascending order. The CustomerOrders example report which this chapter uses as an example contains two groups, customers and orders. The report lists customers in ascending alphabetical order from Advanced Design Corp to TeleSystems Inc. Within each customer group, the report lists orders in ascending numerical order.

How to change the sort order of data at the group level

These instructions assume that you already created groups.

1. Choose Tools➛Grouping. Grouping and Sorting—Grouping displays the groups on the right side.
2. Select the group whose sort order that you want to change.
3. Choose Grouping Options.

   Grouping Options appears.
4. In Grouping Options, for Sort Order, select Ascending or Descending. Choose OK.
5. Choose Report➛Build and Run to check the results of the group sorting. The report appears in the view perspective.

Sorting data at the detail level

Unlike group-level data, data at the detail level, which is data within the innermost group, appears in the order the database sends it to e.Report Designer Professional. The order varies, depending on how the data appears in the database and how you link tables.

You probably want to control the order in which detail information appears in the report. An order item listing, for example, is more useful if sorted alphabetically or by purchase quantity. You can sort by as many fields as you want. For example, you can sort a list of order items by quantity, then by price.

How to sort data at the detail level

1. Choose Tools➛Grouping. Grouping and Sorting—Grouping appears.
2. Choose Sorting. Grouping and Sorting—Sorting appears.
3. In Fields To Sort By, click a blank field. A drop-down list appears. The list contains fields that you did not already use as groups. Here, you sort only data at the detail level.
4. Select a field from the list. For the CustomerOrders example report, select items.quantity as shown in Figure 4-52
5. In Sort Order, specify the sort order by selecting Ascending or Descending.
6. Repeat steps 3–5 for each additional field that you want to sort by. For the CustomerOrders example report, select items.pricequote. The report lists items first by quantity ordered, then by price.
7. To delete a sort field, select the field and choose None from the list.
8. To change the order of fields in the Fields To Sort By list, use the up arrow and down arrow buttons.

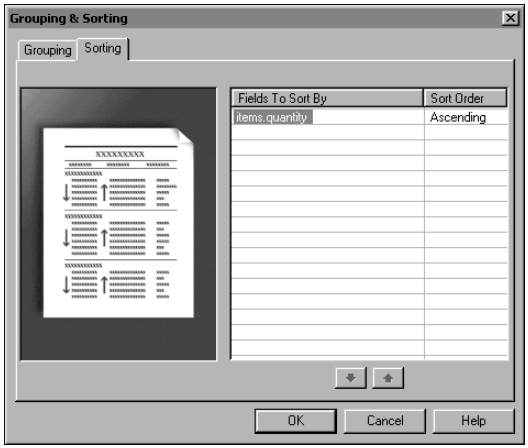


Figure 4-52 Sort data at the detail level

9 Choose OK to close Grouping and Sorting—Sorting.

10 Choose Report➛Build and Run to check the results of the sorting. The report appears in the view perspective.

Enhancing the appearance of a report

After you lay out data in the report, you can fine-tune the appearance of the report by adding images, lines, or text labels. You can use lines to separate parts of a report or underscore the report title, or boxes to highlight information. You can use text labels to identify data.

Remember, to add only literal text, use a label control. Use Insert➛Label to create and add literal text. To add text that combines literal text and data from fields, such as “Number of items for " & [orders.orderID], use Insert➛Computed Field, and select a text control to display the value.

You can enhance the appearance of a report further by formatting the font, colors, and size of controls that display data or information.

If you want to add elements that appear in the same location on every page, such as a copyright line, company name or logo, page number, or report generation date, customize the report’s page layout.

Testing a report design

Unless you have stringent requirements as to how a report must look, designing a report layout is an iterative process. As you create groups, lay out data, and add

decorative elements to the report, you are likely to experiment with layouts to see what looks best. You can change the order of groups or the order of columns in a detailed listing, add and remove headers, add and remove lines and images, and so on. You can start with one layout, try another, then decide on an earlier version. This process can entail creating and re-creating layouts multiple times.

You should save different parts of the report that you might want to use again later. One option is to save multiple versions of the report design file at different stages. Another technique is to store various parts of the report design temporarily on the scratch pad.

Storing report components in the scratch pad

Report components that you store in Scratch Pad remain there—even after you close e.Report Designer Professional—until you delete them or move them into the report design. There is no limit to the number of components that you can store in Scratch Pad. e.Report Designer Professional builds all classes in a report design, including those in Scratch Pad. For this reason, you cannot run a report with a class that does not compile even if you move that class out of the layout or structure windows into Scratch Pad.

If you store several components with the same name, Scratch Pad appends numbers to the names, such as CustomersGroupBefore and CustomersGroupBefore1.

The flexibility of Scratch Pad enables you to create as many versions of report parts as you want and to save them indefinitely. If report users need to review and approve your report design, you can easily create multiple versions of the report so that they can choose the one they prefer.

How to use Scratch Pad

1 Choose View➛Scratch Pad.

2 To store a report component on Scratch Pad, drag the component from Report Structure or Layout and drop it on the Scratch Pad folder icon as shown in Figure 4-53.

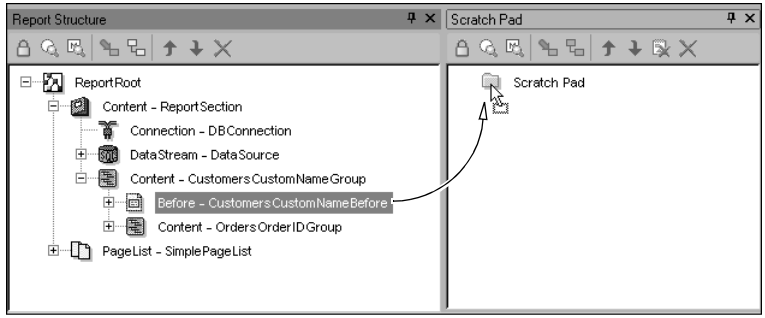


Figure 4-53 Dragging a component onto the Scratch Pad folder

3 To place a report component from Scratch Pad onto your report, drag the component from Scratch Pad and drop it in the appropriate location of the report.

Testing the layout

You should always test your report design as you make layout choices. If you wait until you finish laying out the report before you check the output, and there are errors, it is harder to determine which part of the report causes the problems. When you lay out a report, the earliest that you can check the output is after you place fields on the report. You cannot check the output after you create groups but before placing fields because the report has no data to display.

Aside from previewing the general layout of the report, you should test and verify the data values in your report.

Verifying values of computed fields

Verify that the resulting values of computed fields are what you expect. If you use Actuate Basic or Actuate Foundation Class (AFC) methods in computed fields, e.Report Designer Professional checks for syntax or semantic errors only when you run the report.

If the expression for a computed field contains syntax errors, e.Report Designer Professional displays error messages when you run the report and identifies the component that contains the error. Examine and fix the component's ValueExp property. This property displays the expression that you specified for the computed field.

If a computed field always returns 0, there are no syntax errors in the expression, but there is a semantic error. Examine the component's ValueExp property. Did you use an Actuate Basic function correctly? For example, if you use the Format( ) or Format$( ) function to change the format of a data value, you must use the correct type of control for the function's return value.

For example, the database stores items.pricequote as an integer. When you place this field on the report, you format the value so that it appears in currency format, displaying $100.00 instead of 100. You use the following expression and create an integer control to contain the values:

```
Format([items.pricequote], "Currency")
```

When you run the report, the item price appears as 0. This problem occurs because you used an integer control and the Format( ) function in this case returns a string value. To fix the problem, change the integer control to a text control.

An easier and better solution, however, is to create a currency control in the first place and to use this expression:

```
[items.pricequote]
```

By using the currency control, e.Report Designer Professional automatically formats the integer value as a currency value.

Verifying results of groups and sorts

If you group data by intervals or specify sort criteria for the details section of the report, verify that the data is grouped and sorted the way that you intended.

Checking the running headers and footers

Make sure data or text that you used as group and report headers and footers appear in the correct locations in the report. For example, if you use column headings to appear above data fields in the detail section of a report, verify that the column headings appear on every page of a detail section that spans multiple pages, not just the first page.

For column headings to appear above data fields on every page if the section goes onto multiple pages, you must place the column headings in the group's Before and PageHeader slots.

Correcting a report design with missing classes

If you open a report that contains a class that has been removed from AFC or a library, e.Report Designer Professional displays a warning with the name of the missing class. In any window that displays classes in a tree structure, such as the structure window, the missing class appears with a ?? icon. You are unable to view the property values or the methods of a missing class. This report is not runnable.

If the missing class is an AFC class, open the report design in an earlier release of e.Report Designer Professional. Migrate the component to a class that the new release supports. After you make these changes, you can continue to develop the component in the new release of e.Report Designer Professional.

Checking the structure of a report design

e.Report Designer Professional supports checking for any internal consistency errors in the report design. For example, if you change a control from an AcLabelControl to an AcTextControl, there is a left over property, Text. Consistency errors can result in unpredictable behavior when the report is run.

Select Report➤Verify Design to start the verification process. If an error is detected, an error report appears, describing the inconsistency and asking you what action you wish to take, as shown in Figure 4-54.

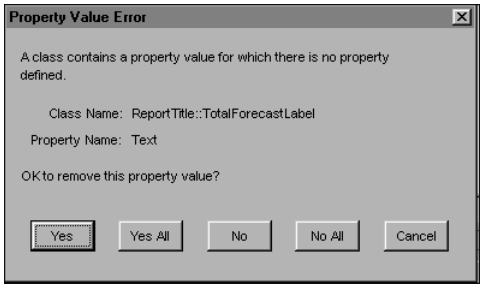


Figure 4-54 Verify Design error

Chapter

5

Formatting report content

This chapter contains the following topics:

- About formatting
- Arranging components
- Formatting data
- Adjusting the spacing between rows of data
- Adding borders and colors to components
- Creating a drop shadow effect within a frame
- Adding shapes and lines
- Adding images
- Formatting report content based on conditions

About formatting

Formatting makes a report visually appealing and effective. You format a report, for example, to highlight certain data, change the display of dates, numbers, or currency values, divide parts of a report, or balance the spacing between blocks of text.

e.Report Designer Professional provides many options for customizing the appearance of components that you place in your report, such as database fields, text labels, or frames. By using various formatting options, you can change the font, size, color, alignment, and other properties of these components. You can add background colors, draw borders around components, add shadows, and so on.

The formatting options are available in the interface elements shown in Figure 5-1, Figure 5-2, Figure 5-3, Figure 5-4, and Figure 5-5:

- The Format toolbar

Figure 5-1 Format toolbar

- Figure 5-2 shows the Font and Appearance pages in the Format dialog box, which appears when you double-click a control

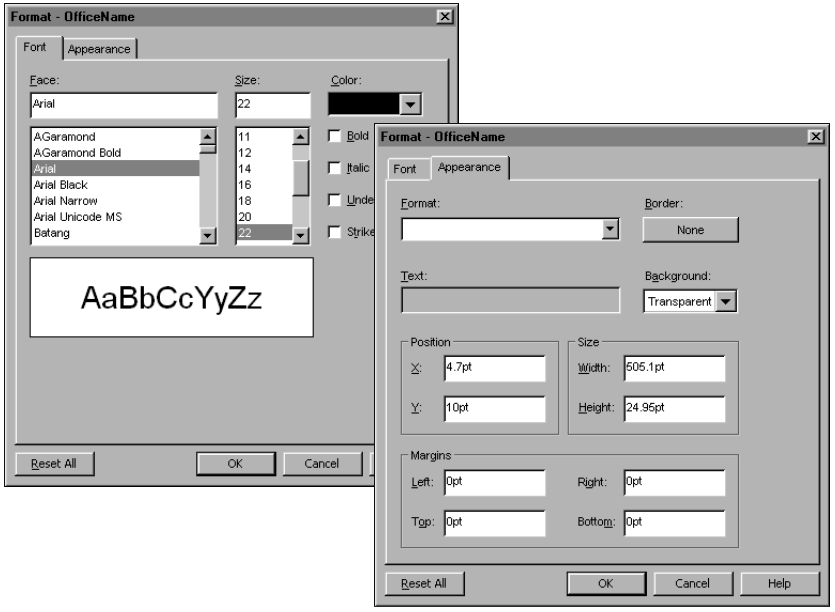


Figure 5-2 Format dialog box

- The Properties page in the Properties window

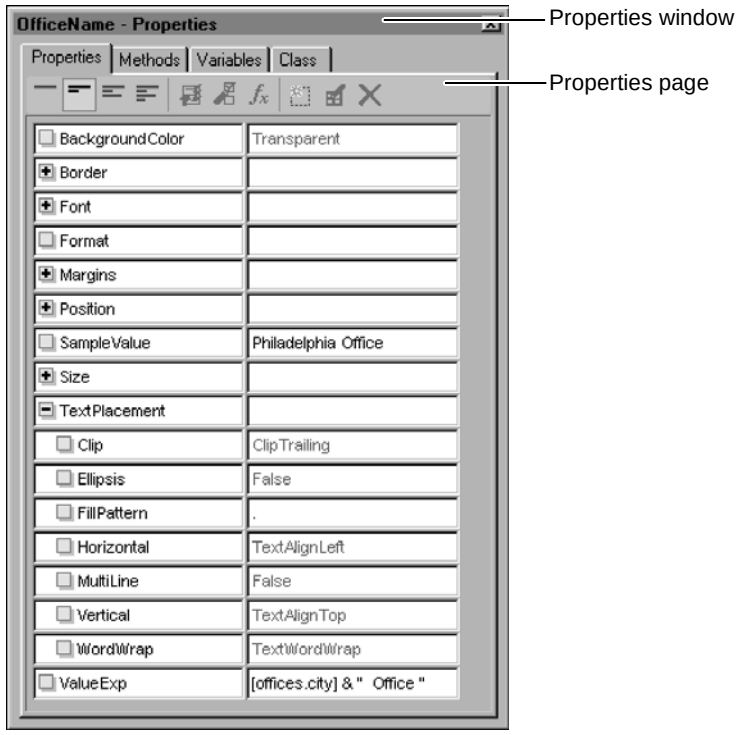


Figure 5-3 Properties window

- The Format menu

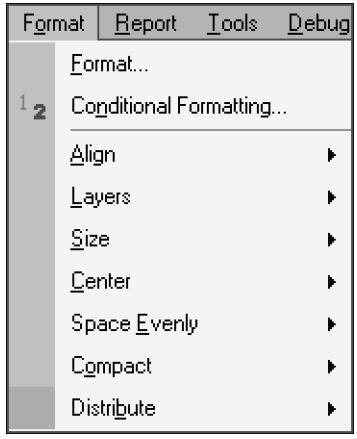


Figure 5-4 Format menu items

- The Conditional Formatting dialog box

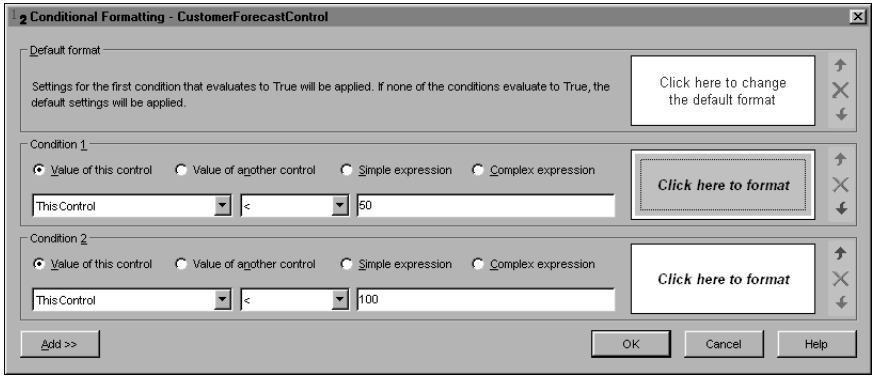


Figure 5-5 Conditional Formatting dialog box

Arranging components

After you place components in a report design, you can rearrange them to create a polished and professional-looking report. This section provides techniques for selecting and arranging components in the following ways:

- Selecting multiple components
- Moving components
- Sizing components
- Aligning components
- Layering components

Selecting multiple components

You can select multiple components to format them all at the same time. After you select multiple components, you can move, align, or resize them as a group. e.Report Designer Professional aligns or sizes multiple components based on the last component that you select. For example, if you select five controls, then choose Format➛Align➛Top, the tops of the controls align to the top of the fifth control.

You can also apply the same font, color, and paragraph style to multiple components. Use the Properties page in the Properties window as shown in Figure 5-6 to format multiple components at the same time.

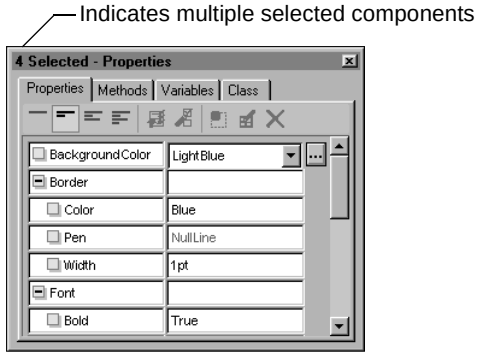


Figure 5-6 Properties window

How to select multiple components

1. Position the mouse pointer on the first component and click once.
2. Press Ctrl while you select another component.
3. Repeat step 2 to select additional components.

How to deselect a component

- To deselect one component in the selection set, press Ctrl while you select the component.
- To deselect all selected components, click the background of the layout window.

Moving components

You can move components in three ways. The technique that you use depends on the level of precision that you want and whether you are moving a component within a frame or to another part of the report. The three options are:

- Use the mouse. This option moves a component from one part of the report to another.
- Use the arrow keys. This option moves a component one pixel at a time, which is useful for small adjustments. If you choose Tools➛Options and enable Snap to grid, the arrow keys move a component one grid square at a time.
- Align components in relation to one another. The alignment feature adjusts the placement of multiple components in a variety of ways, such as distributing components evenly in a frame or aligning the bottoms of selected components.

How to move a component

1. Select the component or components that you want to move.
2. Use one of the following techniques:

- Use the mouse to drag the component to another location.
- Press the left, right, up, or down arrow key.
- Choose one of the options on the Format menu.

Sizing components

When you place a frame or control in a report, e.Report Designer Professional gives the component a default size. You can also use the mouse to size a component at the same time that you place it in the report. In either case, you can easily change the size later.

You can resize a component in three ways:

- Use the mouse.
- Specify values for the component's Size properties.
- Use Format➤Size to resize a component to the same size as another component.

When resizing a frame, make sure it is not wider than the flow on the page. If the frame is too wide, it is truncated.

How to resize a frame

1. Place the mouse pointer at the bottom edge of the frame as shown in Figure 5-7. The pointer changes shape.

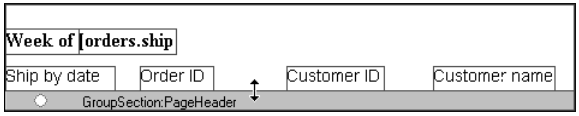


Figure 5-7 Resizing a frame

2. Drag the bottom edge of the frame to the desired position.
3. To set an exact size, set the frame's Size➤Height and Size➤Width properties.

How to resize a component

1. Select the component in the layout.
2. Use the handles to drag the component edges with the mouse until the component is the desired size as is shown in Figure 5-8.

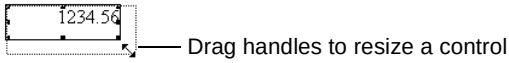


Figure 5-8 Resizing a component

3. To set an exact size, set the component's Size➤Height and Size➤Width properties.

Aligning components

You can align components in the following ways:

- Align selected components in relation to each other using their right, left, top, or bottom edges. You can also align components using their vertical or horizontal centers.
- Space selected components evenly, horizontally or vertically, within the area they occupy.
- Compact selected components so there is no space between them.

You perform these alignment tasks by using commands on the Format menu.

How to align components with each other

1. Select the components in the layout window.
2. Choose Format➛Align.
3. From the menu that appears, choose one of the alignment commands as shown in Figure 5-9.

Figure 5-9 Alignment format menu

The components align to the last component that you selected.

How to space components evenly

1. Select the components in the layout window.
2. Choose Format➛Space Evenly.
3. From the menu that appears, choose either Across or Down as shown in Figure 5-10 and Figure 5-11. The distance between the centers of each pair of rearranged components is equal:
   - Across rearranges components placed horizontally.

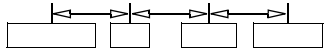


Figure 5-10 Spacing components horizontally

   - Down rearranges components placed vertically.

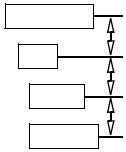


Figure 5-11 Spacing components vertically

How to compact components

1. Select the components in the layout window.
2. Choose Format➛Compact.
3. From the menu that appears, choose either Across or Down. Components appear next to one another without spaces between them as shown in Figure 5-12 and Figure 5-13:
   - Across rearranges components placed horizontally.

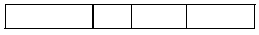


Figure 5-12 Compacting horizontally

   - Down rearranges components placed vertically.

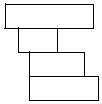


Figure 5-13 Compacting vertically

Arranging components relative to the frame

You can arrange components within a frame in the following ways:

- Center components in relation to the frame as shown in Figure 5-14.
- Distribute selected components evenly, horizontally or vertically, within the frame as shown in Figure 5-15 and Figure 5-16.

You perform these actions by using commands on the Format menu.

How to center components in a frame

1. Select the component or components that you want to center within the frame.
2. Choose Format➛Center.
3. From the menu that appears, choose one of the commands shown in Figure 5-14.

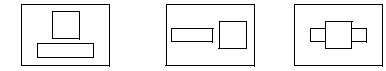
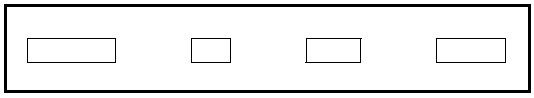


Figure 5-14 Centering components in a frame

How to distribute components evenly in a frame

1. Select the components that you want to distribute evenly within a frame.
2. Choose Format➛Distribute.
3. From the menu that appears, choose either Across or Down. The space between the rearranged components is equal.



Across for components placed horizontally

Figure 5-15 Horizontally distributing components evenly in a frame

Down for components placed vertically

Figure 5-16 Vertically distributing components evenly in a frame

Layering components

You can place any number of components on top of one another. For example, you can layer components to create a graphic image as shown in Figure 5-17. If a component contains a background fill, it obscures the components behind it. You can control the Z-order, or the order in which components overlap, by moving components forward and backward in relation to each other.

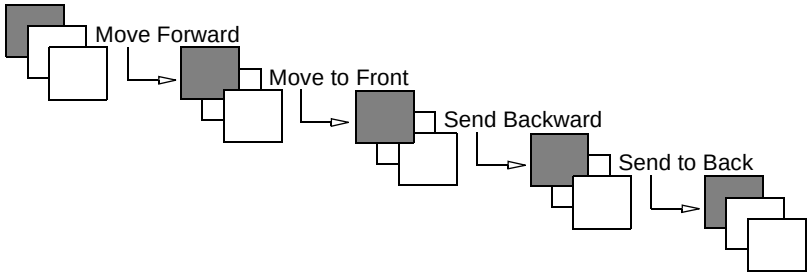


Figure 5-17 Layering components

How to layer components

1. In the layout window, select the component that you want to bring forward or send backward.
2. Choose Format➛Layers.
3. From the menu that appears, choose one of the following commands:
   - To move the component one step closer to the top of the stack, choose Move Forward.
   - To move the component to the top of the stack, choose Move to Front.
   - To move the component one step toward the bottom of the stack, choose Send Backward.
   - To move the component to the bottom of the stack, choose Send to Back.

Formatting data

You format the data in one or more controls by selecting the controls, then making changes using the Format toolbar or the Properties page. If you apply a format using the toolbar, your format choices appear in the Properties page. If you update a format using Properties, the change is reflected in the toolbar.

You can customize how data appears by setting the following:

- Format of numbers, text, dates, and times
  e.Report Designer Professional provides common format styles in which to display numbers, currency, or date values. If you do not choose a format, e.Report Designer Professional displays the data as it appears in the data source. If you want to use some combination of formatting that these styles do not provide, you can specify a custom format using the control's Format property.
- Font face, point size and color
  When choosing fonts, remember that for the report user to view the report with the fonts that you choose, the fonts must also be installed on the user's system.
- Text style as bold, italic, or underline
  These are the styles available on the Format toolbar. Properties includes one additional style, Strikethrough.
- Text justification as left, center, or right
  Within each control, you can display data to the left, center, or right.
- Text alignment at top, center, or bottom

Within each control, you can display data at the top, center, or bottom, assuming the height of the control is larger that the size of data in it.

- Single or multiline text
  e.Report Designer Professional's default behavior is to display a single line of text. If a field in the data source contains a lot of text, you can specify that e.Report Designer Professional display multiple lines of text. Make sure, however, that the control is large enough to accommodate multiple lines. Alternatively, you can use a dynamic text control to display text of varying lengths.
- Method of truncating data or style of overflow characters
  If a single-line control is not wide enough to accommodate the data it contains, you can specify if overflow characters appear or if data should be clipped.

Formatting numeric data

You can display data in an integer or currency control in various formats. You can, for example, display the data with decimal values, in scientific notation, or with a currency symbol. Table 5-1 lists the types of formats that you can choose from the Format toolbar or the control's Format property, and provides examples of how the formatted data appears.

Table 5-1 Numeric data formats

Format type	Example of data display
General Number	3000
Currency	$3,000.00
Fixed	3000.00
Standard	3,000.00
Percent	300000.00%
Scientific	3.00E+03
Yes/No	Yes for a non-zero number No for zero Nothing if the field is empty
True/False	True for a non-zero number False for zero Nothing if the field is empty
On/Off	On for a non-zero number Off for zero Nothing if the field is empty

You can also define your own formats. You can, for example, specify the number of digits after the decimal or add literal characters to the numbers. To define a

custom format, you use special symbols to construct a format pattern for the control's Format property. For a list of these symbols and their descriptions, see the Format function in *Programming with Actuate Basic.*

Table 5-2 shows examples of custom format patterns and their effects on numeric or currency data.

Table 5-2 Custom numeric format patterns

Format pattern	Data in the data source	Results of formatting
0000.00	12.5 124.5 1240.553	0012.50 0124.50 1240.55
$#,###	2000.00 20000.00	$2,000 $20,000
#.000	3000 3000.1237	3000.000 3000.124
"ID "#	15	ID 15

Formatting date and time data

You can display data in a date and time control in many formats. You can display dates and times in short, medium, or long formats, or by month, quarter, or year. Table 5-3 lists the types of formats that you can choose from the Format toolbar or the control's Format property, and provides examples of how the formatted data appears for English (United States) locale.

In e.Report Designer Professional, localemap.xml specifies the formats for these values. The default installation location for localemap.xml file is the <eRDPro_HOME>\config directory. For more information about localemap.xml, see *Working in Multiple Locales using Actuate Basic Technology.*

Table 5-3 Date and time formats in English locale

Format type	Example of displayed data
General Date	4/15/2006
Long Date	Thursday, April 15, 2006
Medium Date	15-Apr-06
Short Date	4/15/2006
Long Time	12:00:00 AM
Medium Time	12:00 AM
Short Time	00:00
Short Week	W16 2006

Table 5-3 Date and time formats in English locale

Format type	Example of displayed data
Short Month	4/2006
Short Quarter	Q2 2006
Short Half	H1 2006
Long Year	2006

You can also define your own date and time formats. You can, for example, specify two-digit months (01 - 12), use two digits for the year, or add literal characters to dates or times. To define a custom format, you use special symbols to construct a format pattern for the control's Format property. For a list of these symbols and their descriptions, see the Format function in *Programming with Actuate Basic.*

Table 5-4 shows examples of custom format patterns and their effects on a date that is stored as 4/15/2006 in the data source.

Table 5-4 Custom format patterns

Format pattern	Results of formatting
mm-dd-yy	04-15-06
ddd, m/d	Thu, 4/15
dddd, m/dd/yy	Thursday, 4/15/06
mmm d	Apr 15
mmmm	April
"Quarter" q	Quarter 2
"Day" y	Day 106

Formatting text data

Typically, you format data in a text control to fix inconsistent or poorly formatted data retrieved from the data source. The data source, for example, can store names with inconsistent capitalization or phone numbers in 1234567890 format. To fix these problems, you use special symbols to construct a format pattern for the text control's Format property. For a list of these symbols and their descriptions, see the Format function in *Programming with Actuate Basic.*

Table 5-5 shows examples of custom format patterns and their effects on text data.

Table 5-5 Custom format patterns and text data

Format pattern	Data in the data source	Results of formatting
(@@@) @@@-@@@@	6175551007 5551007	(617) 555-1007 ( ) 555-1007
!(@@@) @@@-@@@@	6175551007 5551007	(617) 555-1007 (555) 100-7
<	Closed CLOSED	closed closed
>	Closed closed	CLOSED CLOSED

Using code mapping to control the appearance of text

When a report connects to a database that supports code mapping, the code mapping can affect the appearance and behavior of data in text controls. For example, the appearance of a double-byte dash in the report can vary depending on the code mapping and the font you use. The double-byte dash can map to a horizontal bar (0x2015) or to an em-dash (0x2014). The length of the horizontal bar and the em-dash can vary, depending on the font. If you use Arial Unicode MS, the em-dash is longer than the em-dash in MS Gothic. The horizontal bar in MS Gothic is longer than the horizontal bar in Arial Unicode MS. You must consider how the text is mapped and select a font that produces the correct result.

Specifying how data appears if its value overflows the control size

If a control is not wide enough to accommodate the data it contains, you can use the TextPlacement properties to specify whether e.Report Designer Professional displays overflow characters or truncates the data. Figure 5-18 shows the TextPlacement properties.

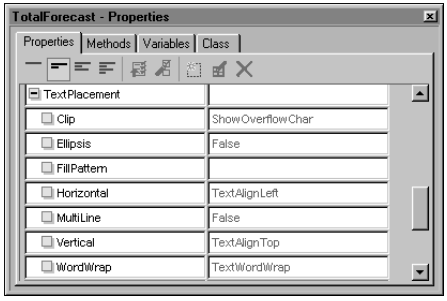


Figure 5-18 TextPlacement properties

It is especially important to specify how large numbers appear. A value such as 500,000,000 can appear as 500,000 or 000,000, or ********, depending on the size of the control and the TextPlacement properties that you set.

Using TextPlacement properties, you can truncate data on the left or right and you can display an ellipsis to indicate that the data is truncated. For text data, you can also specify whether text is truncated by word or character.

Figure 5-19 shows how the settings of a currency control's Clip and Ellipsis properties determine the appearance of the value $1234.50 if the currency control is not wide enough to accommodate the value.

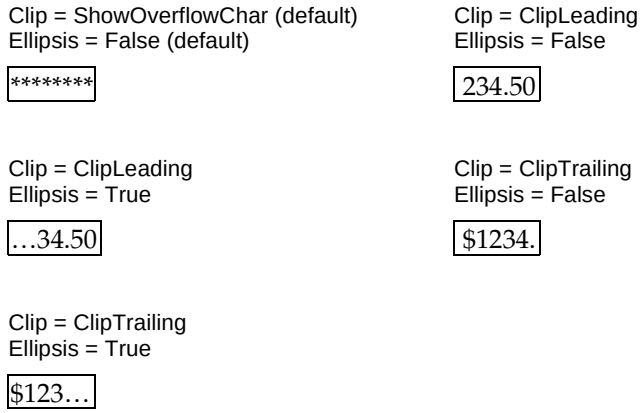


Figure 5-19 Clip and Ellipsis properties

Figure 5-20 show how the settings of a text control's Clip, Ellipsis, and WordWrap properties determine the appearance of the value 555 Franklin South Ave if the text control is not wide enough to accommodate the value.

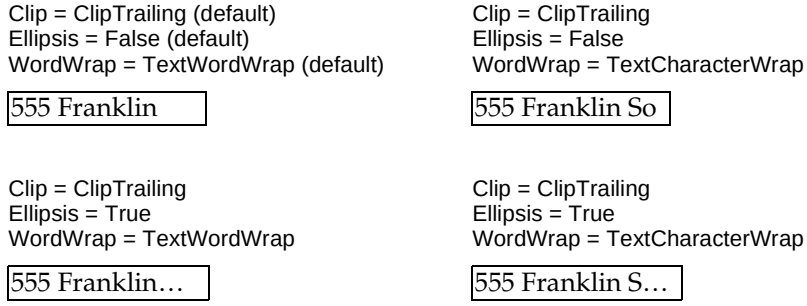


Figure 5-20 Text control box limitation example

Adjusting the spacing between rows of data

The height of the frame containing data controls determines the amount of space that appears between rows on the report. The taller the frame, the more space between rows of data. Compare the following illustrations, Figure 5-21 and Figure 5-22.

Design 1

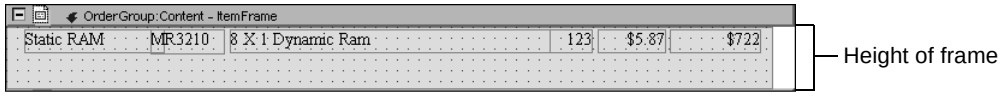
Output 1

Static Ram	MS3290	32M x 9 Static Ram	1713	$150.00	$256,950
Dynamic Ram	MR0480	4M x 8 Dynamic Ram	1729	$9.00	$15,561
Static Ram	MSL3280	32M x 8 Static Ram, 3.3 volts	1746	$210.00	$366,660
Static Ram	MS0490	4M x 9 Static Ram	1734	$27.00	$46,818
Dynamic Ram	MRL1640	16M x 4 Dynamic Ram, 3.3 volts	1758	$53.00	$93,174
Dynamic Ram	MR1680	16M x 8 Dynamic Ram	1715	$33.00	$56,595
DSP	MDSPL04	32 x 16 Programmable DSP, 3.3 volts	1728	$340.00	$587,520

Height of each row of data

Figure 5-21 Short frame height

Design 2



Output 2

Static Ram	MS3290	32M x 9 Static Ram	1713	$150.00	$256,950
Dynamic Ram	MR0480	4M x 8 Dynamic Ram	1729	$9.00	$15,561
Static Ram	MSL3280	32M x 8 Static Ram, 3.3 volts	1746	$210.00	$366,660
Static Ram	MS0490	4M x 9 Static Ram	1734	$27.00	$46,818
Dynamic Ram	MRL1640	16M x 4 Dynamic Ram, 3.3 volts	1758	$53.00	$93,174

Height of each row of data

Figure 5-22 Tall frame height

To adjust the amount of space between rows, change the height of the frame. The same technique applies when adjusting the spacing between different parts of the report.

Adding borders and colors to components

Add borders and background colors to fields or frames on your report to highlight important information. e.Report Designer Professional provides a variety of border styles and colors for frames and controls. For borders, you can select the line's thickness, color, and style. You can then select a background color to fill the space within the border.

You can achieve a similar effect by placing a rectangle control behind the component. It is better, however, to set the border property of the component because the border adjusts to the component's size, whereas the size of the rectangle control is fixed at design time. In addition, using the rectangle control to create borders unnecessarily increases the number of components in a report, which increases report generation time.

How to add a border and color to a component

1. Double-click the control to format.
2. In Format—Font, choose Appearance.
3. In Format—Appearance to specify a border, choose the button for Border.
4. In Line Style, select Add Line and specify the border properties:
   - In Thickness, specify a thickness.
   - In Color, specify the color in one of the following ways:
     - Select a color from the drop-down list.
     - In the color drop-down list, choose Custom Color to select a color from Color, create a custom color, or type the RGB value for the color.
   - In Pattern, select a line style from the drop-down list, as shown in Figure 5-23.

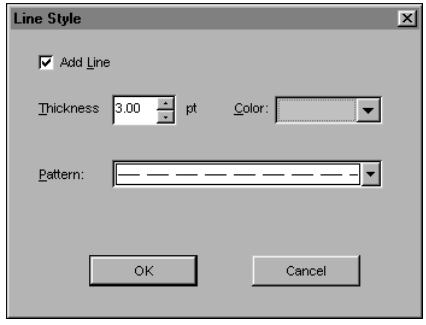


Figure 5-23 Choosing a line style

Choose OK.

5 In Format—Appearance, in Background, to specify a fill color within the border, choose a color or choose Custom Color from the drop-down list, as shown in Figure 5-24.

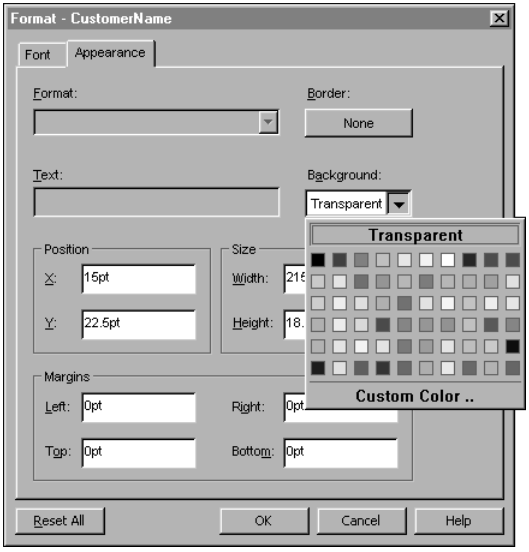


Figure 5-24 Choosing a background color

Choose OK.

Creating a drop shadow effect within a frame

You can create a drop shadow effect within a frame that has a border, as shown in Figure 5-25.

Figure 5-25 Shadow effect behind a frame

The addition of a border and shadow not only draws attention to a frame but also enhances it visually. To create this effect, insert and layer two rectangles behind the controls in the frame.

How to create a drop shadow effect within a frame

1 Choose Insert➛Rectangle and click in the frame. A white rectangle appears in the frame, on top of the controls, as shown in Figure 5-26.

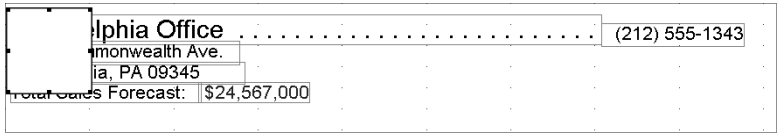


Figure 5-26 Shadow in front of text

2 Place the rectangle behind the controls by choosing Format➛Layers➛Send to Back. Figure 5-27 indicates how the white rectangle appears after it is moved behind the text.

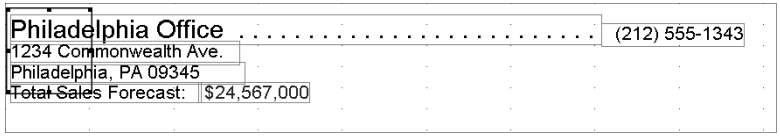


Figure 5-27 Shadow behind text

3 Resize the rectangle so it surrounds the controls in the frame as is shown in Figure 5-28.

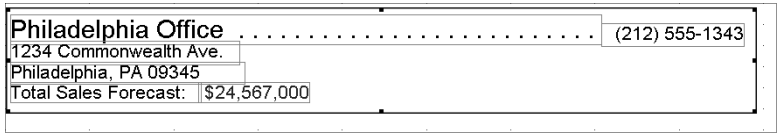


Figure 5-28 Resizing rectangle behind text

4 Insert a second rectangle in the frame. It appears on top of the controls.

5 Resize and reposition the second rectangle so that it is offset slightly from the first rectangle, as shown in Figure 5-29.

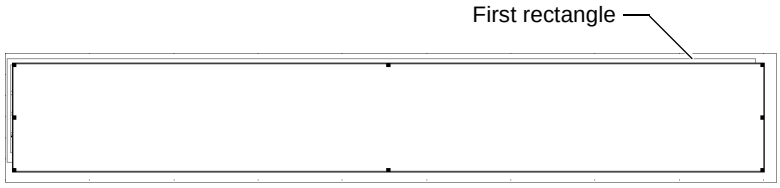


Figure 5-29 Second rectangle

6 Set the second rectangle's FillColor property to the color that you want for the shadow, as shown in Figure 5-30.

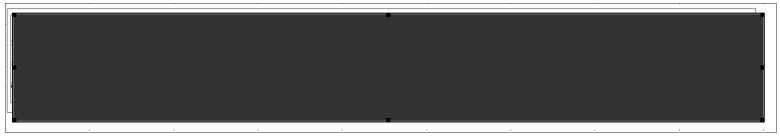


Figure 5-30 Coloring the shadow rectangle

7 To create a shadow whose border is the same color as the shadow itself, set the rectangle's LineStyle➛Pen property to NullLine.

8 Place the second rectangle behind all the controls in the frame by choosing Format➛Layers➛Send to Back, as indicated in Figure 5-31.

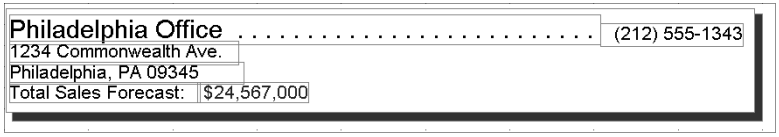


Figure 5-31 Sending the shadow rectangle to the back of the frame

Adding shapes and lines

e.Report Designer Professional provides several drawing controls, such as lines and rectangles, that you can use to design graphics for your report. Lines, in particular, are useful, for example, for dividing report details from summary information or dividing page headers from report details. You can use rectangles to visually group related data or to highlight certain information, as shown in Figure 5-32.

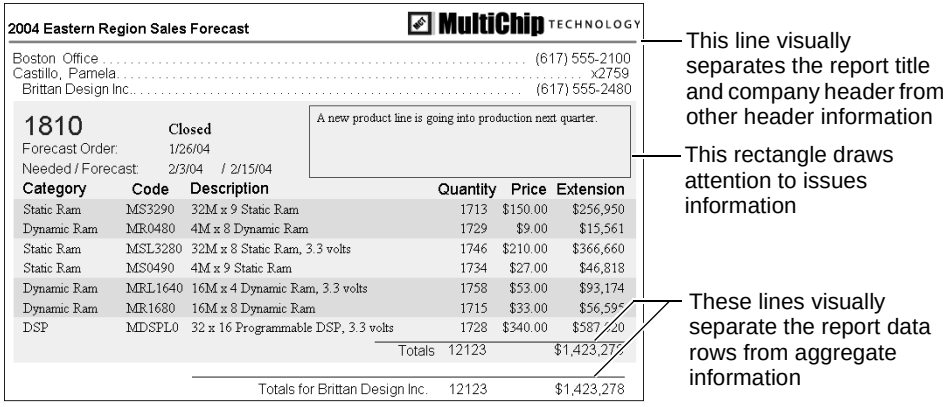


Figure 5-32 Lines distinguish components within a report

When using lines to visually separate different parts of a report, be careful to put them in the correct frame. In the previous illustration, the line that separates the data rows from the aggregate information is placed on top of the aggregate fields in a frame in the After slot, not at the bottom of the detail frame. If you place the line in the detail frame instead, a line appears below every row of data.

You can also use Scalable Vector Graphics (SVG) to create shapes and lines in e.Report Designer Professional. For more information about Scalable Vector

Graphics (SVG), see Class AcDrawingSVGPlane in *Programming with Actuate Foundation Classes.*

How to add shapes and lines

1. Choose Insert➛Line or Rectangle.
2. Click in a frame to insert the rectangle or line. The component appears in the frame.
3. Resize the component.
4. To set its properties, right-click the component, then choose Properties.
5. In the Properties page, set the following properties:
   - To specify a line style, set the LineStyle properties, Color, Pen, and Width.
   - For a rectangle, you can specify a fill color by setting the FillColor property.

Adding images

Images add visual appeal to reports. Using e.Report Designer Professional, a report developer can add a company logo in the page header, include employee photographs, or use icons instead of text labels.

You can place the following types of images in a report design:

- BMP
- GIF
- JPEG
- PCX
- TGA
- The following TIFF formats are supported on only Windows systems. Actuate does not support the TIFF format on UNIX systems.
  - 1 bit per pixel (uncompressed)
  - 8, 16, or 24 bits per pixel (LZW compression)
  - CCITT Fax Groups 3 and 4 compression

You can view all these image types when viewing the generated report on the web. However, some of these image types are not supported on other viewing platforms. The view perspective, for example, does not display GIF images.

Planning to use an image file

When you use an image in a report, the report design either references or embeds the image file. When you modify the image file, the updated image appears in the report. Because the report design references the image file, consider the following factors if you plan to publish the report to an Encyclopedia volume or distribute it using e-mail:

- Where you store the image file
  When you design a report, store image files in the same directory as the report design (.rod) file. You can store the image files in a different directory if you include the directory path in the global search path.
- When e.Report Designer Professional displays the image in the report
  e.Report Designer Professional can display an image in the report at one of the following times:
  - When the application compiles the report object design (.rod) file
  - When the application builds the report object instance (.roi) file
  - When the view perspective displays the ROI or when the report is printed

Inserting an image in a report

A report developer can insert an image in a report design as either an embedded object or as a linked external file. You can embed an image in a report object executable (.rox) file or in a report object instance (.roi) file. Embedded images become part of the report. To update an image, the report user must build and run the file. Embedding an image increases the size of the ROX or ROI but makes the report easier to move and distribute without losing the image.

You can link an external image to an ROI. When you link an image, you must provide a path or a URL to the image file. If you link the image, the ROI is smaller than if you embed the image, and you can change images without re-running the report. e.Report Designer Professional searches the current directory and the global search path to find the image file. If the application cannot find the image, an X appears in the generated report, as shown in Figure 5-33.

To specify whether the image is embedded or linked, use one of the following settings for the Embedded property of the image file:

- ImageDesignTime. To distribute an ROX and an image in the same file, use ImageDesignTime. When the report runs, the resulting ROI also includes the image. This setting embeds the image in the report.
- ImageFactoryTime. Use this setting to embed an image in an ROI and distribute the ROI with the image in the same file.

- ImageFactoryTimeSingle. Include the image only once when the report object instance (.roi) file builds, resulting in faster report generation and smaller ROI and PDF files.
- ImageViewTime. Use this setting to link an ROI to an external image file. When you distribute the ROI, the report loads the image at view time.
- ImageViewTimeSingle. Include the image only once when the ROI appears in the report viewer, making PDF generation faster, and PDF files smaller.

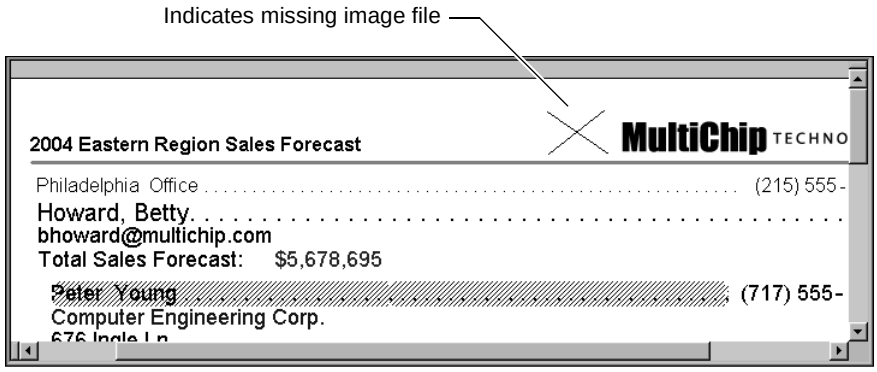


Figure 5-33 Lost link to an external image file

How to embed an external image in a report design

1. Choose Insert➛Image.
2. Click in the frame where you want to insert the image. Component Properties appears, as shown in Figure 5-34.

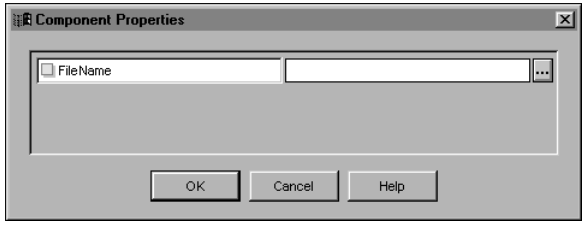


Figure 5-34 Component Properties window

3. To select an image, in FileName, complete one of the following tasks to provide a reference to the image file:

   - Type the file name or a path to the image file. If you do not know the name or the path, choose Browse to navigate to the file.
   - Type a URL using HTTP that specifies the image file. The URL must access the image directly, for example http://MyServer:8900/Images /MyImage.jpg. Using HTTPS in the URL is not supported. If you insert an

image from a servlet, you see an X in the design perspective in place of the image, but the image appears when the report is run.

Choose OK.

4 Use the default Embedded property value, ImageDesignTime. This setting embeds the image in the report when e.Report Designer Professional compiles the report. If you distribute a report using e-mail or an Encyclopedia volume, use the ImageDesignTime setting.

Inserting an image using a database reference

If a database column contains the file name of an image, you can use that column to reference the image and make it available to a report design. For example, a database table for a product catalog can contain a column that references product images. To use a database column to reference an image, you type the table and column name in the FileNameExp property, using the following format:

```
[<tablename>.<columnname>]
```

Note that you cannot load a binary image from a database into an image control. Instead, you use the image name in the data column to create a reference to the image.

How to insert an image using a database column that contains the image name

1 Choose Insert➛Image.

2 Click in the frame where you want to place the image. Component Properties appears.

3 Choose OK without typing a value for FileName. The image control appears in the design.

4 Right-click the image control, then choose Properties. The Properties page for the image control appears.

5 In the FileNameExp property, type the name of the database table and column containing the image name, as shown in the following example. You must enclose the table and column name in brackets.

```
[catalog.itempic]
```

Formatting report content based on conditions

When you format a control by setting the values in the Format dialog or on the Properties window, the formatting applies to all instances of the control in the generated report. For example, if you specify that an item's price appears in Times font and in blue, all item prices in the generated report appear in Times

font and in blue. This type of formatting, known as static formatting, sets the appearance of the data during report design.

You can also change the format of a control based on data or parameters. For example, you can specify that item prices appear in blue if the value exceeds $1,000.00 and in green if the value is less than or equal to $1,000.00. This type of formatting is called conditional formatting. With conditional formatting, the appearance of the data is set when the report runs.

Conditional formatting makes a report more readable. For example, Figure 5-35 shows a part of a report without conditional formatting on the left. Because the regional sales manager wants the order totals for customers whose credit rank is C to stand out in the report, the report developer sets the conditional formatting to display the credit rank and order totals in bold, underlined text for customers whose credit rank is C, as shown in the example on the right.

Order	Customer Name	Credit Rank	Order Total
1670	TeleSystems Inc.	A	$871,655.00
1685	Signal Design Corp.	A	$278,291.00
1690	Advanced Engineering	C	$1,494,853.00
1715	Signal MicroSystems	C	$144,836.00
1720	Computer MicroSystems	B	$202,754.00
1725	TeleBoards Co.	A	$1,135,528.00
1730	Technical Design Corp.	B	$1,777,989.00
1740	CompuEngineering	B	$1,019,075.00

Order	Customer Name	Credit Rank	Order Total
1670	TeleSystems Inc.	A	$871,655.00
1685	Signal Design Corp.	A	$278,291.00
1690	Advanced Engineering	**C**	**$1,494,853.00**
1715	Signal MicroSystems	**C**	**$144,836.00**
1720	Computer MicroSystems	B	$202,754.00
1725	TeleBoards Co.	A	$1,135,528.00
1730	Technical Design Corp.	B	$1,777,989.00
1740	CompuEngineering	B	$1,019,075.00

Figure 5-35 A report without and with conditional formatting

You can also display different parts of a report based on a condition. For example, you can display different sections for permanent or contract employees. You use a conditional section to control these types of reporting needs.

Applying conditional formatting

To apply conditional formatting to a component, you specify a format to apply when a data value meets a condition. A control's ValueExp property value determines the data value that it displays. You can specify multiple conditions to apply multiple formats. The first condition that evaluates to true determines the format that applies to the component.

Review and test the conditions to ensure that they do not conflict with each other and produce unexpected results. For example, consider a reporting requirement for a control's value to appear with the following formats:

- Bold, italic text on a gray background when the value is less than 50
- Bold, italic text on the default background color when the value is less than 100

If you order the conditions with the comparison against 100 first, the formatting rule for a value less than 50 never runs. If you order the conditions with the

comparison against 50 first, but set its format to have only a gray background, the bold and italic font settings for a value that is less than 100 do not apply. You must set all the formatting properties that you need for a condition in the condition itself. To apply the conditions so that the report appears in the way that you need, set the following conditions in the order shown in the following list and in Figure 5-36:

- An order total value that is less than $50 appears in a bold, italic font with a gray background.
- An order total value that is less than $100 appears in a bold, italic font.

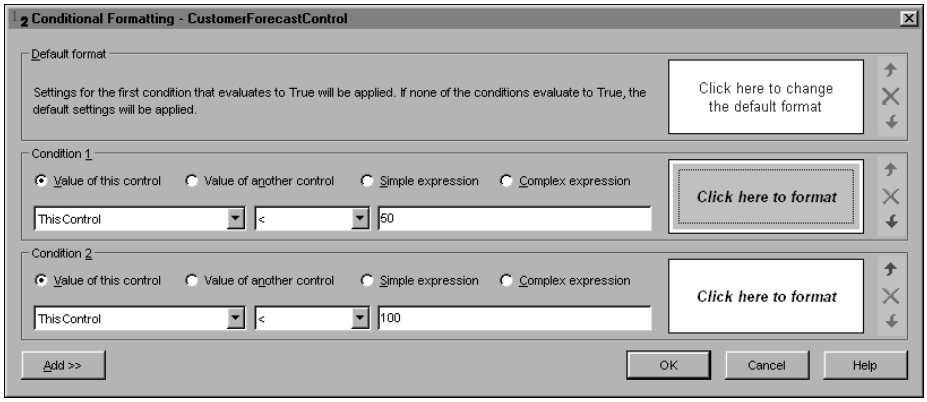


Figure 5-36 Conditional Formatting control

Conditional Formatting supports creating various types of expressions as shown in the following list:

- To base the conditional formatting of a selected control on its data value:
  - Select Value of this control.
  - Select a comparison operator.
  - Provide a value or values for the comparison. The value or values that you type must be literal values, not expressions.

  For example, you can specify that conditional formatting applies when a control's value is less than or equal to 100000, as shown in Figure 5-37.

Figure 5-37 Conditional formatting based on a control's value

- To base the conditional formatting of a selected control on the value of another control in the same frame:
  - Select Value of another control.

- From the first drop-down list, select the control whose value determines the formatting of the other control.
- Select a comparison operator from the second drop-down list.
- Provide a value or values for the comparison. The value or values that you type must be literal values, not expressions.

For example, you can specify that conditional formatting applies when another control's value is between 100000 and 200000, as shown in Figure 5-38.

Figure 5-38 Conditional formatting of a control's value

- To base the conditional formatting of a selected control on a data column:
  - Select Simple expression.
  - Select a column name from the first drop-down list.
  - Select a comparison operator from the second drop-down list.
  - Provide a value or values for the comparison. The value or values that you type must be literal values, not expressions.

  For example, you can specify that conditional formatting applies when the [customers.creditrank] column value is like C, as shown in Figure 5-39.

Figure 5-39 Conditional formatting for a simple expression

- To base the conditional formatting of a selected control on a more complex expression, such as multiple expressions that must all be true or aggregate expressions:
  - Select Complex expression.
  - Type an expression.
  - Choose Expression Builder to use Expression Builder to provide an expression.

  For example, you can specify that conditional formatting applies when a sum is greater than or equal to $500 and is less than or equal to $1000, as shown in Figure 5-40.

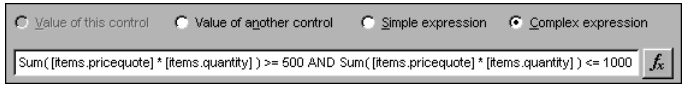


Figure 5-40 Conditional formatting for a complex expression

- To base the conditional formatting of a selected control on a parameter, select Complex expression and provide the fully qualified parameter name as the comparison value.

  For example, you can specify that conditional formatting applies when a control's value equals the value of a parameter, as shown in Figure 5-41. In this example, a user provides the ad hoc parameter value on Requester when the report runs.

Figure 5-41 Conditional formatting for a parameter

How to conditionally format a control

1. In Layout or Report Structure, select a control.
2. Choose Format➛Conditional Formatting.
3. In Conditional Formatting, complete the following tasks:
   - Select one of the following options:
     - Value of this control
     - Value of another control
     - Simple expression
     - Complex expression
   - Use the drop-down lists and text boxes to create an expression. This expression specifies a condition. If that condition evaluates to true when the report runs, the conditional formatting that you specify in steps 5 and 6 applies.
4. Choose Click here to format.
5. In Format—Font, specify font property values. Choose Appearance.
6. In Format—Appearance, specify property values for the control's appearance. Choose OK.

   Conditional Formatting displays a preview of the format that you specified on the Click here to format button. The button appears using black, bold, italic text on a gray background with a 1-point, black, solid-line border.
7. You can modify conditions in the following ways:
   - To add another condition, choose Add then specify a condition and format.

- If you specify more than one condition for a control, you can change the order of precedence for the conditions by choosing the up or down arrow.
- To remove a condition, choose Delete.

Choose OK.

To conditionally format a control, you can also create or modify a property to enable conditionally formatting it.

Inheriting conditional formatting

A class can inherit conditional formatting from its parent class. This behavior supports adding a component that uses conditional formatting to a library then including that library in a report design. If a component inherits conditional formatting, Conditional Formatting displays the condition as unavailable to change, as shown in Figure 5-42.

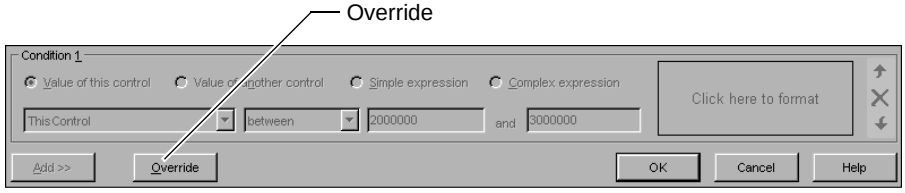


Figure 5-42 Overriding an inherited conditional format

In Conditional Formatting, choose Override to disengage the inherited conditional format.

Chapter

6

Customizing page layout

This chapter contains the following topics:

- About page layout
- Planning the page layout
- Specifying page size, orientation, and margins
- Understanding the effects of page size on online and printed reports
- Specifying headers and footers
- Designing background patterns
- Designing a multicolumn page layout
- Designing multiple page layouts

About page layout

A page layout specifies the dimensions and style of the pages on which report data appears. You can specify, for example, that the page size is 7" x 9", the printable area is 6" x 8", that the company logo always appears in the top left corner, and the page number appears in the bottom right corner. You can create single- or multiple-column page layouts, or design special effects, such as a watermark that appears behind the report data.

Figure 6-1 shows a few examples of page layouts that you can create using e.Report Designer Professional.

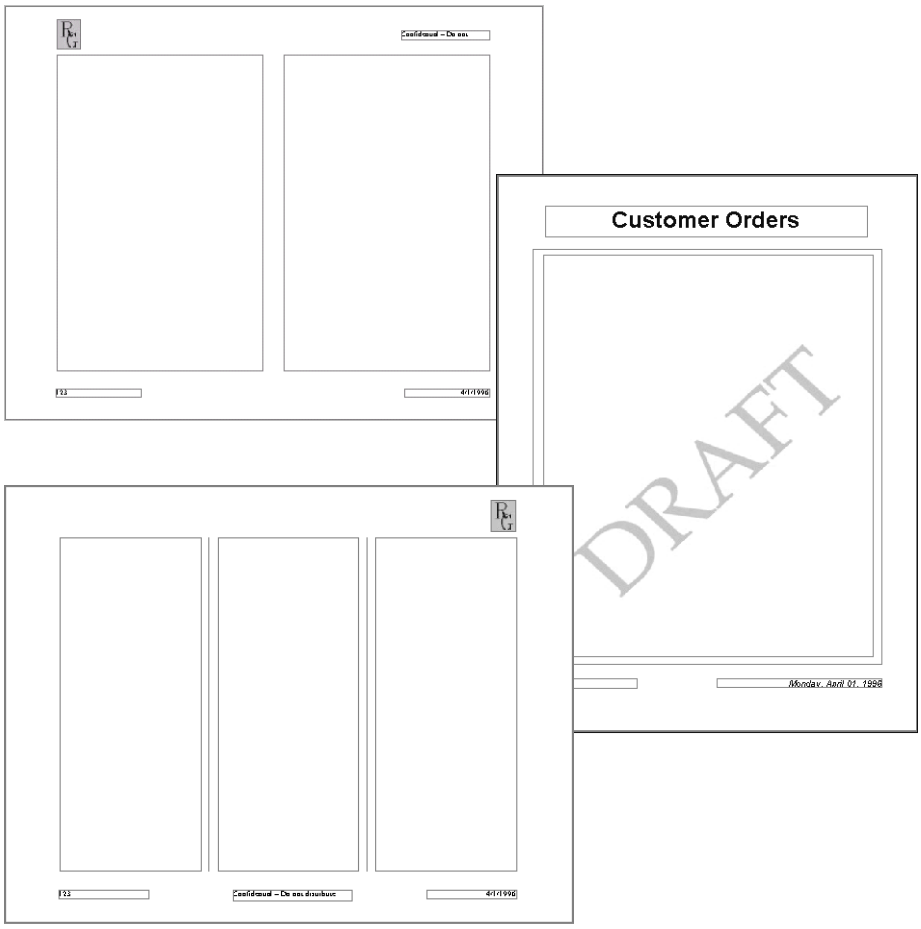


Figure 6-1 Page layout examples using e.Report Designer Professional

Understanding the relationship between page layout and report layout

Laying out a page and laying out a report are two separate tasks:

- When you lay out a report, you place data in the report and specify how to group, sort, and present it. Laying out a report is one of the primary tasks of building a report.
- When you lay out a page, you design the appearance of report pages without having to understand how report data is organized. Think of this task as designing a physical piece of paper, akin to letterhead paper or corporate stationery, on which report data is printed. Designing a page layout is an optional task. e.Report Designer Professional provides a default page layout when you create a new report.

Report layout typically determines how you design the page layout. For example, to display data in a table that contains many columns, you can design a page with landscape orientation to accommodate all the columns. Page size, margins, and orientation, however, determine the amount of space available for laying out data. When you start a new report, it is helpful to perform the initial layout tasks in the following order:

- Design a prototype of the report on paper. The prototype, which can be a simple pencil sketch, should show the report data, page orientation, page headers and footers.
- Set up, at a minimum, the basic page layout, including the paper size, margins, and orientation. This basic layout defines the area in which you can place data.
- Lay out the report data.

After creating the initial page and report layouts, you typically need to fine-tune both layouts. As you place, resize, and reposition data, you might, for example, need to adjust the page margins to accommodate the data.

Viewing the page layout

When you create or open a report design, the Layout window always shows the report layout. To view a page layout, expand PageList in Report Structure, then right-click the Page component. In the context menu that appears, choose Edit Layout, as shown in Figure 6-2.

In the Layout window, you can view either the report layout or the page layout but not both at the same time.

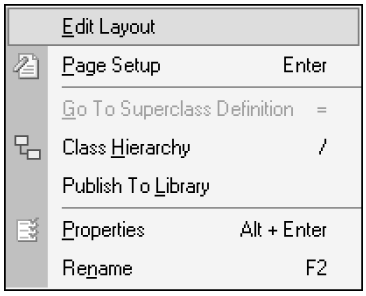


Figure 6-2 Edit Layout

Figure 6-3 shows the default page layout when you create a new report using the Blank Report option.

Default page layout

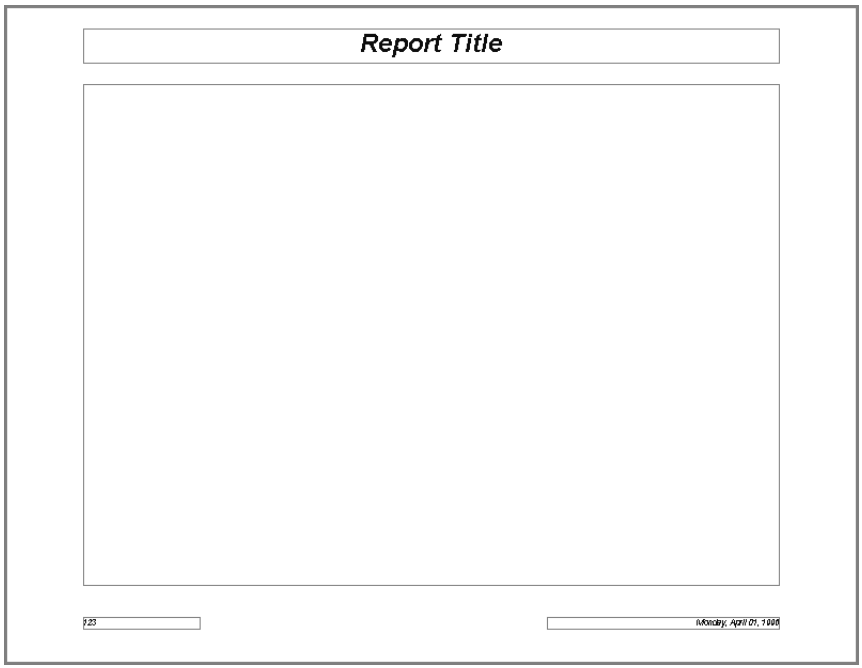
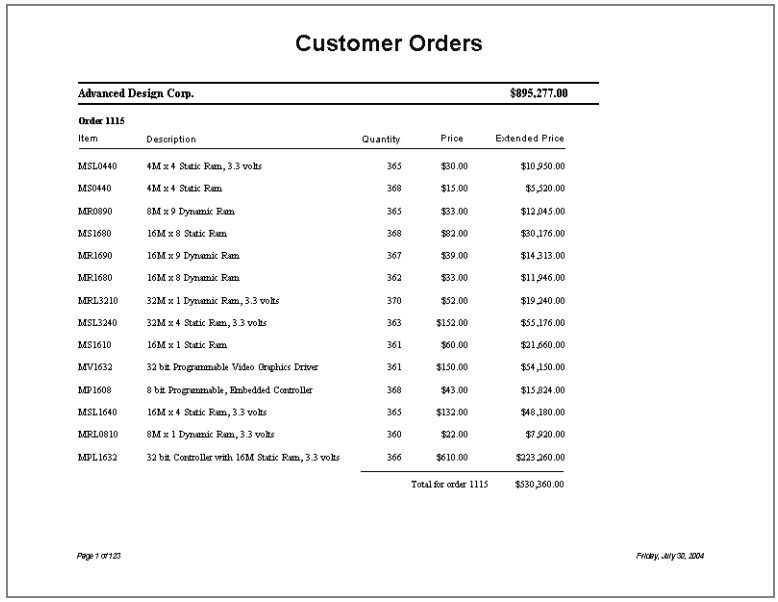


Figure 6-3 Default page layout

Figure 6-4 shows a page of a generated report that uses the page layout with one change. Customer Orders replaces Report Title.

Report output



Customer Orders

Advanced Design Corp. **$895,277.00**

Order 1115

Item	Description	Quantity	Price	Extended Price
MSL0440	4M x 4 Static Ram, 3.3 volts	365	$30.00	$10,950.00
MS0440	4M x 4 Static Ram	368	$15.00	$5,520.00
MR0890	8M x 9 Dynamic Ram	365	$33.00	$12,045.00
MS1680	16M x 8 Static Ram	368	$82.00	$30,176.00
MR1690	16M x 9 Dynamic Ram	367	$39.00	$14,313.00
MR1680	16M x 8 Dynamic Ram	362	$33.00	$11,946.00
MRL3210	32M x 1 Dynamic Ram, 3.3 volts	370	$52.00	$19,240.00
MSL3240	32M x 4 Static Ram, 3.3 volts	363	$152.00	$55,176.00
MS1610	16M x 1 Static Ram	361	$60.00	$21,660.00
MV1632	32 bit Programmable Video Graphics Driver	361	$150.00	$54,150.00
MP1608	8 bit Programmable, Embedded Controller	368	$43.00	$15,824.00
MSL1640	16M x 4 Static Ram, 3.3 volts	365	$132.00	$48,180.00
MRL0810	8M x 1 Dynamic Ram, 3.3 volts	360	$22.00	$7,920.00
MPL1632	32 bit Controller with 16M Static Ram, 3.3 volts	366	$610.00	$223,260.00
			Total for order 1115	$530,360.00

Page 1 of 123

Friday, July 30, 2004

Figure 6-4 Page layout with title change

About the page layout components

When you start a new report using a wizard or the blank report option, e.Report Designer Professional adds the following required components to your report design. These components form the basic structure of a page layout:

- PageList

  This component specifies the overall page style for the report. e.Report Designer Professional provides three types of page lists:

  - Simple page list. Use this page list to create one page layout for all pages in the finished report. This style is the default.
  - Title and body page list. Select this page list to create two page layouts, one for the first page of the report, another for the rest of the report pages.
  - Left and right page list. Select this page list to create two page layouts, one for even-numbered pages, another for odd-numbered pages.

- Page

  This component represents a physical page. Use it to design your page layout visually. You can add controls to a page to display images or text that you want to appear on every page of the finished report. The number of pages that you use in a report design depends on the style of page list that you select. The

simple page list requires one page. The title-and-body and left-and-right-page lists each require two pages.

- Flow

  This component determines the area of the page where report data appears when you run or print the report. Because all report data must appear within a flow, each page component must contain at least one flow. You can add multiple flows to a page to create a multiple-column page layout.

  Images or text that are part of the page layout typically go directly on the page, not on a flow. In a generated report, any image or text placed on the flow appears behind report data.

Figure 6-5 shows an example of page layout components as they appear in Report Structure and Layout. In the example, the report uses the Title and body page list to create two pages with different layouts. Report Structure shows the hierarchical relationship of the components. The page list contains the pages, and the pages contain flows and other controls. Layout displays the visual page layout for the title page. The flow indicates where data appears in the report. Static controls, such as the image, the line controls, and the two labels, appear outside the flow.

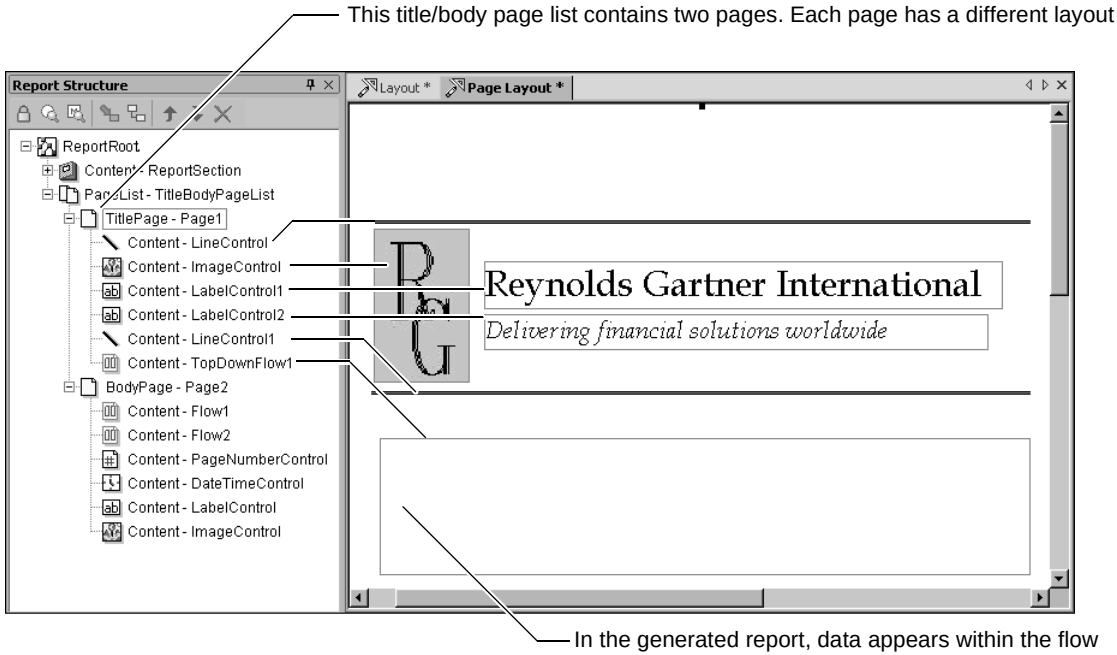


Figure 6-5 Page layout components in Report Structure and Layout

Planning the page layout

e.Report Designer Professional provides a default page layout. You can begin planning a page layout by reviewing the default page layout and deciding what changes, if any, to make. After reviewing the default page layout, make the following design decisions:

- Decide whether you are designing a page layout primarily for a print or online report, or a page layout that is suitable for both. Aside from standard design considerations, such as platform-appropriate fonts, colors, and image types, an important decision is whether report data appears on a single scrollable page or on multiple fixed-sized pages. A single page is typical for online viewing while fixed-size pages tend to be more suitable for both online and printed reports.
- Decide whether you want to use the same layout for every page or different layouts for different pages.
- Decide the page dimensions—paper size, margin sizes, and page orientation. If you use multiple page layouts, you can set different margin sizes for each page. You can also set a different page size and orientation for each page, although such a design would be unusual. If the report is for international distribution, consider using a page size that conforms to standard paper sizes in those countries.
- Decide the design of the page. If you use multiple page layouts, create a design for each page. Some things to consider are the number of columns, and informational text or images that you want to display on the report pages. These tasks are discussed later in this chapter.

After making these design decisions, you have a better idea whether it is easier to modify the default page layout or delete the existing page layout components and start from the ground up. The most significant change is if you decide to use multiple page layouts. In this situation, you have to replace the default page list.

Specifying page size, orientation, and margins

e.Report Designer Professional uses the following default page settings:

- Letter size, 8.5" x 11"
- Top, bottom, left, and right margins of 1"
- Portrait orientation if you create a report using a wizard and do not select a page orientation
- Landscape orientation if you create a report using the Blank Report option

You can change these page attributes at any time. For example, report developers in locales that use A4 paper might want to change the page size. It is best to set page size before you begin laying out report data. If, for example, you change the page orientation from landscape to portrait after you lay out your report data, frames and controls are likely to be truncated because the width of the page decreases significantly.

To specify page attributes, choose Tools➤Page Setup to display the Page Setup dialog shown in Figure 6-6.

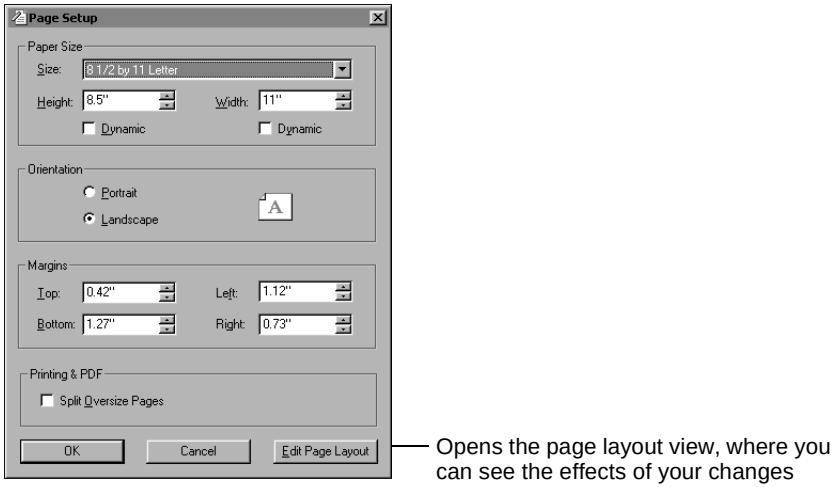


Figure 6-6 Page setup dialog box

Use Edit Page Layout to open the page layout view and adjust the fit of the page components to the new page size. While the other Page Setup options are self-explanatory, you should be aware of the effects that some of your choices have on the online and printed report. The following sections describe how to specify page attributes and the effects they have.

Specifying a fixed page size

You can select from a list of standard paper sizes, such as letter size, legal size, A4, or B4. You can also specify a custom size by entering a specific width and height. Use one of the following ways to specify the page size:

- Select the page for which you want to set the height, then choose Tools➤Page Setup. Then, select from a list of standard paper sizes or specify a specific height and width for paper size.
- Set values for the page's Size➤Height and Size➤Width properties.

The page size you specify determines the size of the generated report pages. It does not specify the size of the physical paper to use when you print the report. In some cases, particularly if you specify a custom size, the appearance of the

printed report is different from the appearance of the online report. The margins, page breaks, and number of pages can differ.

When you specify a paper size in the report design, you should print the report on standard paper sizes to check the appearance of the printed report. To design a page layout that is suitable for both online and print, specify a standard paper size that matches the paper size to which most of your report users print.

Specifying a dynamic page size

If you are designing a report for online or web viewing where users are accustomed to viewing information on a single scrollable web page, you can specify that the report appear on a single page. For usability reasons, you should choose this option only for short reports. If the report is too long, the user has to scroll too much to read the entire report.

To specify that a report appear on a single scrollable page, choose Tools➛Page Setup, then select Dynamic under Paper Size—Height or Width, as shown in Figure 6-7.

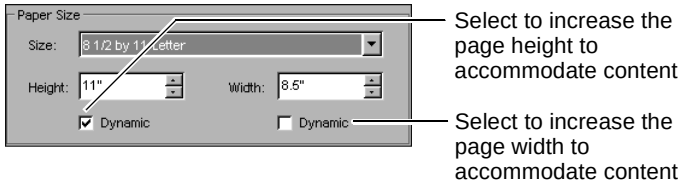


Figure 6-7 Specifying report page size

If you select dynamic height or width, this setting has precedence over the value set in Height or Width.

Figure 6-8 shows how report output differs when you specify a dynamic page size or set a specific page size. In the first report, which uses dynamic page size, the page expands to contain all the data. In the second report, which uses a fixed page size, data appears on two pages because it does not fit on one page.

When you select dynamic page size, all data appears on one page, except when any of the following conditions apply:

- You specify explicit page breaks in the report, such as a page break after a group section.
- You apply page-level security to manage user access to different report pages. Page-level security forces page breaks in the report.
- The data exceeds the maximum page length determined by the operating system or web browser. For example, Windows 2000 and Adobe Acrobat 4.0 display report pages up to 200 inches in height. In this situation, e.Report Designer Professional inserts page breaks to limit the length of each page.

Report output 1: Dynamic Page Height specified

Report output 2: Fixed page size specified

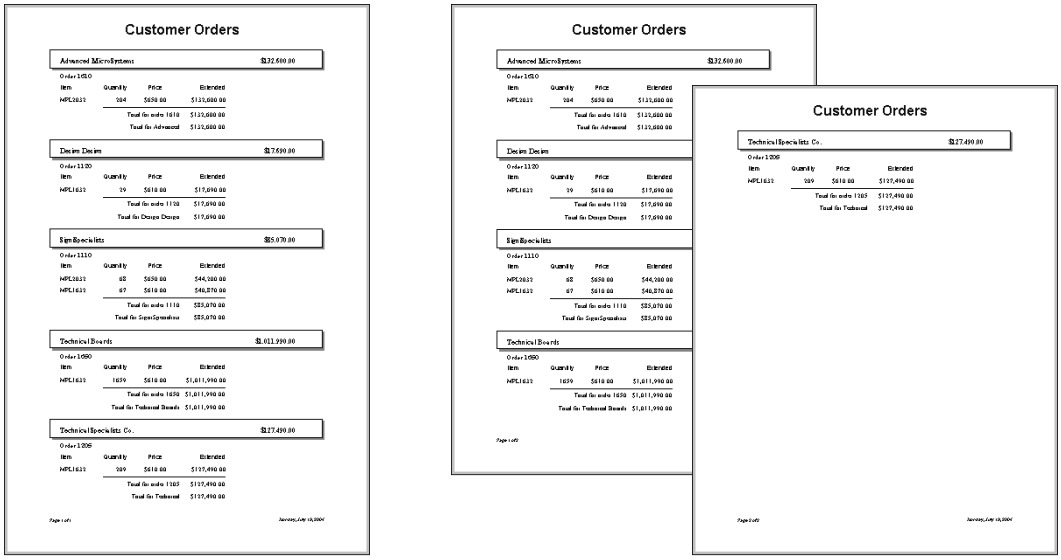


Figure 6-8 Dynamic page size versus fixed page size

Setting page orientation

The initial height and width values that you specify for page size determine the page orientation. If the height is greater than the width, the page orientation is set to portrait. Conversely, changing the page orientation from portrait to landscape reverses the height and width values.

To change the page orientation, choose Tools➛Page Setup and select Portrait or Landscape, as shown in Figure 6-9.

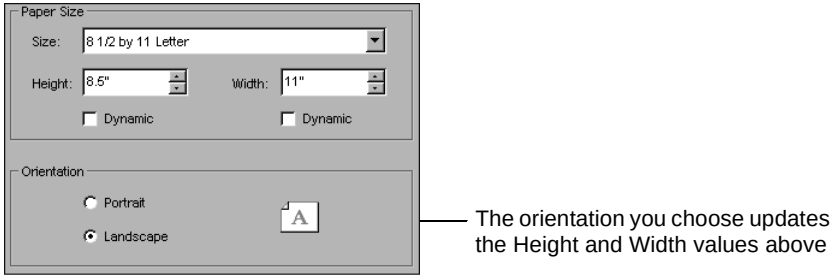


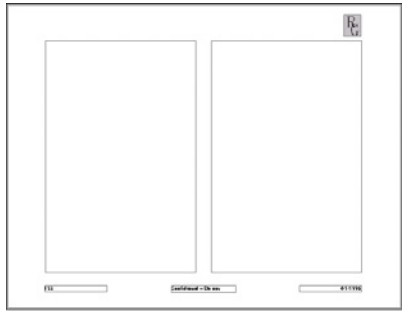
Figure 6-9 Specifying report page orientation

If you change the page orientation during report design, make sure that you make the following adjustments:

- In page layout, adjust the flows and controls to fit within the page.
- In report layout, adjust the frames and controls to fit within the flow.

Figure 6-10 and Figure 6-11 show a page designed for landscape mode, and its appearance after the page orientation is changed to portrait. The second flow no longer fits within the page, and the flows overlap the three controls.

Page designed for landscape mode



Page layout when orientation is changed to portrait

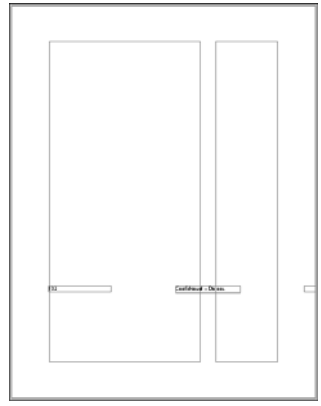
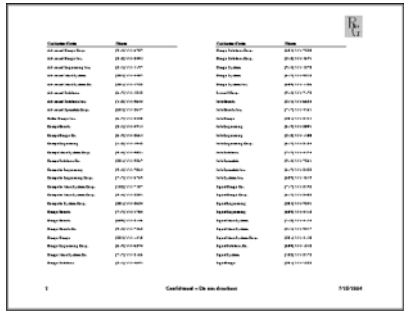


Figure 6-10 Adjusting page orientation and layout

When you generate the report, data in the second flow is truncated. Data also overlaps the text that the three page layout controls display.

Report output 1



Report output 2

Figure 6-11 Adjusting page orientation and layout (continued)

Setting page margins

You can set margins at the top, bottom, left, and right of a page. The margins determine the size of the flow where report data appears. As Figure 6-12 shows, you can place controls in the margins to display images, page numbers, or text.

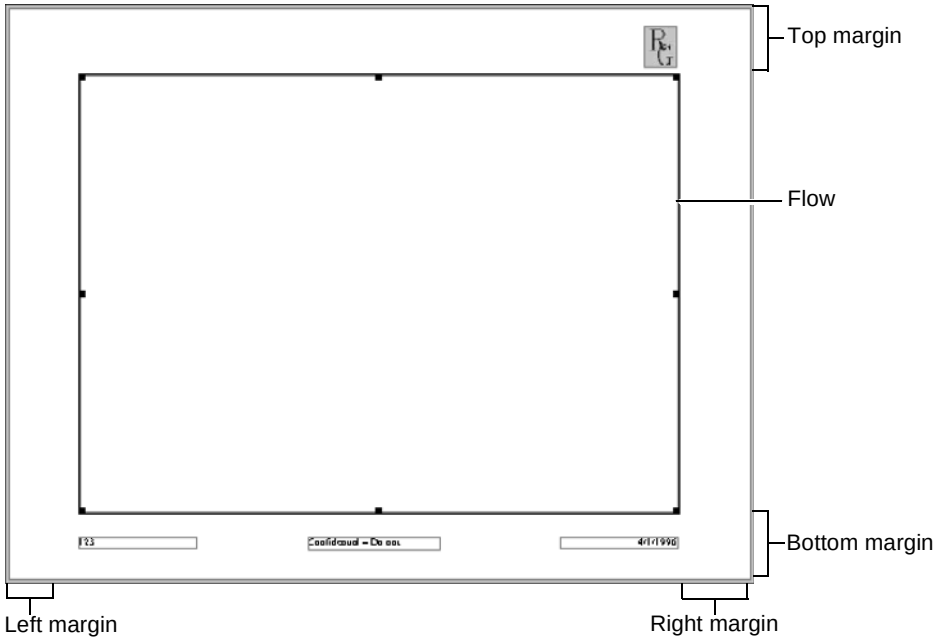


Figure 6-12 Report page margins

To set page margins, choose Tools➛Page Setup and specify values for the margins as shown in Figure 6-13.

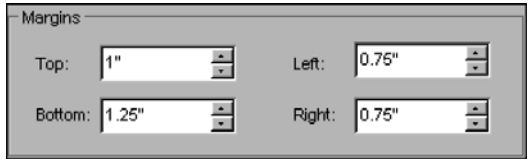


Figure 6-13 Setting page margins

If you adjust the size of a flow visually in the Layout window, the margin values in Page Setup change accordingly.

Understanding the effects of page size on online and printed reports

Although there are always differences between online and printed reports, there are ways to minimize the differences. Decide whether printing or online viewing is more important, design your page layout primarily for that platform, but take both into account. Always test your reports on both platforms.

The following sections describe how fixed and dynamic page size affects online and printed reports.

Comparing online and printed output for reports with fixed page size

When you view a finished report online, you view one page at a time, choosing Next or Prev to page through the report. Viewed at 100%, the page is the size that you specified. When you print the report, however, the printout can be different from the online version, depending on the size of the printer paper. For example, if you specified legal paper size, the online report appears on a 8.5" x 14" page. If you then print the report on legal size paper, the report appears the same in both outputs.

If, however, you print the report on letter size paper, output that fits on one online page prints on two sheets of paper. When you print the report, a 10-page online report becomes a 20-page report. If the page layout contains a page number control, the page numbers are 1 to 10 on both the online and printed report because e.Report Designer Professional generates page numbers when you run the report, not when you print it.

If the report contains a page footer, the footer appears at the bottom of every report page, not at the bottom of every printed page. Similarly, if the report contains a page header, the header appears at the top of every report page, not at the top of every printed page.

These are not limitations in the printed report or software. You can specify the appropriate page size so that page numbers, headers, and footers appear on every printed page. In this case, you optimize the design for print.

Figure 6-14 shows how a report appears online and in print if you set the paper size to legal size but print the report on letter-sized paper. Each online report page prints on two sheets of paper.

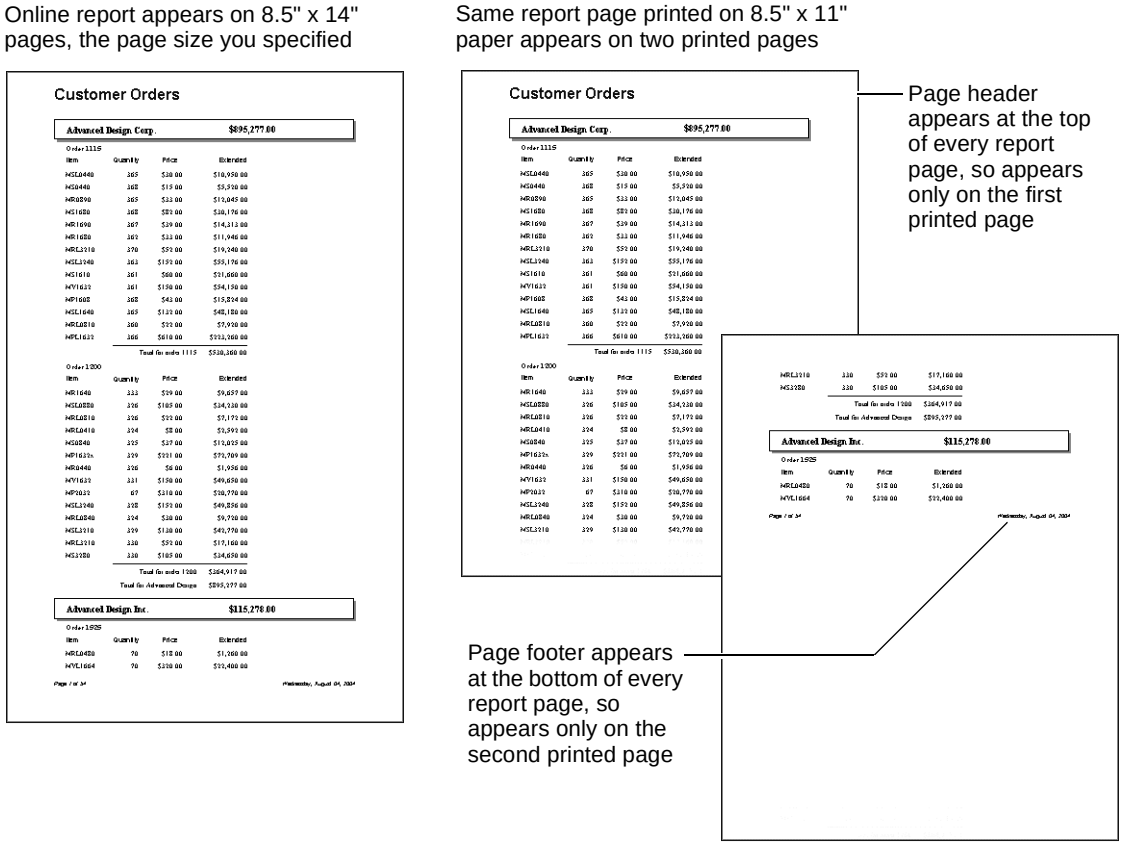


Figure 6-14 Comparing online and printed reports with fixed page size

Comparing online and print output for reports with dynamic page size

If you specify a dynamic page size, the report displays all data on one page. This layout is appropriate for online viewing because you can scroll down or across the page to read the report. When you print the report, however, a lengthy report that fits on a single online page cannot fit on one physical piece of paper. The report requires several print pages.

Consider this major difference between the online and printed output when you design the page layout for a report whose page size you set to dynamic. For example, it does not make sense to include a page number control. Online, page numbers are meaningless because there is only one page. In print, page 1 appears on the last page of a multiple-page printout.

Figure 6-15 shows how a dynamically sized report appears online and in print.

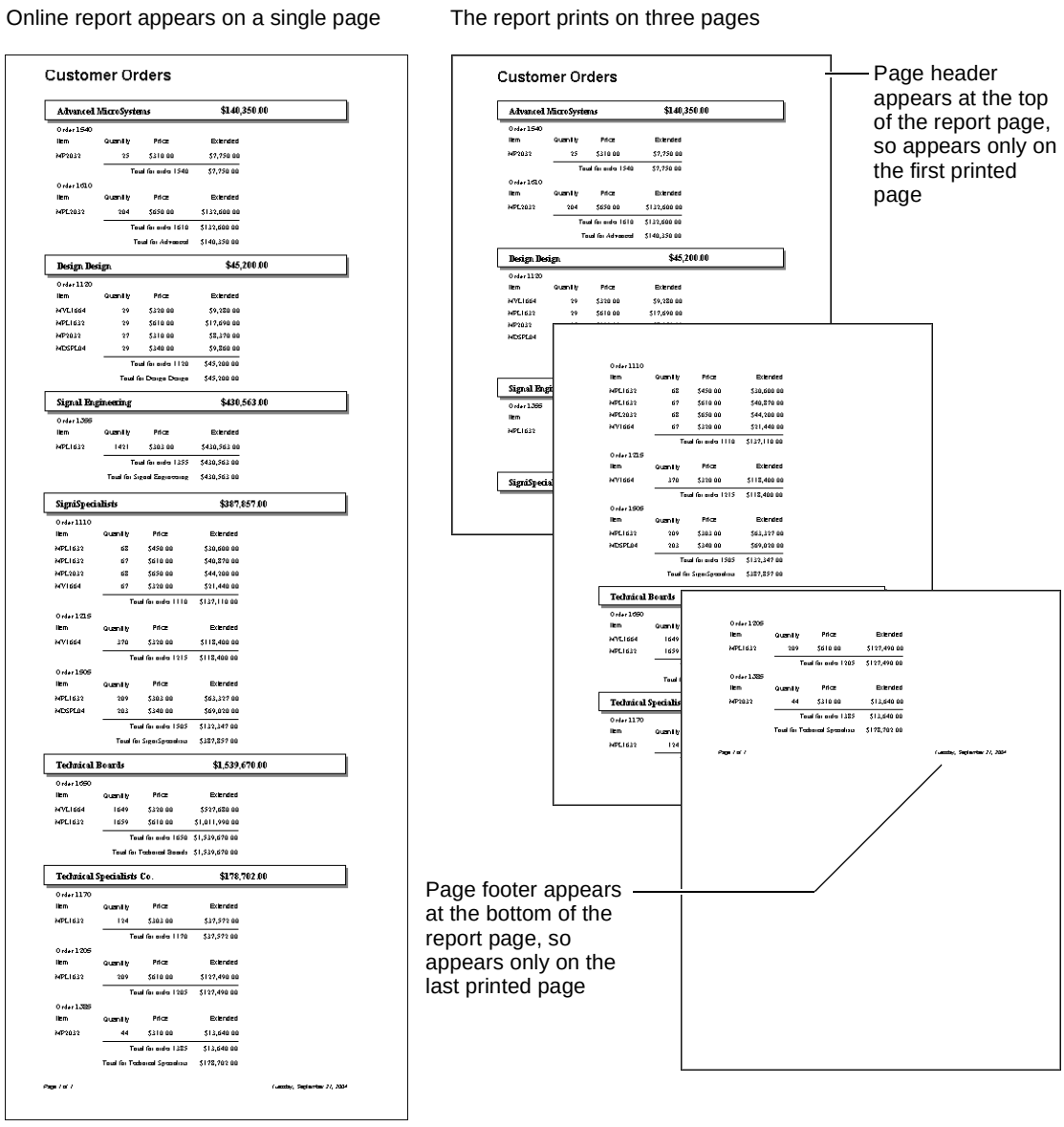


Figure 6-15 Comparing online and printed reports with dynamic page size

Printing a report with large report pages

If you specify a dynamic or large page size, a single report page prints on multiple pages. e.Report Designer Professional handles this task properly using the Printing & PDF—Split Oversize Pages option on Page Setup, as shown in Figure 6-16.

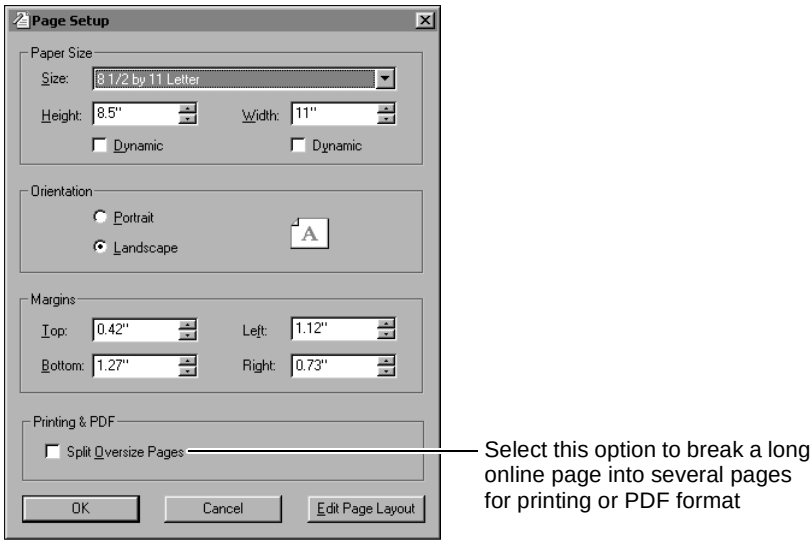


Figure 6-16 Page Setup—Printing & PDF

If you do not select this option, only one page prints and the report data is truncated. Split Oversize Pages has no effect on the online report.

Figure 6-17 shows how the report printouts differ, depending on the Split Oversize Pages setting.

When you use the Split Oversize Pages option, e.Report Designer Professional sets reasonable margins for the printed pages. You can change the default margins by changing the values of the page's Printing and PDF properties, as described in "Specifying margins for the printed report," later in this chapter.

When Split Oversize Pages is not selected, you get only one printed page

When Split Oversize Pages is selected, the report splits onto multiple printed pages

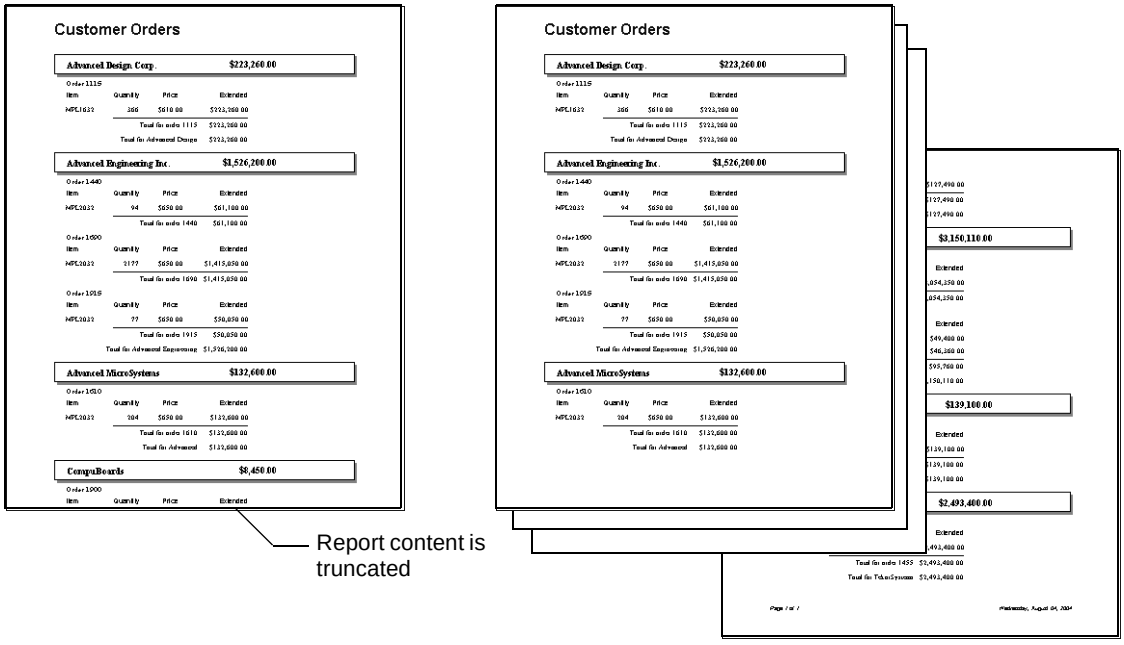


Figure 6-17 Using the Split Oversize Pages option

Specifying margins for the printed report

The margins that you set using Tools➛Page Setup apply to the generated report page, not to the printed page. To ensure that report data does not print at the edges of pages, e.Report Designer Professional adds margins of 0.75 inches to the printed pages, using the following Printing and PDF properties of the page:

- SplitMarginBottom
- SplitMarginTop
- SplitMarginLeft
- SplitMarginRight

To change the margins of the printed pages, set these properties to the values that you want. These properties are relevant only for large online pages that e.Report Designer Professional splits to print on multiple pages.

Figure 6-18 shows the results of setting the SplitMarginBottom and SplitMarginTop properties to 0, for no margins, and 0.75 inches, the default setting. When set to 0, report data prints near the edge of the paper. When set to

0.75, report data prints on more pages because, using the margins, less data fits on each page.

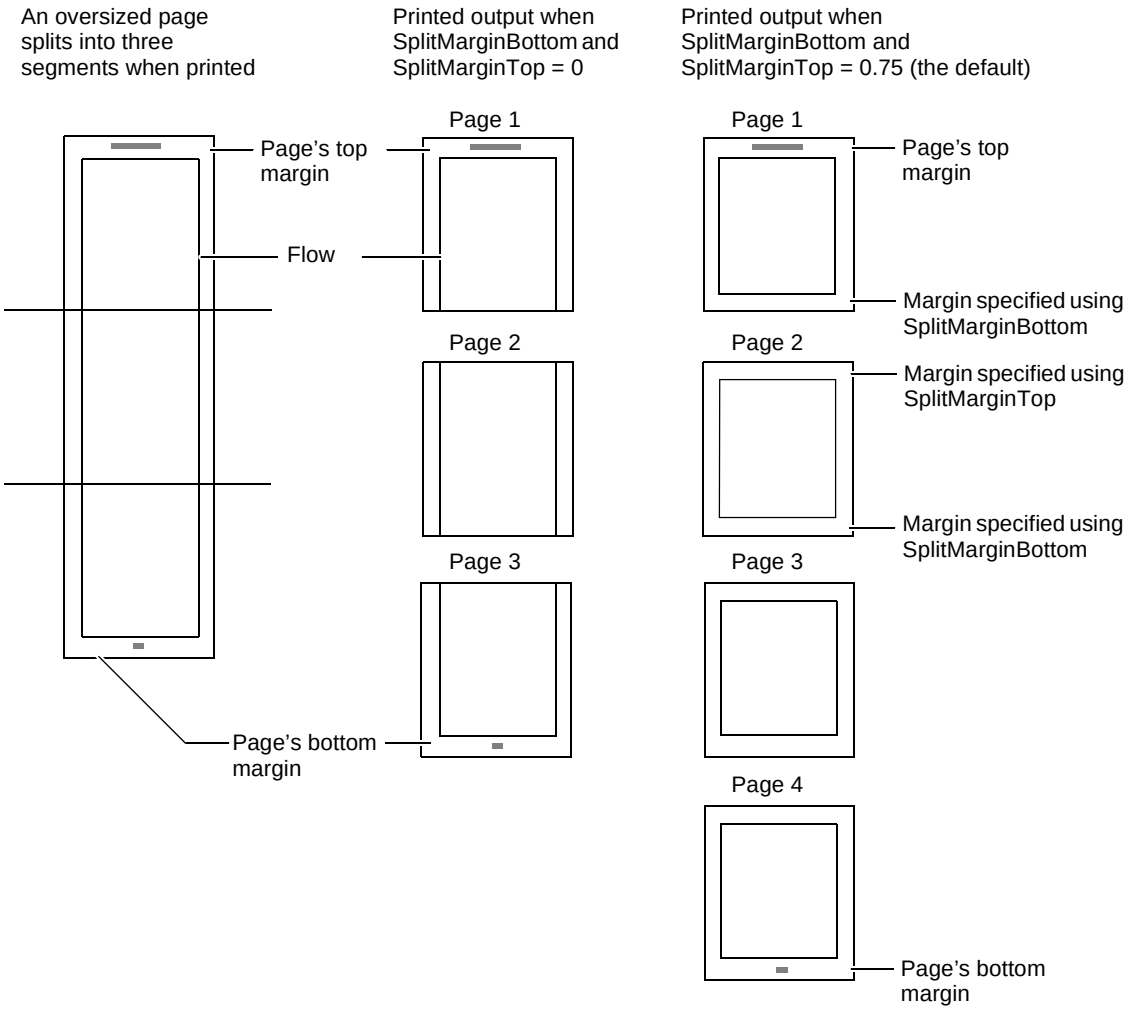


Figure 6-18 Setting SplitMargin properties for printed reports

The SplitMarginLeft and SplitMarginRight properties behave the same as the SplitMarginBottom and SplitMarginTop properties. To see the effects of setting the SplitMarginLeft and SplitMarginRight properties, imagine the previous illustration rotated 90 degrees counterclockwise. In this case, replace Bottom with Right and Top with Left.

Specifying headers and footers

Common header and footer items include page number, report generation date, company name, logo, and copyright or confidentiality statements. You specify page headers and footers by placing controls in the page component. You can place them anywhere on the page except superimposed on a flow. If you superimpose a control on a flow, the control's contents obscure report data. The default page layout includes the following header and footer items:

- A report title placeholder at the top of the page, if you create a report with the Blank Report option
- Page number at the bottom left of the page
- Report-generation date at the bottom right of the page

Figure 6-19 shows the default page layout components for a report that you create using the Blank Report option.

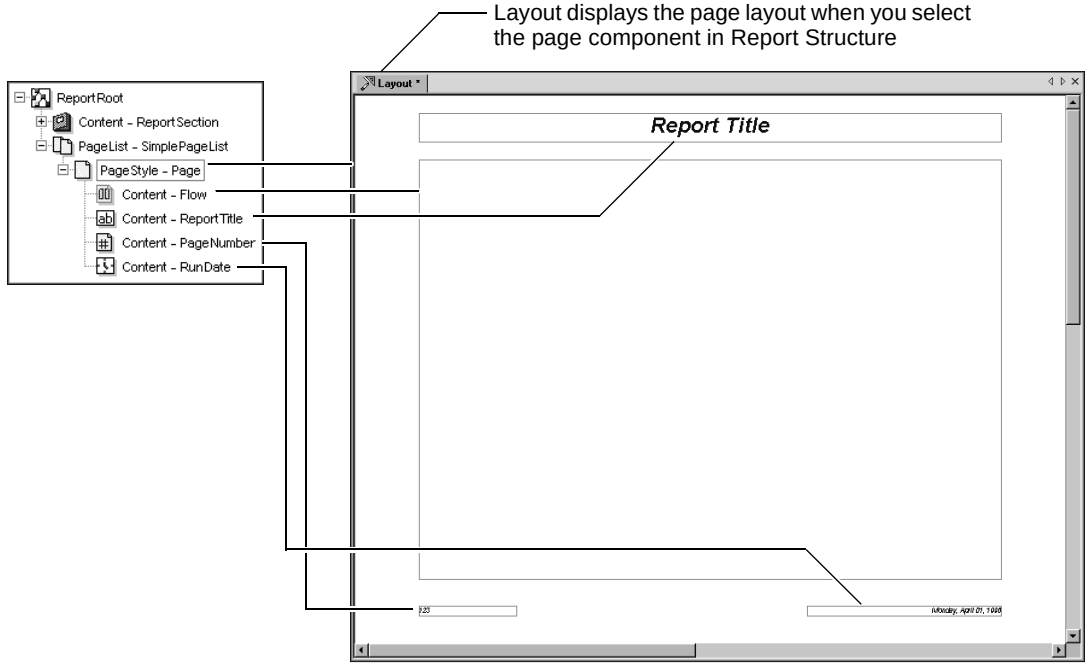


Figure 6-19 Default page layout showing header and footer components

Comparing headers and footers in page layout and report layout

Headers and footers that you specify for a page layout are different from those that you create for sections in a report. Section headers and footers are associated with a particular set of data and they appear within a flow. You create them by placing components in the Before, After, PageHeader, and PageFooter slots, or by using Insert➛Header and Insert➛Footer. In a customer orders report, for example, you can create a section header to repeat a customer name on each page if data about the customer continues on multiple pages.

Figure 6-20 shows both report and page headers and footers as they appear in a finished report. Notice that page header and footer items appear in the same location on each page.

Report headers and footers, on the other hand, appear before and after groups of data, and their location on the page depends on the amount of data in each group.

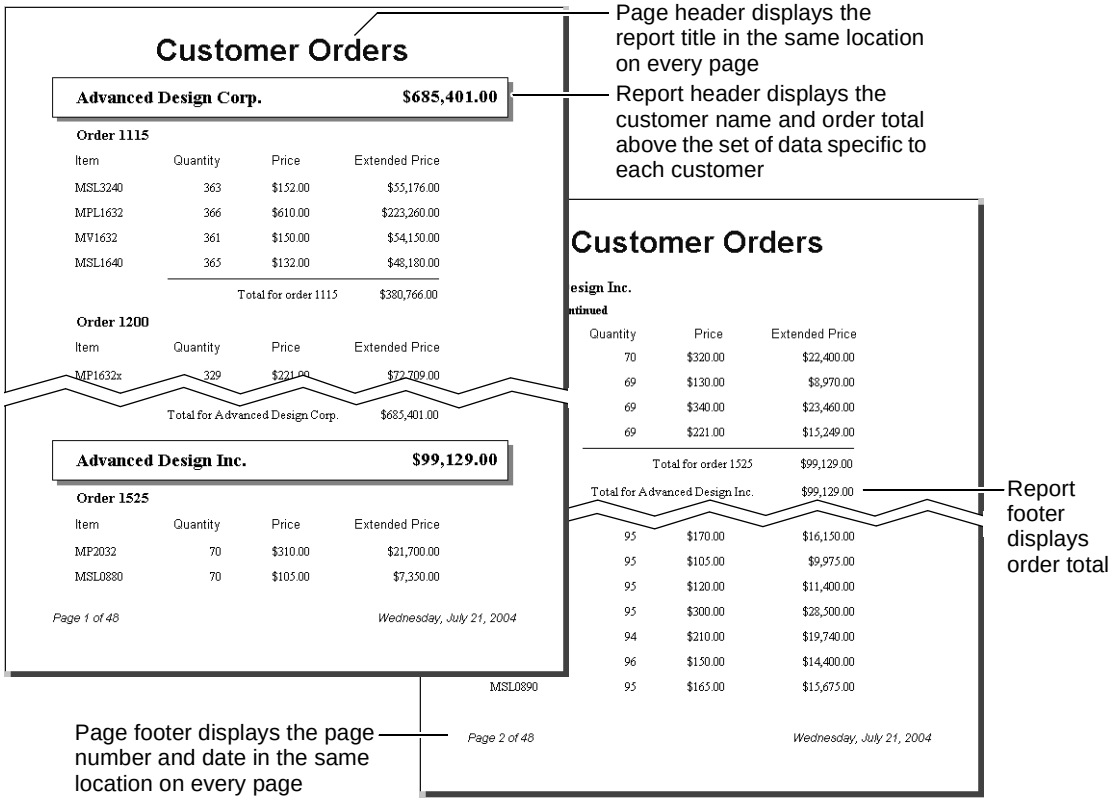


Figure 6-20 Page headers and footers versus report headers and footers

Specifying page numbers

The default page layout uses a page number control to display page numbers at the bottom left of the page in Page *x* of *y* format. The following examples show the default page number when the locale is English (United States) and when the locale is German (Germany):

```
Page 1 of 20
Seite 1 von 20
```

You can change the format by selecting a different value for the control's PageNumberType property. You can also define a custom format.

Using a standard page number format

The PageNumberType property of a page number control provides several page number formats, as shown in Figure 6-21.

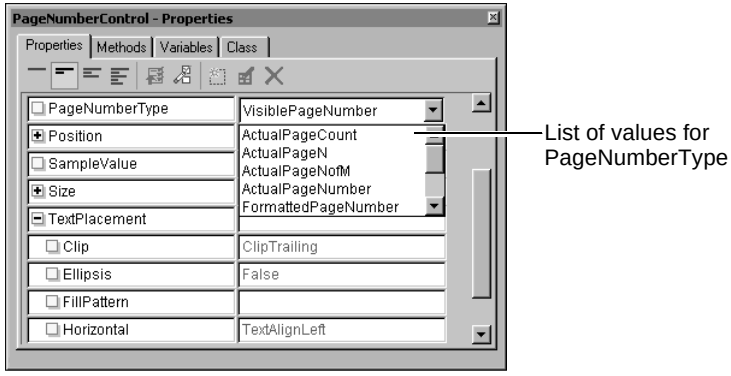


Figure 6-21 Specifying page number format

There are two general categories of page number types that you can choose, actual and visible. Actual and visible page numbers are different only if your report implements the page-level security feature. Using page-level security, you control which pages different users can see. For example, user A can see 4 of 6 pages while user B sees all 6.

Because different users can see different numbers of pages, you can choose to display page numbers that correspond to the full (actual) 6-page report or to the set of pages that are visible to each user. If you choose to display actual page numbers, the page numbers on the report that user A sees are likely to be out of sequence, such as pages 1, 3, 5, and 6. If you choose visible page numbers, the displayed page numbers are 1, 2, 3, and 4.

Within each category of page number type, you can choose a display format, such as 1, Page 1, or Page 1 of 6. For example, selecting ActualPageN or VisiblePageN displays the page number in the following format:

```
Page 1
```

For more information about each PageNumberType property value, see *Programming with Actuate Foundation Classes.*

Using a custom page number format

To display page numbers in a custom format:

- Set the page number control's PageNumberType property to FormattedPageNumber.
- Set the PageNumberFormat property of the page that contains the page number control to the desired format pattern.

The format pattern that you specify for PageNumberFormat must be compatible with the Format( ) function. It can contain any of the keywords or special symbols available to Format( ). For a list and description of valid keywords and symbols, see the Format( ) function in *Programming with Actuate Basic.*

Table 6-1 shows examples of custom format patterns that you can set for the PageNumberFormat property and their effects on page number display.

Table 6-1 Custom page number formats

Format pattern	Results
`"Pg "#`	Pg 1, Pg 2, …, Pg 10, …
`"Section A-"#`	Section A-1, Section A-2, …, Section A-10, …
`(#)`	(1), (2), … (10), …
`#)`	1), 2), …, 10), …
`-#-`	-1-, -2-, …, -10-, …
`0#`	01, 02, …, 10, …, 100, …
`00#`	001, 002, …, 010, …, 100, …

The format pattern that you specify for the PageNumberFormat property of the page applies only if you set the page number control's PageNumberType property to FormattedPageNumber. If PageNumberType is set to any other value, the format in PageNumberFormat has no effect on the page number display.

Specifying a date and time stamp

The default page layout uses a report run date control to display the report-generation date at the bottom right of the page in long date format. The following examples show dates in long date format when the locale is English (United States) and when the locale is German (Germany):

```
Wednesday, July 7, 2004
Mittwoch, 7. Juli 2004
```

You can change the date format by specifying a different value for the control's Format property. You can select one of the predefined format types or define your own date and time format.

Designing background patterns

You can add visual interest to your report by applying colors to flows and pages or creating a watermark that appears in the background of each page.

Adding color and borders to flows and pages

As with other visual components, such as frames and controls, you can add borders and background colors to flows and pages by setting their Border and BackgroundColor properties. Use BackgroundColor principally for online pages. Setting a background color for a printed page can make report data hard to read and uses too much toner or ink.

If you specify colors for controls and frames in the report layout, and for flows and pages in the page layout, check that the results match your design concept. Also, test your report on a black and white printer to see if that output is acceptable if a user prints the page. The colors overlay each other in the following order, starting with the color at the top: control, frame, flow, page. Figure 6-22 shows how a finished report page appears when colors are applied to those components.

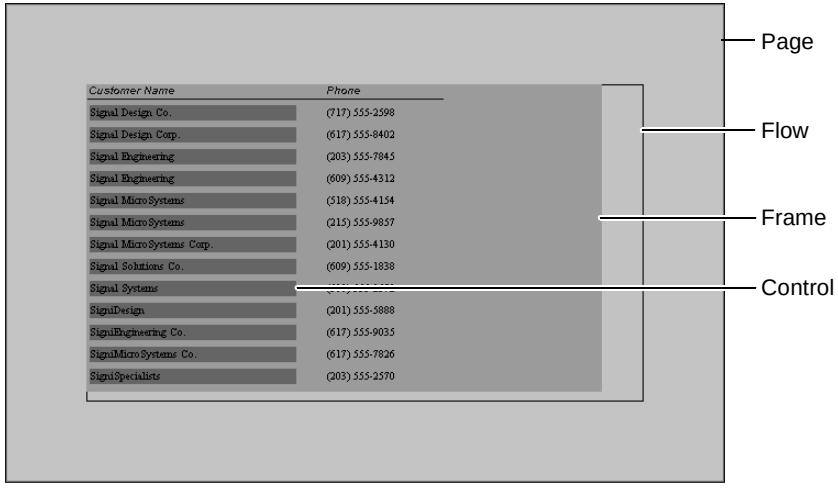


Figure 6-22 Applying color to flows and pages

Adding a watermark

A page can contain an image or text that appears behind the report data. This effect is called a watermark. Government documents or certificates, for example, can contain a watermark that displays a department seal. Corporate reports can display a company logo or a confidentiality statement. Figure 6-23 shows examples of report pages with watermarks.

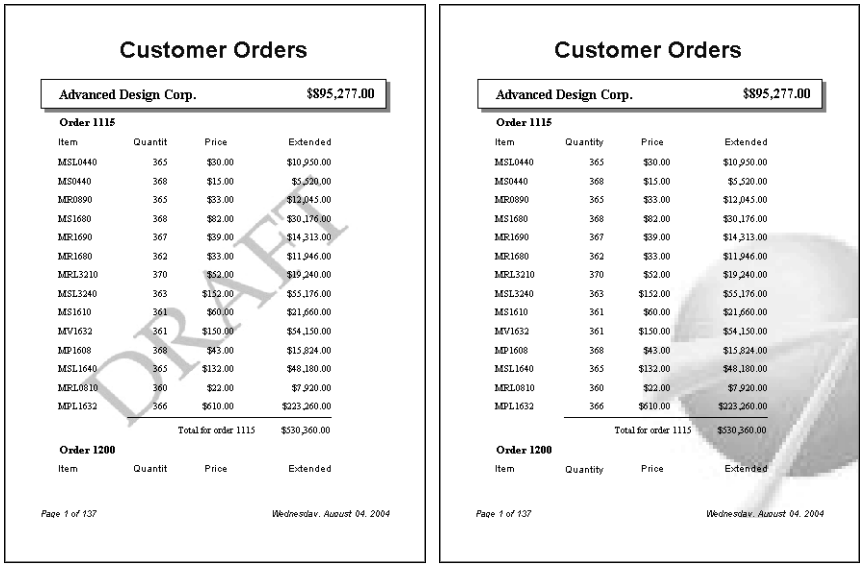


Figure 6-23 Watermark examples

To add a watermark, place an image control in the desired location on the page. Make sure that you use the appropriate image file type for your viewing platform.

To ensure that the image appears behind the report data and shows through the data, you must do the following:

- Send the image to the background by selecting the image, then choosing Format➛Layers➛Send To Back. If the image is in front of the flow, it obscures report data because that data appears in the flow.
- Set the flow's BackgroundColor property to Transparent, the default setting. If the flow has a color, it obscures the image.
- In the report layout, set the BackgroundColor property of the data frame or frames to Transparent, the default setting. If the frame has a color, it obscures the image.

Figure 6-24 shows the page layouts that contain images that display watermarks.

Figure 6-24 Page layouts containing images that display watermarks

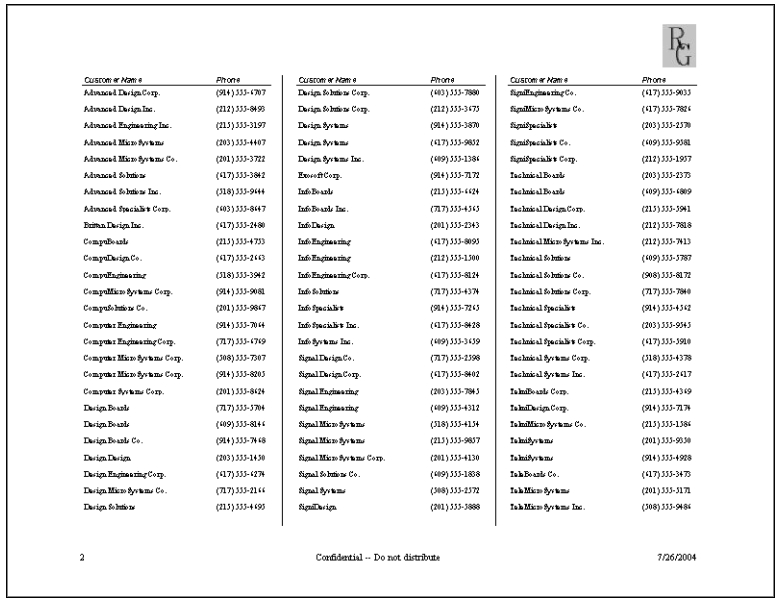
Designing a multicolumn page layout

The default page layout contains one flow to display data in a single column. To display data in multiple-column newspaper style, add multiple flows, one for each column.

This style is appropriate for long lists of names or other data that fit in narrow columns, as shown in Figure 6-25.

To set up a multicolumn page layout, do the following:

- Place the first flow on the page and set it to the desired size.
- To create columns of equal size, copy the first flow to create additional flows. To copy a flow, press Ctrl and drag the flow to another location.
- To align the flows or balance the spacing between them, use the Format menu commands.
- To distribute report data evenly among the flows, set the page's Pagination➤BalanceFlows property to True. If this property is set to False (the default), data fills each flow before continuing on the next flow. Figure 6-26 shows how report output differs when BalanceFlows is set to False and when it is set to True.

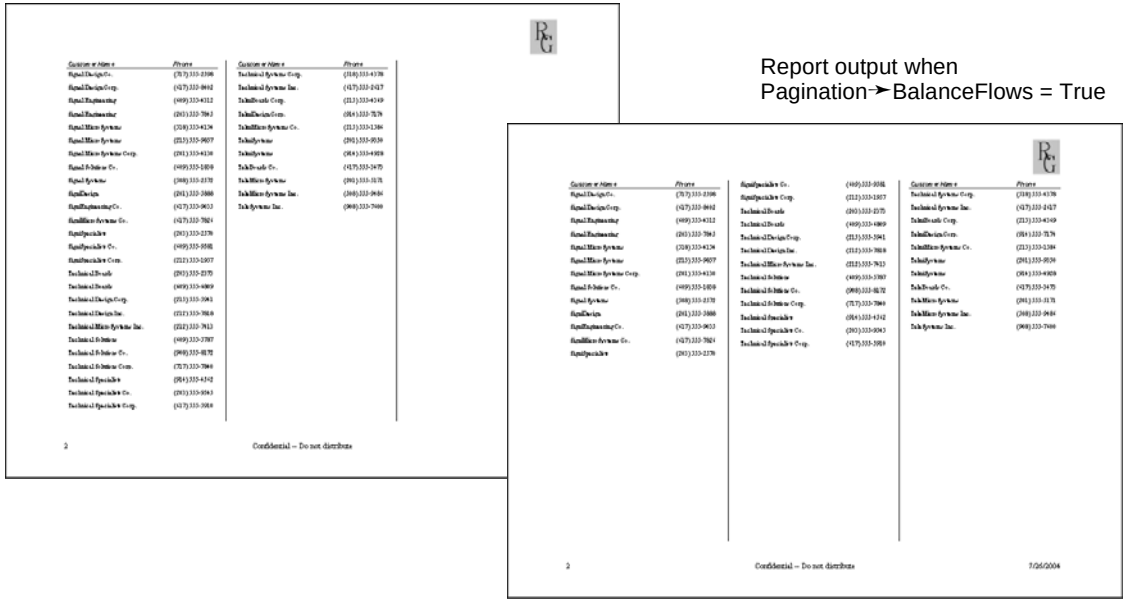


Customer Name	Phone	Customer Name	Phone	Customer Name	Phone
Advanced Design Corp.	(914) 333-4707	Design Solutions Corp.	(403) 333-7880	SignalEngineering Co.	(417) 333-9033
Advanced Design Inc.	(212) 333-8493	Design Solutions Corp.	(212) 333-3473	SignalMicro Systems Co.	(417) 333-7824
Advanced Engineering Inc.	(215) 333-3197	Design Systems	(914) 333-3870	SignalSpecialties	(203) 333-2370
Advanced Micro Systems	(203) 333-4407	Design Systems	(417) 333-9832	SignalSpecialties Co.	(409) 333-9381
Advanced Micro Systems Co.	(201) 333-3722	Design Systems Inc.	(409) 333-1384	SignalSpecialties Corp.	(212) 333-1957
Advanced Solutions	(417) 333-3862	EnviroFit Corp.	(914) 333-7172	Technical Boards	(203) 333-2373
Advanced Solutions Inc.	(318) 333-9444	Info Boards	(215) 333-4424	Technical Boards	(409) 333-4869
Advanced Specialties Corp.	(403) 333-8447	Info Boards Inc.	(717) 333-4343	Technical Design Corp.	(215) 333-3961
Britton Design Inc.	(417) 333-2480	Info Design	(201) 333-2343	Technical Design Inc.	(212) 333-7818
CompuBoards	(215) 333-4733	Info Engineering	(417) 333-8693	Technical Micro Systems Inc.	(212) 333-7413
CompuDesign Co.	(417) 333-2443	Info Engineering	(212) 333-1300	Technical Solutions	(409) 333-3787
CompuEngineering	(318) 333-3962	Info Engineering Corp.	(417) 333-8124	Technical Solutions Co.	(908) 333-8172
CompuMicro Systems Corp.	(914) 333-9081	Info Solutions	(717) 333-4378	Technical Solutions Corp.	(717) 333-7860
CompuSolutions Co.	(201) 333-9847	Info Specialties	(914) 333-7243	Technical Specialties	(914) 333-4342
Computer Engineering	(914) 333-7044	Info Specialties Inc.	(417) 333-8428	Technical Specialties Co.	(203) 333-9343
Computer Engineering Corp.	(717) 333-4749	Info Systems Inc.	(409) 333-3439	Technical Specialties Corp.	(417) 333-3910
Computer Micro Systems Corp.	(308) 333-7307	Signal Design Co.	(717) 333-2598	Technical Systems Corp.	(318) 333-4378
Computer Micro Systems Corp.	(914) 333-8203	Signal Design Corp.	(417) 333-8402	Technical Systems Inc.	(417) 333-2417
Computer Systems Corp.	(201) 333-8424	Signal Engineering	(203) 333-7845	TalkBoards Corp.	(215) 333-4349
Design Boards	(717) 333-3704	Signal Engineering	(409) 333-4312	TalkDesign Corp.	(914) 333-7178
Design Boards	(409) 333-8144	Signal Micro Systems	(318) 333-4134	TalkMicro Systems Co.	(215) 333-1384
Design Boards Co.	(914) 333-7448	Signal Micro Systems	(215) 333-9857	TalkSystems	(201) 333-5030
Design Design	(203) 333-1430	Signal Micro Systems Corp.	(201) 333-4130	TalkSystems	(914) 333-4908
Design Engineering Corp.	(417) 333-4278	Signal Solutions Co.	(409) 333-1838	Talk Boards Co.	(417) 333-3473
Design Micro Systems Co.	(717) 333-2144	Signal Systems	(308) 333-2372	Talk Micro Systems	(201) 333-3171
Design Solutions	(215) 333-4493	SignalDesign	(201) 333-3888	Talk Micro Systems Inc.	(308) 333-9484

2 Confidential -- Do not distribute 7/26/2004

Figure 6-25 Multicolumn page layout example

Report output when Pagination➛BalanceFlows = False



Report output when Pagination➛BalanceFlows = True

Figure 6-26 Setting the BalanceFlows property in a multicolumn page layout

Designing multiple page layouts

For most reports, one layout for all pages is sufficient, particularly if the report is for online viewing.

You can also create multiple page layouts for a report, for example, to achieve the following effects:

- One layout for left pages and another for right pages. Use this style for reports that you want to print and bind in book fashion. You can design a symmetrical look by placing the page number on the right side of a right page and on the left side of a left page.
- One layout for the report's first page and another for the rest of the pages. Use this style to design a cover page followed by report data. You can also use this style to display summary data on the first page and detailed information on the rest of the pages.
- A different layout for each subreport or section in a report. For example, if your report contains two subreports, a customer listing report and a sales detail report, you can create a multicolumn layout for the customer listing report, and a single-column layout for the sales detail report.

Creating a left and right page layout

To create a left and right page layout:

- Replace the default simple page list component with left and right page list.
- Add two pages to the page list and at least one flow in each page.
- Design each page with different layouts.

Figure 6-30 shows sample page layouts for a left and right page, and Figure 6-31 shows pages of the finished report.

How to set up a left-right page layout

The instructions in this section assume that you have created a new report that uses the default page layout.

1. In Report Structure, expand PageList—SimplePageList. This default page list component specifies one layout for all pages.
2. Expand PageStyle—Page and view the page layout as shown in Figure 6-27.

   If you want to re-use the default page, drag it to the Scratch Pad.
3. Delete PageList—SimplePageList.

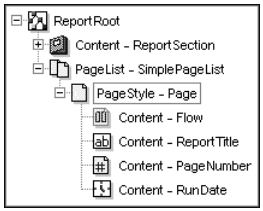


Figure 6-27 Default page structure

4 Drag the page list component from Toolbox—Pages and drop it in the PageList slot. Select Component appears as shown in Figure 6-28.

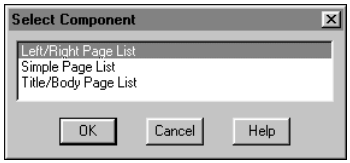


Figure 6-28 Selecting the left/right page list component

5 Select Left and Right Page List. Choose OK. The page list appears in Report Structure. It contains two slots, LeftPage and RightPage.

6 Drag a page component from Toolbox—Pages and drop it in the LeftPage slot. The page component appears in Report Structure. It contains a Content slot.

7 Drag a flow from Toolbox—Pages and drop it in Content.

8 Drag a page from Toolbox—Pages and drop it in the page list's RightPage slot. A second page appears in Report Structure.

9 Drag a flow from Toolbox—Pages and drop it in the Content slot of RightPage.

You have set up the basic structure for a two-page layout, as shown in Figure 6-29.

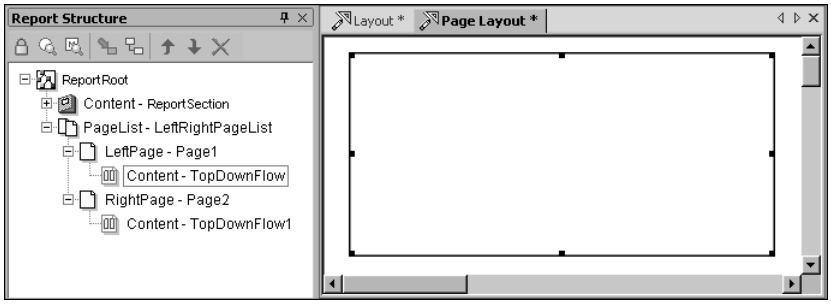


Figure 6-29 Sample report structure containing left/right page layout

To finish designing each page, you can change the page size or orientation, resize the flow, add additional flows, or add controls to display header and footer information. These tasks are described earlier in this chapter.

When you use Tools➤Page Setup, remember to first select the page whose attributes you want to set. If you do not select either page, e.Report Designer Professional applies the page settings to both pages.

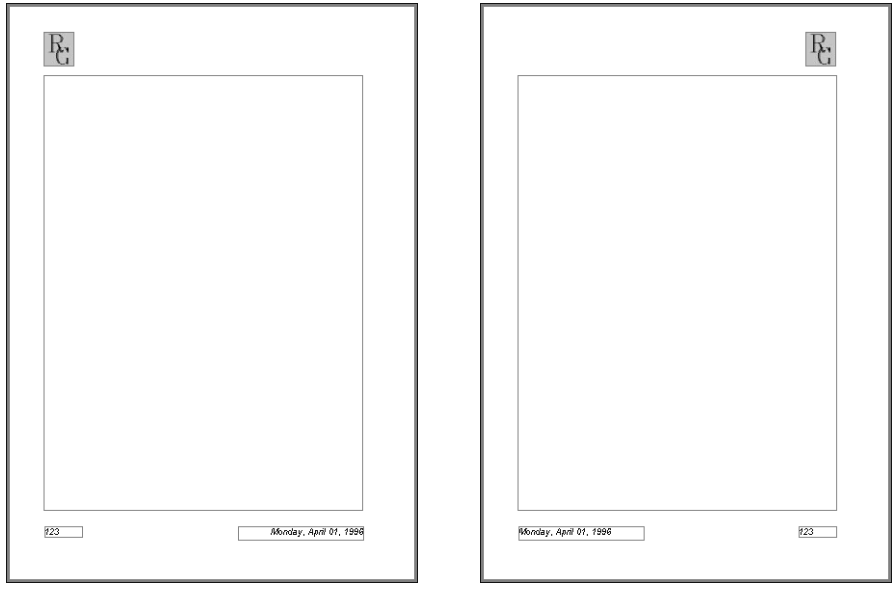


Figure 6-30 Left and right page layouts

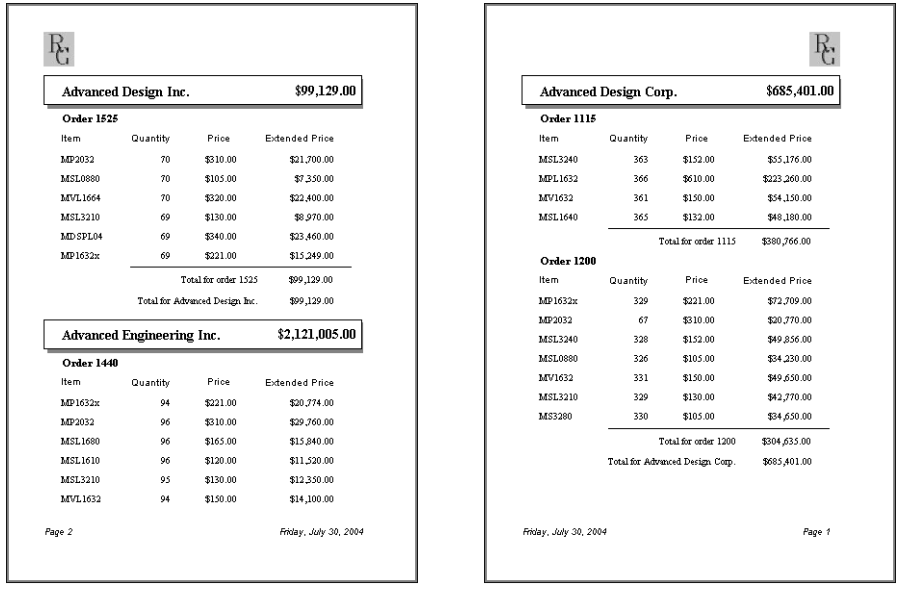


Figure 6-31 Left and right pages of the finished report

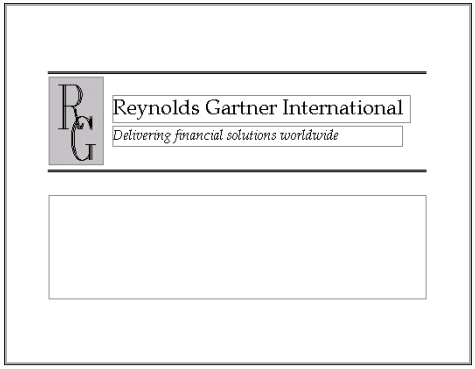
Creating a title and body page layout

To create a title page and body page layout:

- Replace the default simple page list component with title/body page list.
- Add two pages to the page list and at least one flow in each page.
- Design each page with different layouts.

Figure 6-32 shows sample page layouts for a title and body page, and Figure 6-33 shows pages of the finished report.

Page layout for a report's title page



Page layout for a report's body pages

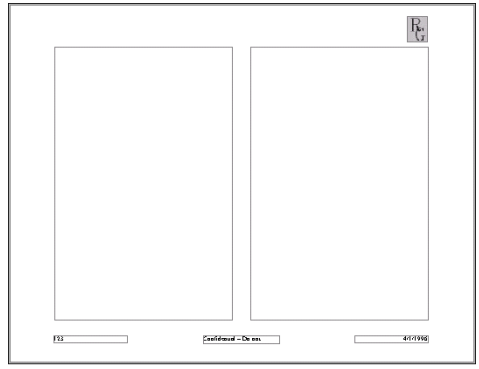


Figure 6-32 Sample title and body page layouts

First page of the finished report

Second page of the finished report

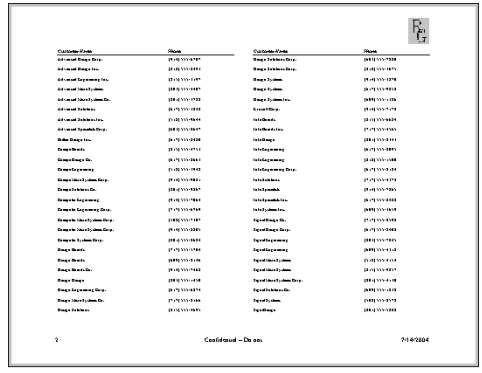


Figure 6-33 Sample title and body finished report page layouts

When you use the title and body page style, ensure that the data in the report layout fits the page layout. If you want only a particular data frame, such as a report title frame, to appear on the first page, specify a page break after that frame. If you do not specify a page break, data from subsequent frames fills all available space on the first page.

Figure 6-34 shows the report layout for the report shown in the previous illustration.

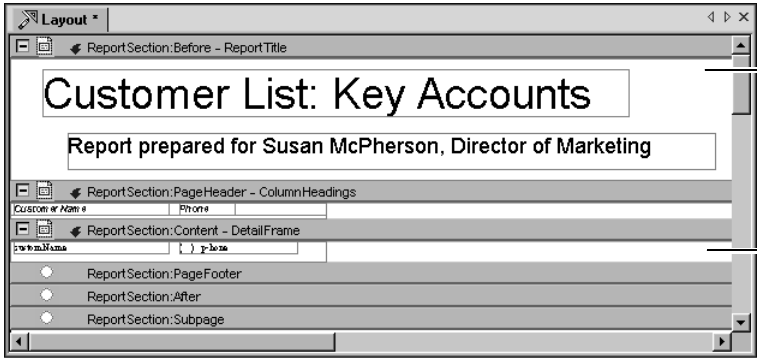


Figure 6-34 Report layout containing title and body pages

If you use the default property settings for the report title frame, the output appears as shown in Figure 6-35.

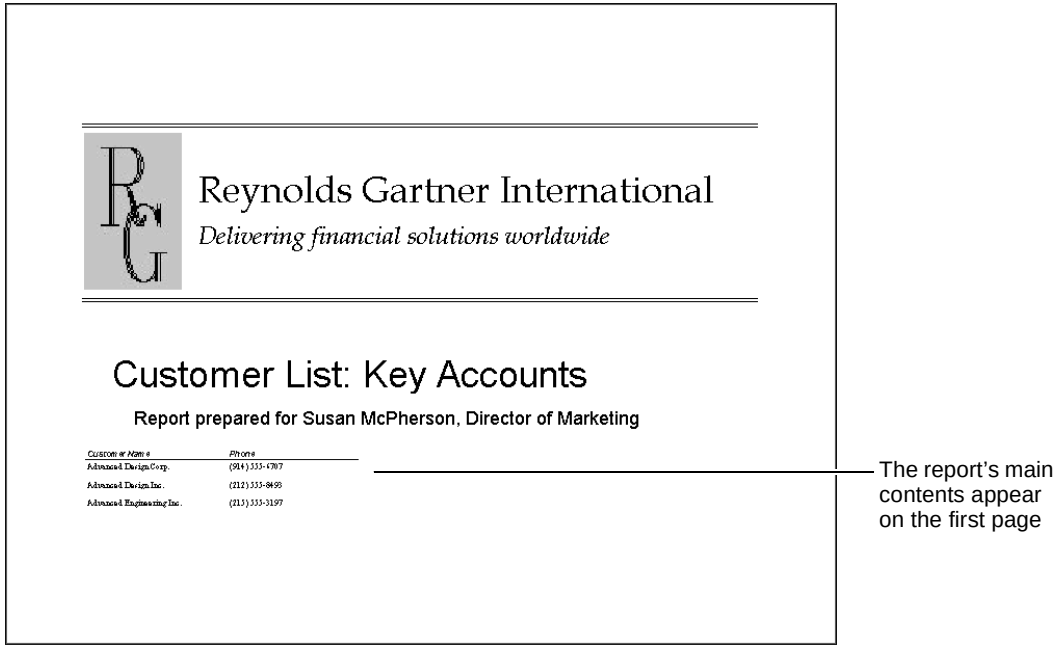


Figure 6-35 Report output using default property settings for title frame

The report's main contents appear on the first page rather than on the second page. To start the main content on the second page, specify a page break after the report title.

How to set a page break after a frame

1 In the report layout, right-click the report title frame and choose Properties. The Properties page for the frame appears.

2 Set the frame's Pagination➛PageBreakAfter property to True.

How to set up a title and body page layout

The instructions in this section assume that you have created a new report that uses the default page layout.

1 In Report Structure, expand PageList—SimplePageList. This default page list component specifies one layout for all pages.

2 Expand PageStyle—Page and view the page layout, as shown in Figure 6-36.

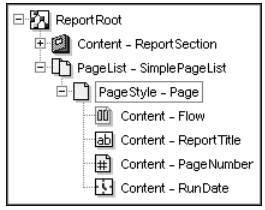


Figure 6-36 Setting up title and body page layouts

If you want to reuse the default page, drag it from the list and drop in on Scratch Pad.

3 Delete PageList—SimplePageList.

4 Drag a page list component from Toolbox—Pages and drop it in the topmost report component. Select Component appears, as shown in Figure 6-37.

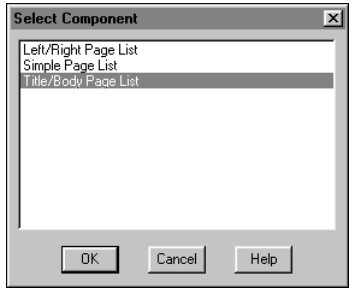


Figure 6-37 Selecting page list components

5 Select Title and Body Page List. Choose OK. The page list appears in Report Structure. It contains two slots, TitlePage and BodyPage.

6 Drag a page from Toolbox—Pages and drop it in the TitlePage slot. The page component appears in Report Structure. It contains a Content slot.

7 Drag a flow from Toolbox—Pages and drop it in Content.

8 Drag a page from Toolbox—Pages and drop it in the page list's BodyPage slot. A second page appears in Report Structure.

9 Drag a flow from Toolbox—Pages and drop it in the Content slot of BodyPage. You have set up the basic structure for a two-page layout, as shown in Figure 6-38.

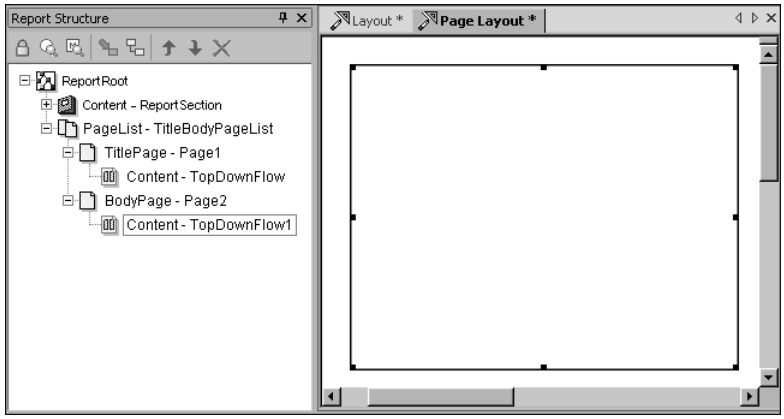


Figure 6-38 Basic structure of a two-page layout

To finish designing each page, you can change the page size or orientation, resize the flow, add additional flows, or add controls to display header and footer information.

When you use Tools➤Page Setup, remember to first select the page for which you want to set properties. If you do not select a page, e.Report Designer Professional applies the page settings to both pages.

Chapter

7

Understanding charting concepts

This chapter contains the following topics:

- About Actuate charts
- Recognizing the parts of a chart
- Understanding types of charts
- Understanding chart layers
- Understanding chart data
- Using a chart hyperlink

About Actuate charts

A chart is a graphic layout that displays data or the relationships between sets of data. Charts support the display of complex data in an easy-to-assimilate and familiar format.

To create a chart, you use Chart Builder, a wizard that guides you through chart creation. You use Chart Builder to select a chart type and sub-type, specify chart data, and set chart formatting options. After creating a chart, you can use Chart Builder to add parts to a chart or to modify chart formatting.

Actuate charts are Unicode compliant, supporting data display in multiple languages and across locales. Charts render identically on Windows and UNIX platforms. A chart can collect data from one or more rows.

You create Actuate charts using the AcChart class. For compatibility with earlier releases, Actuate continues to support the AcChart6 class, which is a version of the AcChart class used in Releases 6 through 7 Service Pack 1.

In releases prior to Release 8 Server Pack 1, charts in Actuate Basic reports were rendered as PNG images in some output formats. In Release 8 Service Pack 1 and later, charts in Actuate Basic reports are rendered as PNG images in all output formats.

For information about the AcChart6 class, refer to the documentation for that release. For information about accessing documentation from an earlier release, refer to the product release notes.

The Actuate Foundation Classes offer some additional chart features not described in this document. For more information about using the Actuate Foundation Classes, see *Programming with Actuate Foundation Classes.*

Recognizing the parts of a chart

A chart displays data as one or more sets of points. A series is a set of related points. Every chart contains at least one series. A chart can also group points into categories. For example, a chart that shows yearly sales can use sales representative IDs as series and months as categories. A different sales chart could use one series to show the total sales each month.

Figure 7-1 shows the parts of a basic bar chart. The chart shows 2003 sales totals and uses item types as series and sales regions as categories.

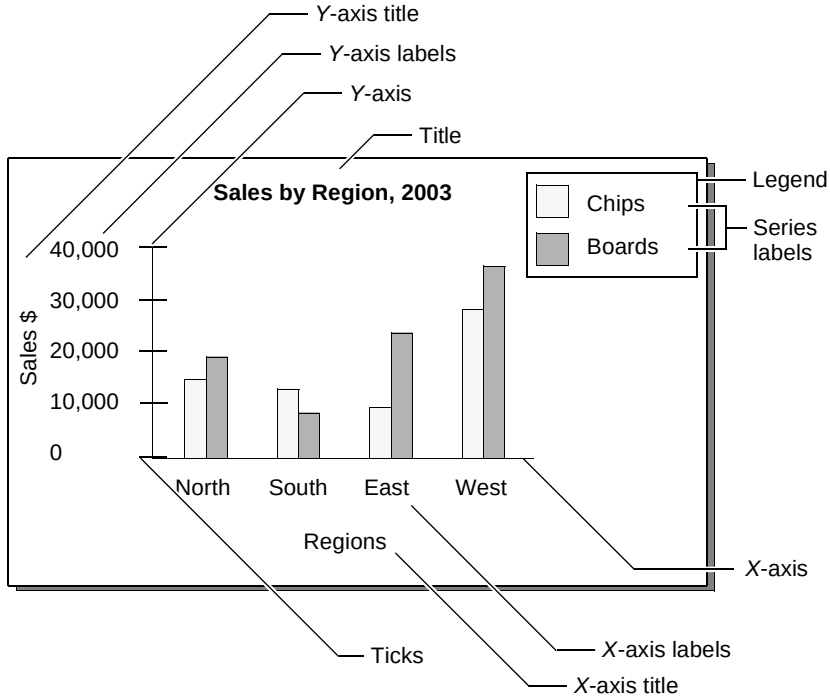


Figure 7-1 Parts of a basic bar chart

e.Report Designer Professional supports the chart parts described in Table 7-1.

Table 7-1 Chart components

Part of chart	Description
Drop line	A line that leads from a data point to the *x*-axis.
Grid line	A horizontal or vertical line that extends an axis tick across a chart's plot area.
High-low line	A vertical line that connects the highest value in a category to the lowest value in a category. For example, in a stock chart, you can use high-low lines to connect the highest and lowest prices in each trading day.
Plot	The area of a chart that displays chart data. In a chart without depth, the area between the *x*-axis and the *y*-axis. In a chart with depth, the area between the chart walls and the chart floor.
Point label	A label that identifies a data point.
Tick	A mark that identifies a regular segment on an axis.

(continues)

Table 7-1 Chart components (continued)

Part of chart	Description
Up or down bar	In a line chart, a bar used to connect the first series in a category to the last series in the same category. In a candlestick stock chart, a bar used to connect the open series in a category to the close series in the same category. Typically shows a stock's price at market opening and closing.

Understanding types of charts

Actuate supports the following types of charts, as shown in Figure 7-2:

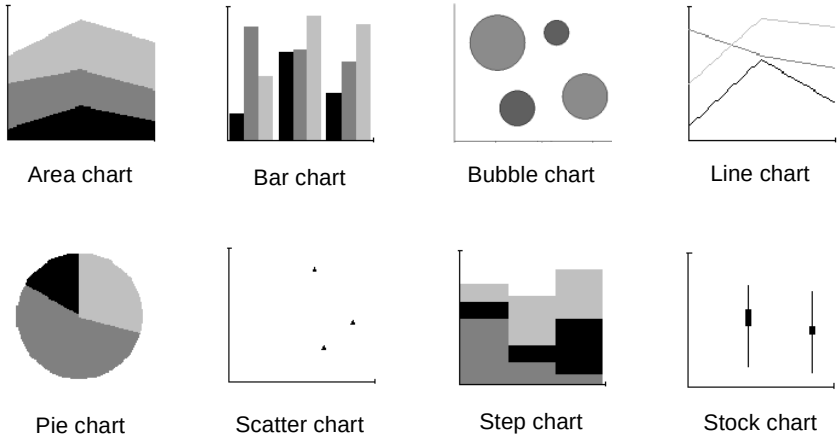


Figure 7-2 Supported types of charts

e.Report Designer Professional offers several sub-types for each basic chart type. For example, Figure 7-3 shows the three step chart sub-types.

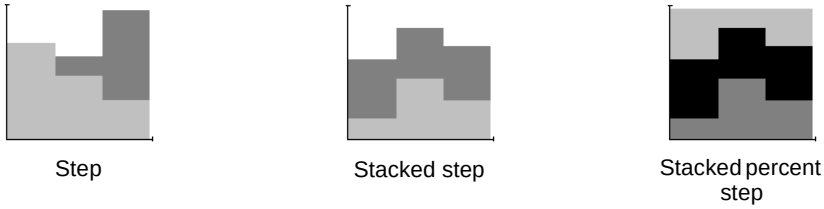


Figure 7-3 Three step chart sub types

The kind of information that you present in a chart determines the chart type appropriate for your report.

About area charts

An area chart displays data values as a set of points connected by lines. If you include several series on an area chart, the chart displays filled areas that overlap.

In the two-dimensional area chart shown in Figure 7-4, the North, South, and East regions are the categories. The dollar values of the sales of CPUs, boards, and chips are the series.

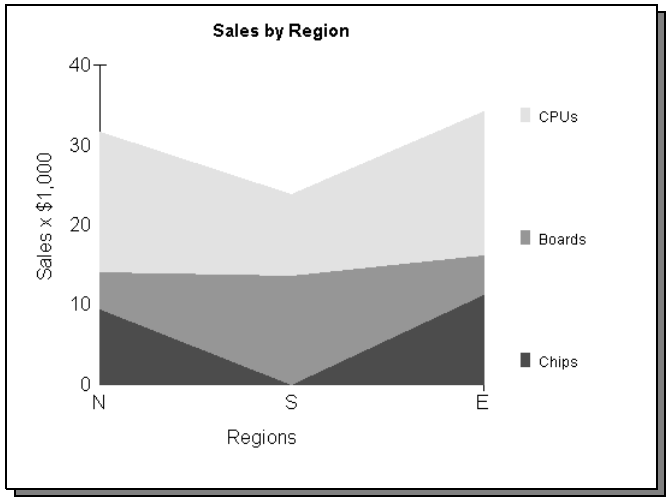


Figure 7-4 Area chart

About bar charts

A bar chart typically displays data values as a set of vertical columns. You can also flip the axes of a bar chart to display data values as a set of horizontal bars.

Figure 7-5 shows a bar chart showing global internet usage in the United States by connection speed and region.

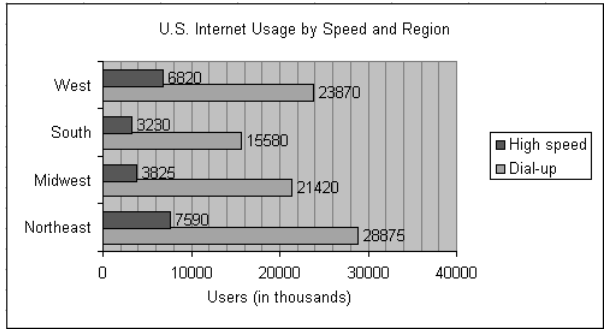


Figure 7-5 Bar chart

The chart series are High speed and Dial-up. The chart categories are West, South, Midwest, and Northeast. The chart axes are flipped to display bars horizontally. Bars within the categories overlap slightly because the report developer set the bar overlap percent to a positive number.

About bubble charts

A bubble chart displays data points as bubbles. A bubble chart allows specific values to be visually represented by different bubble sizes, which is useful when representing financial data.

Bubble charts use three different sets of values for each data point. The chart shows two numbers as XY coordinates and a third number as the size of each data point, or bubble.

Figure 7-6 shows a bubble chart representing wireless internet usage for three world regions. The data points for each series are the region's population, the number of internet users in the region, and the number of wireless internet users in the region.

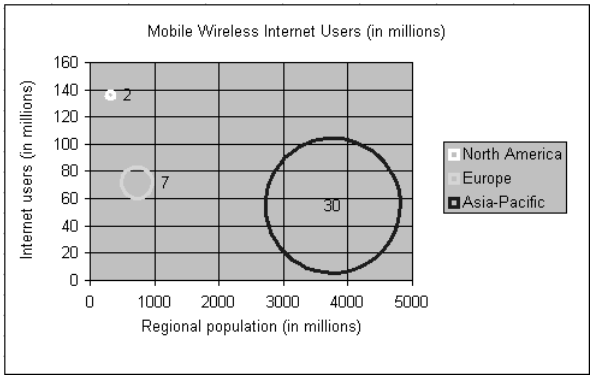


Figure 7-6 Bubble chart example

About line charts

A line chart displays data as a sequence of points connected by a line. You can also add stock data, such as high-low lines and up or down bars, to line charts.

Figure 7-7 shows a line chart that uses two series and four categories. The series are the dollar value of sales of chips and boards. The categories are quarters of the year.

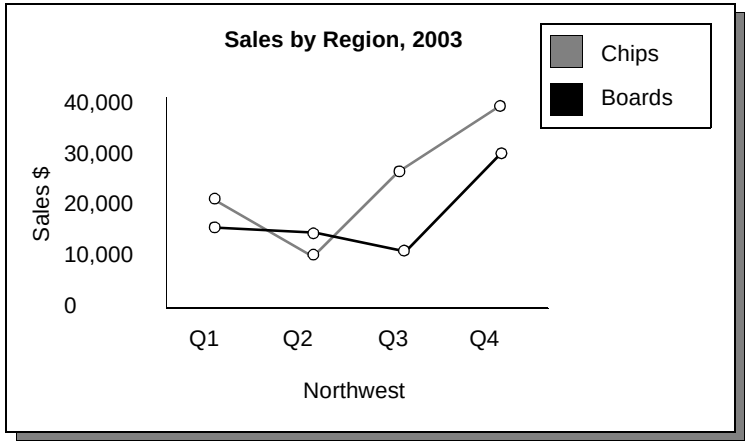


Figure 7-7 Line chart

Figure 7-8 shows the same chart created using a data source localized in Korean.

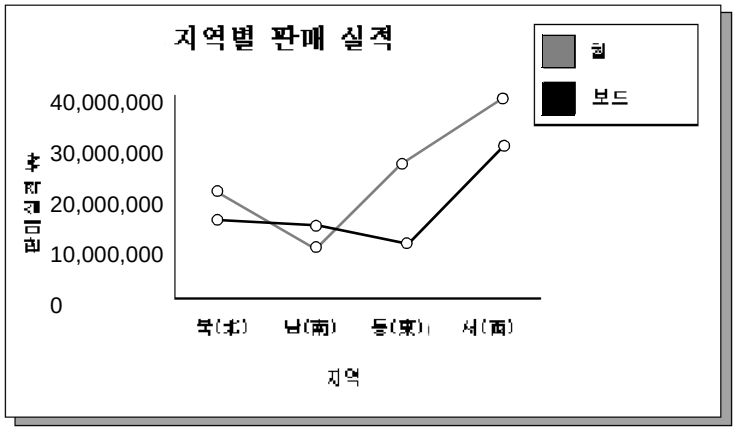


Figure 7-8 The line chart in Figure 7-7, using a Korean-localized data source

About pie charts

Use a pie chart to show data as part of a total. A pie chart displays values as sectors of the pie.

For example, Figure 7-9 shows a pie chart displaying price quotes for a range of inventory items. Each sector shows the total price quotes for an item. The entire chart represents the total for all the price quotes.

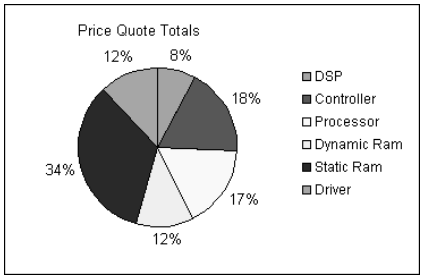


Figure 7-9 Pie charts

Figure 7-10 shows the same pie chart created using a data source localized in Simplified Chinese.

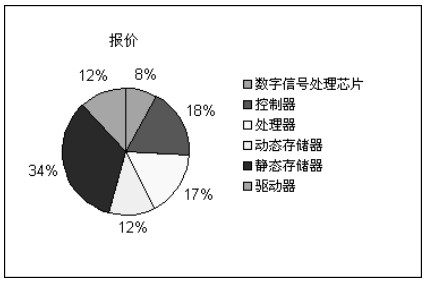


Figure 7-10 Pie chart in Chinese

About scatter charts

A scatter chart shows data as points. In a scatter chart, both axes show values.

The scatter chart shown in Figure 7-11 compares two sets of numerical data. The *x*-axis displays the top speed of certain sports cars. The *y*-axis displays the sports car prices in dollar amounts.

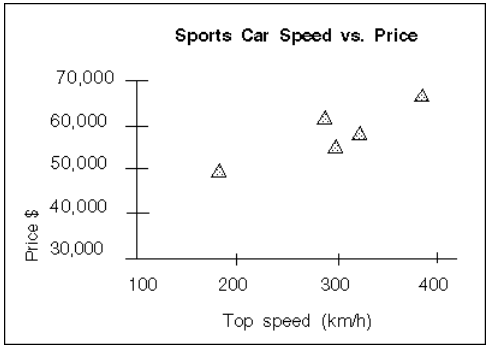


Figure 7-11 Scatter chart

About step charts

A step chart shows each data point as a step in a series of steps.

Figure 7-12 shows a step chart displaying projected wireless internet usage in three global regions. The chart series are the regions. The chart categories are years.

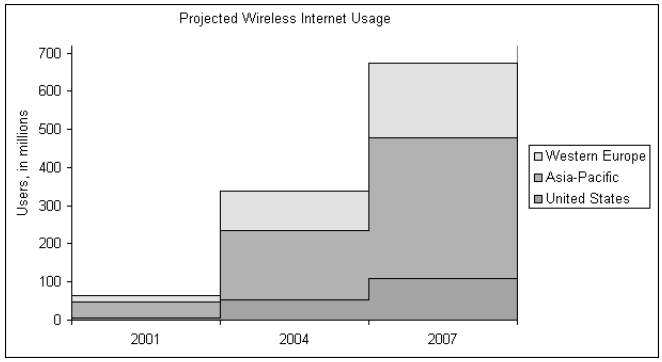


Figure 7-12 Step chart

About stock charts

A stock chart shows data as points on a time continuum, such as a business day. A stock chart displays a set of stock values in one of two ways:

- As a vertical line. Two short horizontal bars at either end of the vertical line mark the high and low values in a day. The chart can also display horizontal bars to mark the value at the beginning of the trading day, or open value, and the value at the end of the trading day, or close value. Figure 7-13 shows the chart using high, low, and close values to show each day's record and average temperatures.

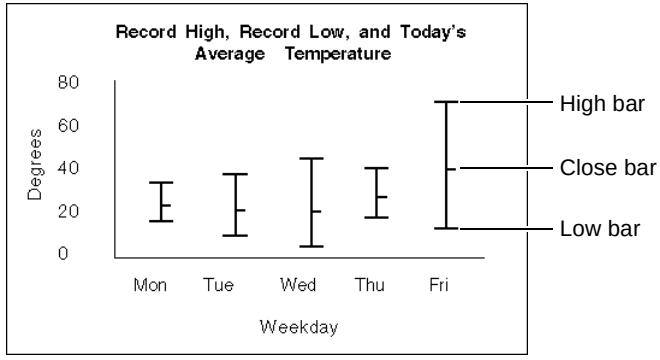


Figure 7-13 Stock charts

- As a candlestick, a box with lines extending up and down from the ends, as shown in Figure 7-14. Open and close values mark the upper and lower edges of the box. High and low values mark the upper and lower points of the line. Default bar color depends on the chart values. If the open value is higher than the close value, the box, called a down bar, is black. If the close value is higher than the open value, the box, called an up bar, is white. You can change the color of up or down bars.

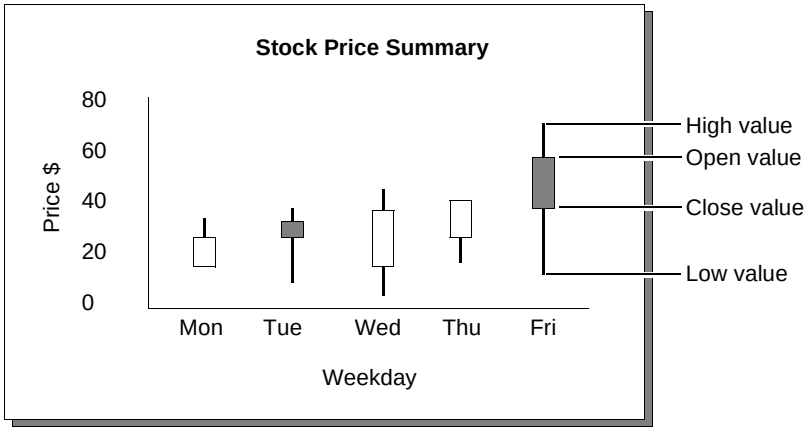


Figure 7-14 Candlestick stock chart

Understanding chart layers

A chart contains one or more chart layers. A layer is a set of points plotted against a single *y*-axis, or a set of points plotted as a pie. A chart must contain a base layer. A chart can also contain the following types of layers:

- Overlay chart layer. An overlay chart layer displays data in the same plot area as the base chart layer.
- Study chart layer. A study chart layer displays additional data below the base chart.

Using an overlay chart layer

An overlay chart layer plots points in the same area as the points in the base chart layer. An overlay chart layer shares the base chart layer's *x*-axis, but uses a second *y*-axis opposite the base chart layer's *y*-axis. For example, in Figure 7-15, the *y*-axis for the base chart layer, the bar chart, is on the left.

The *y*-axis for the overlay chart layer, the line chart, is on the right.

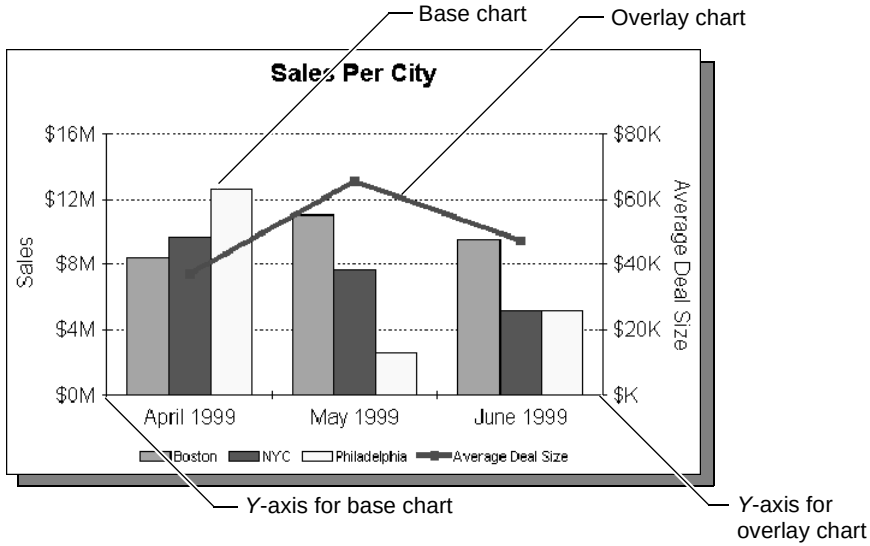


Figure 7-15 Overlay chart layer

When you use an overlay chart layer, neither the base chart layer nor the overlay chart layer can be a chart with depth. With most base chart layers, you can use the following types of charts as overlay layers:

- Area
- Bar
- Line
- Step

When you use a scatter chart base layer, you can use a scatter chart overlay layer.

Using a study chart layer

A study chart layer uses a separate set of axes shown below the base chart layer in the same plot area. A study chart layer shares *x*-axis formatting with the base chart layer. You can use different *y*-axis formatting for a study chart layer and a base chart layer.

For example, in Figure 7-16, the base chart layer is a bar chart that shows the total sales for all orders in three offices. The study chart layer is a bar chart that shows the total number of items ordered in each office.

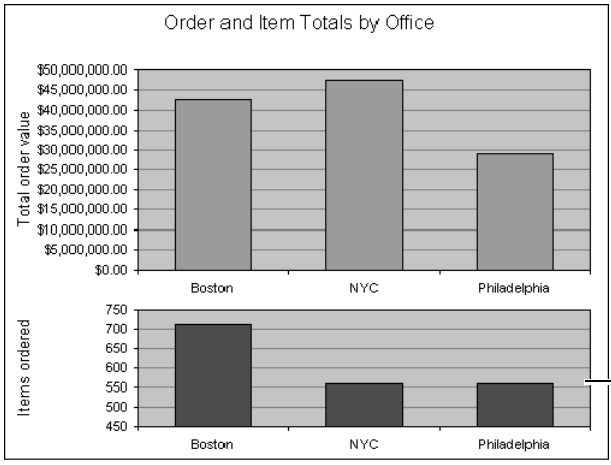


Figure 7-16 Study chart layer

When you use a study chart layer, neither the base chart layer nor the study chart layer can be a chart with depth. With most base chart layers, you can use the following types of charts as study layers:

- Area
- Bar
- Line
- Step

When you use a scatter chart base layer, you can use a scatter chart study layer.

Understanding chart data

When you build a chart, you select the data to display in the base chart layer. If necessary, you also select data for the overlay chart layer or study chart layer. To specify chart data, you can select a database column or create a data expression. An expression can use a function, such as Sum or Count, to aggregate data. You can sort or group the data to change how it appears on the report. For example, you can group data to create weekly sales totals, or to show sales by state.

Each chart type displays source data differently. The way your data source stores data and the type of chart that you use determine how you specify the data on Chart Builder. The following sections describe how you use data in a chart:

- Specifying data for an area, bar, line, or step chart
- Specifying data for a pie chart

- Specifying data for a scatter chart
- Specifying data for a stock chart
- Specifying data for an overlay or study chart layer

Specifying data for an area, bar, line, or step chart

When you build an area, bar, line, or step chart, you select data to use as chart values and series. You can also select data to use as chart categories. Every chart uses at least one value expression. A value expression determines how Chart Builder selects chart data points. How you specify series and categories depends on how your data appears in the data source, for example:

- To plot different values from one column as series or categories, specify a series or category key.
- To use different columns as series or categories, specify the columns as values, then plot them as series or categories.

For example, Figure 7-17 shows a bar chart that plots quarterly sales totals over four years.

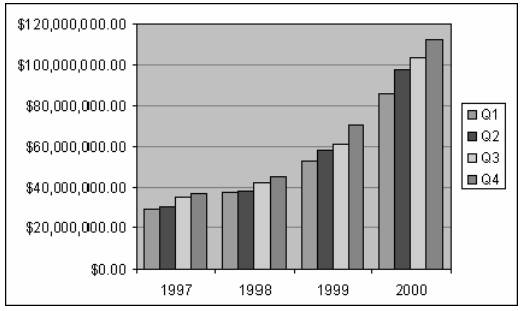


Figure 7-17 Bar chart

In one sample data source, the chart data appears in the Year, Quarter, and TotalSales columns in the QuarterlySales table. For example, Table 7-2 lists the first five data records in QuarterlySales.

Table 7-2 QuarterlySales sample data source

Year	Quarter	TotalSales
1997	Q1	30,010,000
1997	Q2	32,510,500
1997	Q3	36,903,300
1997	Q4	38,002,675
1998	Q1	38,890,900

To create the sample chart using this data, you use the following chart data expressions:

- QuarterlySales.TotalSales as values
- QuarterlySales.Year as the category key
- QuarterlySales.Quarter as the series key

In a different sample data source, the chart data appears in the TotalSalesQ1, TotalSalesQ2, TotalSalesQ3, Total SalesQ4, and Year columns in the QuarterlySales2 table. For example, Table 7-3 lists the first five data records in QuarterlySales2.

Table 7-3 TotalSales sample data source

TotalSalesQ1	TotalSalesQ2	TotalSalesQ3	TotalSalesQ4	Year
30,010,000				1997
	32,510,500			1997
		36,903,300		1997
			38,002,675	1997
38,890,900				1998

To create the chart using this data source, you use the following chart data expressions:

- QuarterlySales2.TotalSalesQ1, QuarterlySales2.TotalSalesQ2, QuarterlySales2.TotalSalesQ3, and QuarterlySales2.TotalSalesQ4 as chart value expressions plotted as series
- QuarterlySales2.Year as the chart category key

Specifying data for a pie chart

When you build a pie chart, you can use two types of data expressions, slice size and slice key. A slice size expression determines what values contribute to the size of a pie sector. A slice key expression determines how Chart Builder splits the pie into sectors.

How you specify pie sectors and sector size depends on how your data appears in the data source, for example::

- To plot different values from one column as pie sectors, specify a slice key.
- To use different columns as pie sectors, select the columns as slice size expressions, then plot them as slices.

For example, the chart in Figure 7-18 shows sales for four offices as sectors of the pie. The entire pie represents total company sales.

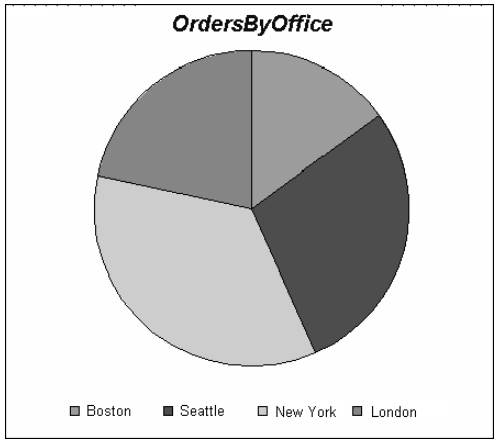


Figure 7-18 Pie chart data example

In one sample data source, the chart data appears in the Office and Orders columns in the OrdersByOffice2003 table. For example, Table 7-4 lists the first five data records in OrdersByOffice2003.

Table 7-4 Data records OrdersByOffice2003

Office	Orders
Boston	5,010
London	3,500
New York	36,903
Boston	6,750
Seattle	900

To create the sample chart using this data, you use the following chart data expressions:

- Sum([OrdersByOffice2003.Orders]) as slice size
- OrdersByOffice2003.Office as the slice key

In a different sample data source, the chart data appears in the OrdersBoston, OrdersSeattle, OrdersNewYork, and OrdersLondon columns in the Sales2003 table. For example, Table 7-5 lists the first five data records in Sales2003.

Table 7-5 Data records Sales2003

OrdersBoston	OrdersSeattle	OrdersNewYork	OrdersLondon
5,010			

(continues)

Table 7-5 Data records Sales2003 (continued)

OrdersBoston	OrdersSeattle	OrdersNewYork	OrdersLondon
			3,500
		36,903	
6,750			
	900		

To create the chart using this data source, you use OrdersBoston, OrdersSeattle, OrdersNewYork, and OrdersLondon as Sum value expressions plotted as slices.

Specifying data for a scatter chart

A scatter chart uses three types of data expressions:

- A Y value expression determines which data the chart plots on the *y*-axis.
- An X value expression determines which data the chart plots on the *x*-axis.
- A series key expression determines how the chart allocates data to series.

How you specify chart series depends on how the data appears in your data source:

- To plot different values from one column as series, specify a series key.
- To use different columns as series, select the columns as Y value expressions, then plot them as series.

A scatter chart cannot plot aggregate expressions as values.

For example, Figure 7-19 shows that the chart plots the number of employees and the yearly sales totals for several office locations.

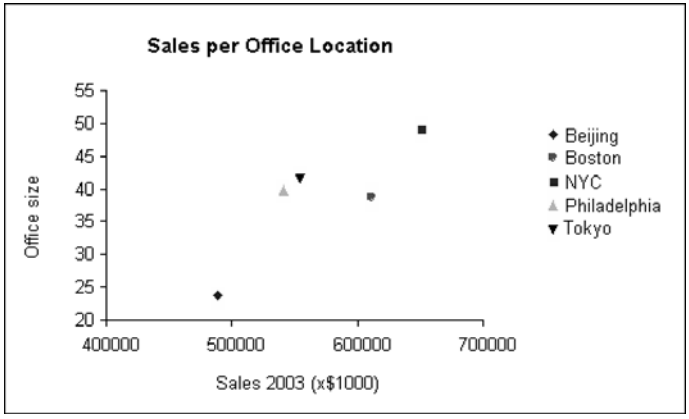


Figure 7-19 Scatter chart

In one sample data source, the chart data appears in the Office, Sales, and Employees columns in the OfficeSales table. For example, Table 7-6 lists the first five data records in OfficeSales.

Table 7-6 Data records in OfficeSales

Office	Sales	Employees
Beijing	490,100	24
Boston	615,540	39
NYC	649,905	47
Philadelphia	555,650	40
Tokyo	548,330	42

To create the sample chart using this data source, you use the following data expressions:

- OfficeSales.Sales as an X Value expression
- OfficeSales.Employees as a Y Value expression
- OfficeSales.Office as the series key expression

In a different sample data source, the chart data appears in the SalesBeijing, SalesBoston, SalesNYC, SalesPhiladelphia, and SalesTokyo columns in the 2003Sales table and the Size column in the OfficeDetails table. For example, Table 7-7 lists the first five data records in 2003Sales.

Table 7-7 Data records in 2003Sales

SalesBeijing	SalesBoston	SalesNYC	Sales Philadelphia	SalesTokyo
490,100				
			535,650	
	615,540			
				548,330
		649,905		

To create the sample chart using this data source, you use the following data expressions:

- 2003Sales.SalesBeijing, 2003Sales.SalesBoston, 2003Sales.SalesNYC, 2003Sales.SalesPhiladelphia, and 2003Sales.SalesTokyo as Y Value expressions plotted as series
- OfficeDetails.Size as an X Value expression

Specifying data for a stock chart

To create a stock chart, you use a combination of the following values:

- An open value is the price of a stock at market opening.
- A high value is the highest price a stock reaches in one trading day.
- A low value is the lowest price a stock reaches in one trading day.
- A close value is the price of a stock at market closing.
- A date value is the market date.

A stock chart must include high, low, and date values. A candlestick stock chart must also include open and close values.

For example, the candlestick chart in Figure 7-20 displays stock price data for the first two weeks in September.

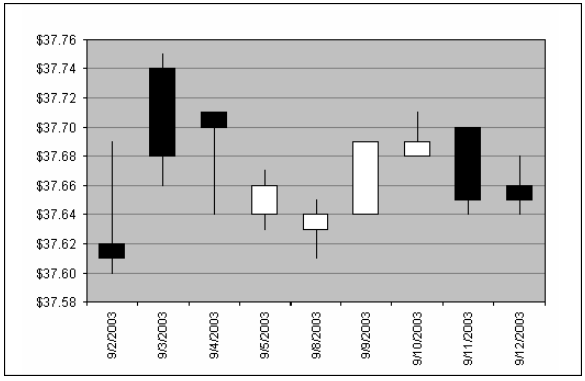


Figure 7-20 Candlestick chart

In one sample data source, the chart data appears in the Open, High, Low, Close, and Date columns in the StockData table. For example, Table 7-8 lists the first five data records in StockData.

Table 7-8 Data records in StockData

Open	High	Low	Close	Date
37.62	37.69	37.60	37.61	9/2/2003
37.74	37.75	37.66	37.68	9/3/2003
37.71	37.71	37.64	37.70	9/4/2003
37.64	37.67	37.63	37.66	9/5/2003
37.63	37.65	37.61	37.64	9/8/2003

To create the sample chart using this data source, you use the following data expressions:

- StockData.Open as an Open expression
- StockData.High as a High expression
- StockData.Low as a Low expression
- StockData.Close as a Close expression
- StockData.Date as a Date expression

In a different sample data source, the chart data appears in the Value and Timestamp columns in the Trades table. For example, Table 7-9 lists the first five data records in Trades.

Table 7-9 Data records in Trades

Value	Timestamp
37.62	9/2/2003 10:05:00 AM
37.69	9/2/2003 10:45:00 AM
37.65	9/2/2003 10:57:00 AM
37.61	9/2/2003 11:10:00 AM
37.62	9/2/2003 11:15:00 AM

To create the sample chart using this data source, you use the following data expressions:

- FIRST([Trades.Value]) as an Open expression
- MAX([Trades.Value]) as a High expression
- MIN([Trades.Value]) as a Low expression
- LAST([Trades.Value]) as a Close expression
- Trades.Timestamp as a Date expression

Specifying data for an overlay or study chart layer

Overlay chart layers and study chart layers use the same *x*-axis and categories as the main chart layer. When you specify overlay or study data, you select chart values and chart series only. How you select data depends on the type of overlay or study chart layer that you use.

Using a chart hyperlink

When you use a chart hyperlink, a user can choose a chart part to navigate to data or a report section that contains details about the chart's data. For example, in the chart that appears in Figure 7-21, Chips and Boards are the series. In the report design, you can create a link from the chart's Chips bars to the report section that contains the data in the Chips series. When a user chooses a Chips bar, the details about the data in the Chips series appears.

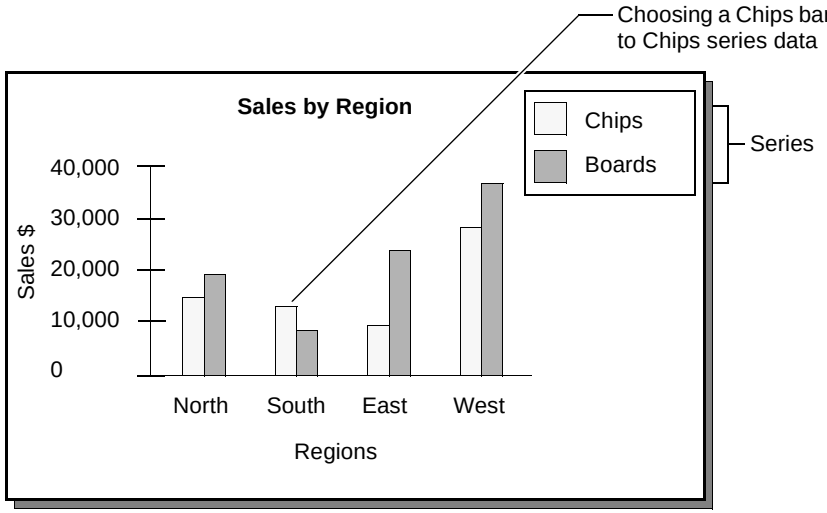


Figure 7-21 Chart hyperlink

e.Report Designer Professional supports linking to report data from the following chart parts:

- Chart control. The full chart display area, surrounding but not including specific data points, series, and categories.
- Category. Category data, such as a bar in a bar chart and a pie sector in a pie chart.
- Series. Series data, such as a line in a line chart or a bar in a bar chart.
- Data point. Specific point where a data value appears, such as a scatter chart data point marker.

Table 7-10 describes the keywords a link expression can contain.

Table 7-10 Keywords in a link expression

Keyword	Description
Category	Retrieves the Expressions➛CategoryLabelExp value for a category chart element to support linking to that data value in a report
Series	Retrieves the Expressions➛SeriesLabelExp value for a series chart element to support linking to that data value in a report
X, Y, & Z	Retrieves the Expressions➛X, Expressions➛Y, and Expressions➛Z data point values. These keywords are only valid for scatter and bubble charts. The Z value for scatter charts is always null.

For example, to link to the data for a category, you use the following link expression:

```
"detail.roi;1#SalesDetail::OfficeTitleFrame::txtOfficeName="""&
   Category & " Office"& """"
```

When you run the report, Chart Builder inserts the name of the category for each category link. For example, if the user chooses the Boston category, Chart Builder inserts Boston in txtOfficeName.

After you create chart hyperlinks, you specify whether to open a linked data section in the same window as the chart, or in a different window.

Chapter

8

Displaying data in a chart

This chapter contains the following topics:

- Creating a chart
- Modifying a chart
- Rendering a chart on Windows and UNIX

Creating a chart

To create a chart, you insert a chart control, then use Chart Builder to select a base chart type and sub-type and specify the data to use in the chart. If you use an overlay chart layer or study chart layer, you must also select a chart type and specify data for that layer. After creating a chart, you can use Chart Builder to modify chart data or attributes or to add parts to a chart.

Chart Builder uses the data in the chart to create chart text such as chart and axis titles, axis labels, and a chart legend. You can use the default settings or create custom title, label, or legend text. You can also specify formatting attributes for the chart.

Chart Builder uses a default font for chart text. You can change the default font attributes for chart titles, chart axis titles, and chart labels and legends.

You can use Chart Builder to reset all settings for a chart control, including chart data and formatting. To clear all settings for a chart, on Chart Builder, choose Reset all.

The Actuate Foundation Classes offer some chart features not described in this document. For more information about using the Actuate Foundation Classes, see *Programming with Actuate Foundation Classes*.

The following sections describe how to build a chart:

- Inserting a chart control
- Selecting a chart type
- Specifying chart data
- Sorting chart data
- Grouping chart data

Inserting a chart control

The first step in creating a chart is inserting a chart control.

How to insert a chart control

1. Drag a frame from Toolbox—Structure and drop it in the report design.
2. Choose Insert➛Chart. Position the chart control in the frame and click to drop it.
3. Use Chart Builder to create the chart.

Selecting a chart type

After inserting a chart control, you must select a chart type.

How to select a chart type

1 From the list in the upper left corner of Chart Builder—Type, select a basic chart type. Chart Builder—Type displays the chart type's associated sub-types, as shown in Figure 8-1.

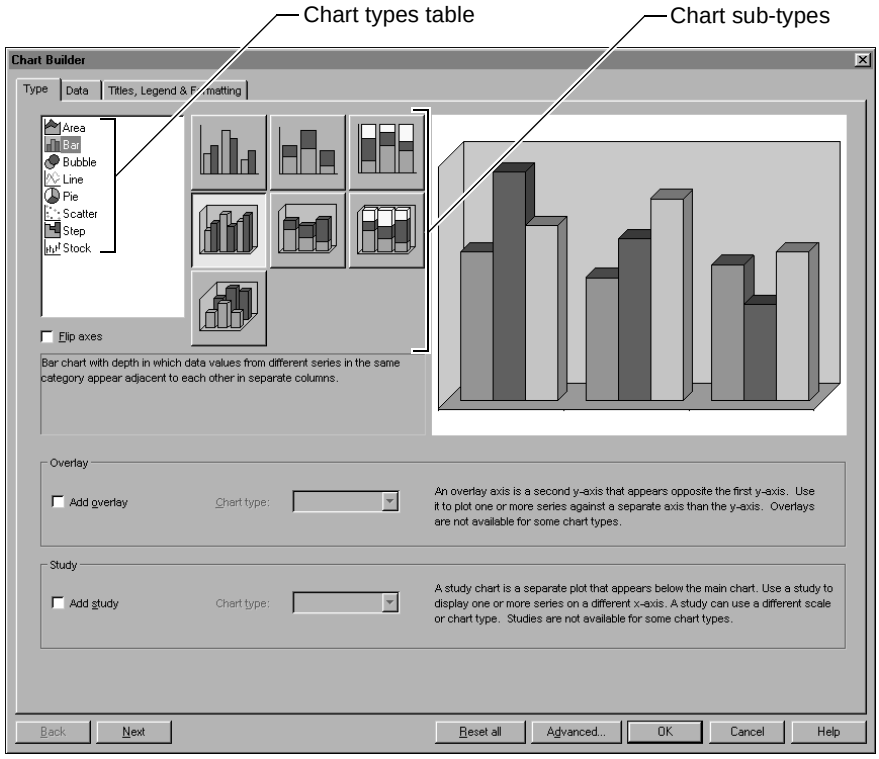


Figure 8-1 Chart Builder—Type and sub-types display

2 Select a chart sub-type. To display the bars on a bar chart horizontally rather than vertically, select Flip axes.

3 To use an overlay chart layer, in Overlay, select Add overlay. In Chart type, select an overlay layer chart type.

4 To use a study chart layer, in Study, select Add study. In Chart type, select a study layer chart type.

5 Choose Next to open Chart Builder—Data. You use Chart Builder—Data to select data for the chart.

Specifying chart data

After selecting a chart type, you must specify the data to use in the chart. If you use an overlay chart layer or a study chart layer, you must specify the data for the

base chart, then specify data for the overlay chart layer or study chart layer. How you specify chart data depends on the chart type that you use and on how your data source organizes data. For example, the steps that you use to specify stock chart data are different from the steps that you use to specify bar chart data.

When you specify chart data, you can set up data sorting and grouping to control how the data appears in the chart. If the chart displays date-and-time data, you can specify the date-and-time scale settings for the chart. You can also use Chart Builder to change the sorting, grouping, or date-and-time scale for an existing chart.

This section describes how to select chart data in the following procedures:

- How to specify data for an area, bar, line, or step chart
- How to specify data for a pie chart
- How to specify data for a scatter chart
- How to specify data for a stock chart
- How to specify data for an overlay chart layer
- How to specify data for a study chart layer

How to specify data for an area, bar, line, or step chart

1 In Chart Builder—Data, Field list presents the fields that you can use in the chart. Figure 8-2 shows Chart Builder—Data for a bar chart. Field list presents fields from the customers and items tables from the Sfdata sample database.

   To include a field in the chart, select the field in Field list, then drag it to Values. The field appears as part of a SUM expression in Values. To change the value expression, complete one of the following tasks:

   - Double-click the expression, then type the new expression text.
   - Choose Summary options, then select a function from the drop-down list.
   - Choose Expression Builder.

2 To use multiple value expressions as series or categories, complete one of the following tasks:

   - To use multiple value expressions as series, repeat step 1 until you have added all the chart series fields. In Multiple expressions, select Series. Edit series labels appears in Multiple expressions. To modify the name of a series, choose Edit series labels.
   - To use multiple value expressions as categories, repeat step 1 until you have added all the chart category fields. In Multiple expressions, select Categories. Edit categories labels appears in Multiple expressions. To modify the name of a category, choose Edit categories labels.

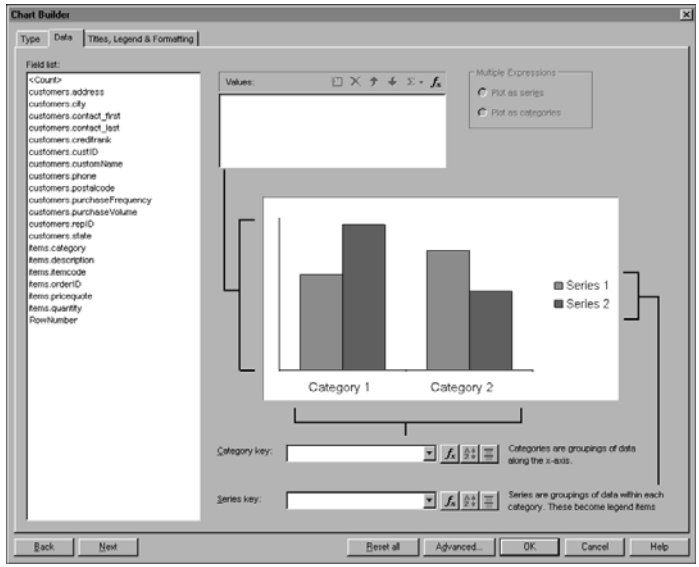


Figure 8-2 Chart Builder—Data for a bar chart

3 If you did not specify chart categories in step 2, to use a category key, complete one of the following tasks:

- Select a field to use as a category key and drag it to Category key.
- Select a field from the Category key drop-down list.
- In Category key, type a category key expression.
- Choose Expression Builder.

You cannot use an aggregate expression as a category key.

4 To sort the categories on the chart, choose Sorting.

5 To group the categories on the chart, choose Grouping.

6 If you did not specify chart series in step 2, to use a series key, complete one of the following tasks:

- Select a field to use as a series key and drag it to Series key.
- Select a field from the Series key drop-down list.
- In Series key, type a series key expression.
- Choose Expression Builder.

You cannot use an aggregate expression as a series key.

7 To sort the series, choose Sorting.

8 To group the series, choose Grouping.

9 In Chart Builder—Data, complete one of the following steps:

- To use default chart text and formatting, choose OK. The chart control appears on the report design.
- To set basic formatting attributes, choose Next. Chart Builder—Titles, Legend, and Formatting appears.
- To set advanced formatting attributes, choose Advanced. Advanced Chart Options appears.

How to specify data for a pie chart

1 In Chart Builder—Data, Field list presents the fields that you can use in the chart. In Figure 8-3, Field list presents fields from the customers and items tables from the Sfdata sample database.

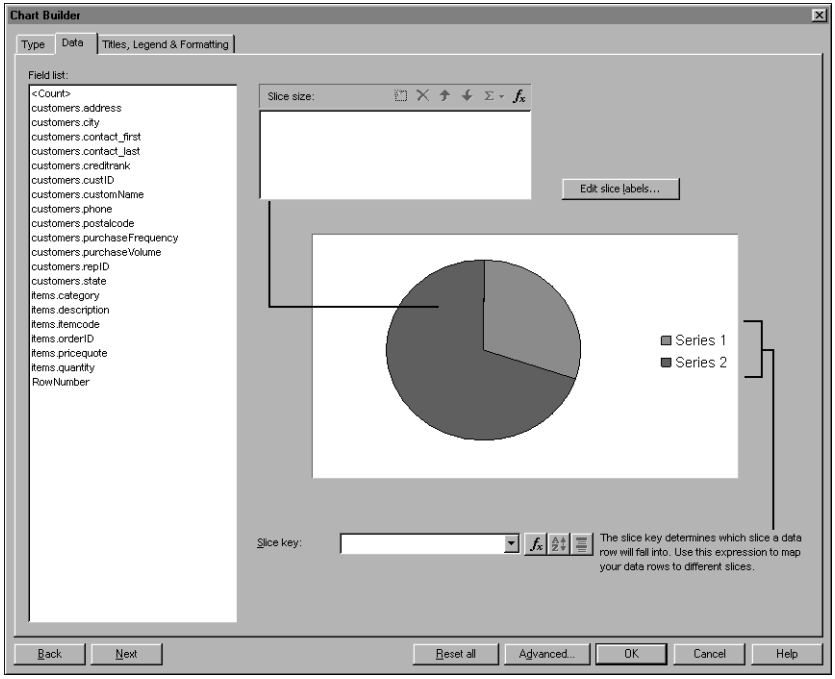


Figure 8-3 Chart Builder—Data specifying data for a pie chart

To include a field in the chart, select the field in Field list, then drag it to Slice size. The field appears as part of a SUM expression in Slice size. To change the value expression, complete one of the following tasks:

- Double-click the expression, then type the new expression text.
- Choose Summary options, then select a function from the drop-down list.

- Choose Expression Builder.

2 To use multiple value expressions as sectors, repeat step 1 until you have added all the sector fields. Edit slice labels appears. To change the name of a sector, choose Edit Slice Labels.

3 If you did not specify chart sectors in step 2, to use a sector key, complete one of the following tasks:

   - Select a field to use as a sector key and drag it to Slice key.
   - Select a field from the Slice key drop-down list.
   - In Slice key, type a sector key expression.
   - Choose Expression Builder.

   You cannot use an aggregate expression as a sector key.

4 To sort the sectors on the chart, choose Sorting.

5 To group the sectors on the chart, choose Grouping.

6 In Chart Builder—Data, complete one of the following steps:

   - To use default chart text and formatting, choose OK. The chart control appears on the report design.
   - To set basic formatting attributes, choose Next. Chart Builder—Titles, Legend, and Formatting appears.
   - To set advanced formatting attributes, choose Advanced. Advanced Chart Options appears.

How to specify data for a scatter chart

1 In Chart Builder—Data, Field list presents the fields that you can use in the chart. In Figure 8-4, Field list presents fields from the customers and items tables from the Sfdata sample database.

   To include a field in the chart, select a field and drag it to Y value. The field appears as part of a SUM expression in Y value. To change the value expression, complete one of the following tasks:

   - Double-click the expression, then type the new expression text.
   - Choose Expression Builder.

   You cannot use an aggregate expression as a Y value expression.

2 To use multiple Y value expressions as chart series, repeat step 1 until you have added all the *y*-axis value fields.

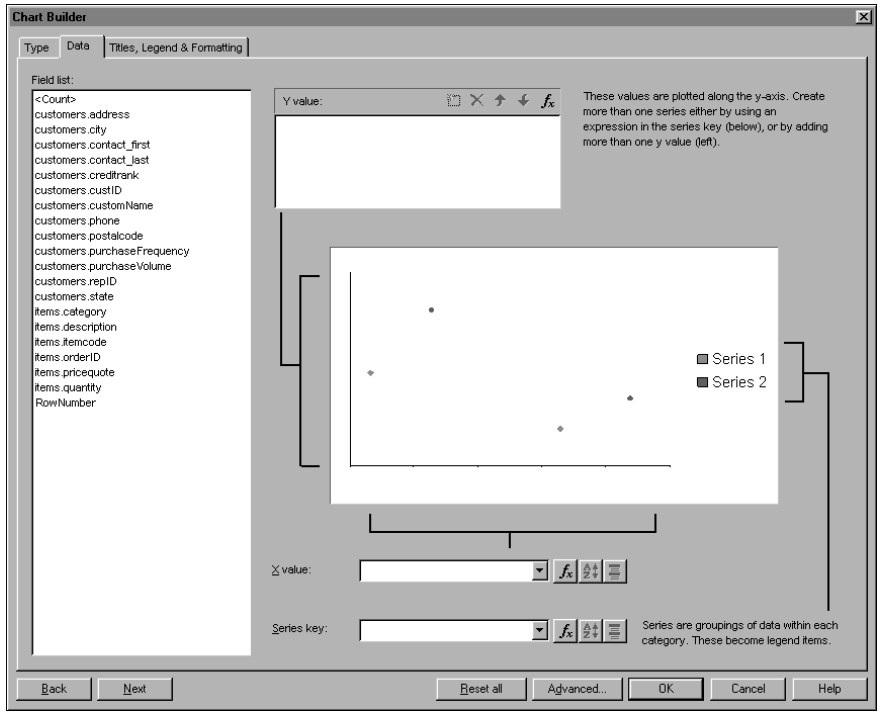


Figure 8-4 Chart Builder—Data specifying data for a scatter chart

3 To specify *x*-axis values, complete one of the following tasks:

- Select a field and drag it to X value.
- Select a field from the X value drop-down list.
- In X value, type an *x*-axis value expression.
- Choose Expression Builder.

You cannot use an aggregate expression as an X value expression.

4 To sort the *x*-axis values on the chart, choose Sorting.

5 To group the *x*-axis values on the chart, choose Grouping.

6 If you did not specify chart series in step 2, to use a series key, complete one of the following tasks:

- Select a field to use as a series key and drag it to Series key.
- Select a field from the Series key drop-down list.
- In Series key, type a series key expression.
- Choose Expression Builder.

You cannot use an aggregate expression as a series key.

7 To sort the series, choose Sorting.

8 To group the series, choose Grouping.

9 In Chart Builder—Data, complete one of the following steps:

- To use default chart text and formatting, choose OK. The chart control appears on the report design.
- To set basic formatting attributes, choose Next. Chart Builder—Titles, Legend, and Formatting appears.
- To set advanced formatting attributes, choose Advanced. Advanced Chart Options appears.

How to specify data for a stock chart

1 In Chart Builder—Data, Field list presents the fields that you can use in the chart. In Figure 8-5, Field list presents fields from the trades table from a sample database.

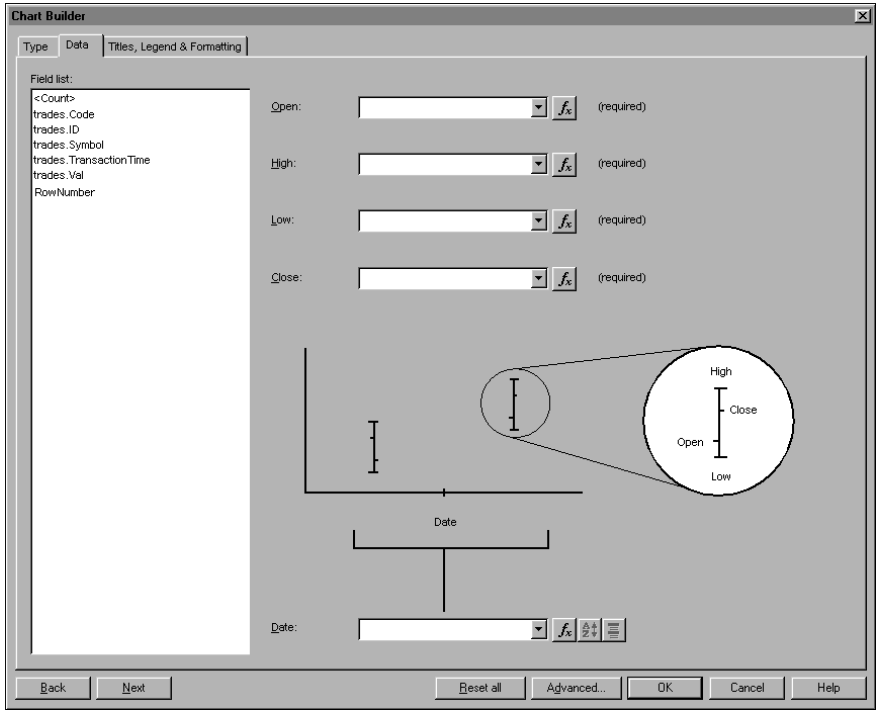


Figure 8-5 Chart Builder—Data specifying data for a stock chart

2 To specify open values, complete one of the following steps:

- Select a field and drag it to Open.
- Select an open value expression from the Open drop-down list.
- In Open, type an open value expression.
- Choose Expression Builder.

3 To specify high values, complete one of the following steps:
- Select a field and drag it to High.
- Select a high value expression from the High drop-down list.
- In High, type a high value expression.
- Choose Expression Builder.

4 To specify low values, complete one of the following steps:
- Select a field and drag it to Low.
- Select a low value expression from the Low drop-down list.
- In Low, type a low value expression.
- Choose Expression Builder.

5 To specify close values, complete one of the following steps:
- Select a field and drag it to Close.
- Select a close value expression from the Close drop-down list.
- In Close, type a close value expression.
- Choose Expression Builder.

6 Select a date field and drag it to Date.

7 In Chart Builder—Data, complete one of the following steps:
- To use default chart text and formatting, choose OK. The chart control appears on the report design.
- To set basic formatting attributes, choose Next. Chart Builder—Titles, Legend, and Formatting appears.
- To set advanced formatting attributes, choose Advanced. Advanced Chart Options appears.

How to specify data for an overlay chart layer

1 In Chart Builder—Overlay Data, Field list presents the fields that you can use in the chart. In Figure 8-6, Field list presents fields from the customers and items tables from the Sfdata sample database. Category key presents the category expression that you selected for the base chart.

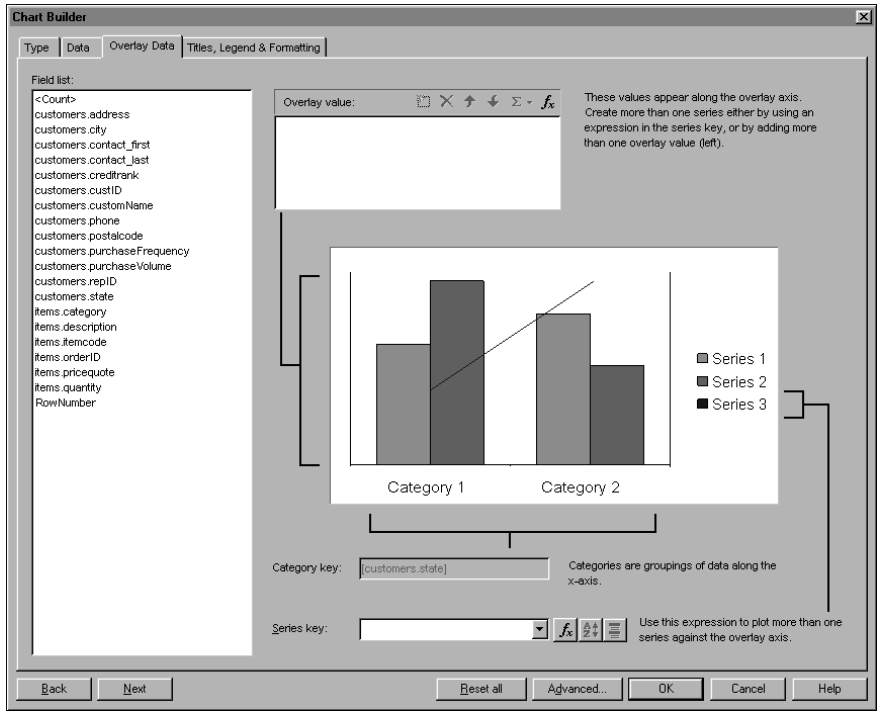


Figure 8-6 Data for a chart overlay

To include a field in the overlay chart layer, select a field and drag it to Overlay value. The field appears as part of a SUM expression in Overlay value. To change the value expression, complete one of the following tasks:

- Double-click the expression, then type the new expression text.
- Choose Summary options, then select a function from the drop-down list.

- Choose Expression Builder.

2 To use multiple overlay value expressions as series, repeat step 1 until you have added all the overlay value fields.

3 If you did not specify chart series in step 2, to use a series key, complete one of the following tasks:

- Select a field to use as a series key and drag it to Series key.
- Select a field from the Series key drop-down list.
- In Series key, type a series key expression.
- Choose Expression Builder.

You cannot use an aggregate expression as a series key.

4 To sort the series on the chart, choose Sorting.

5 To group the series on the chart, choose Grouping.

6 In Chart Builder—Data, complete one of the following steps:

- To use default chart text and formatting, choose OK. The chart control appears on the report design.
- To set basic formatting attributes, choose Next. Chart Builder—Titles, Legend, and Formatting appears.
- To set advanced formatting attributes, choose Advanced. Advanced Chart Options appears.

How to specify data for a study chart layer

1 In Chart Builder—Study Data, Field list presents the fields that you can use in the chart. In Figure 8-7, Field list presents fields from the customers and items tables from the Sfdata sample database. Category key presents the category expression that you selected for the base chart.

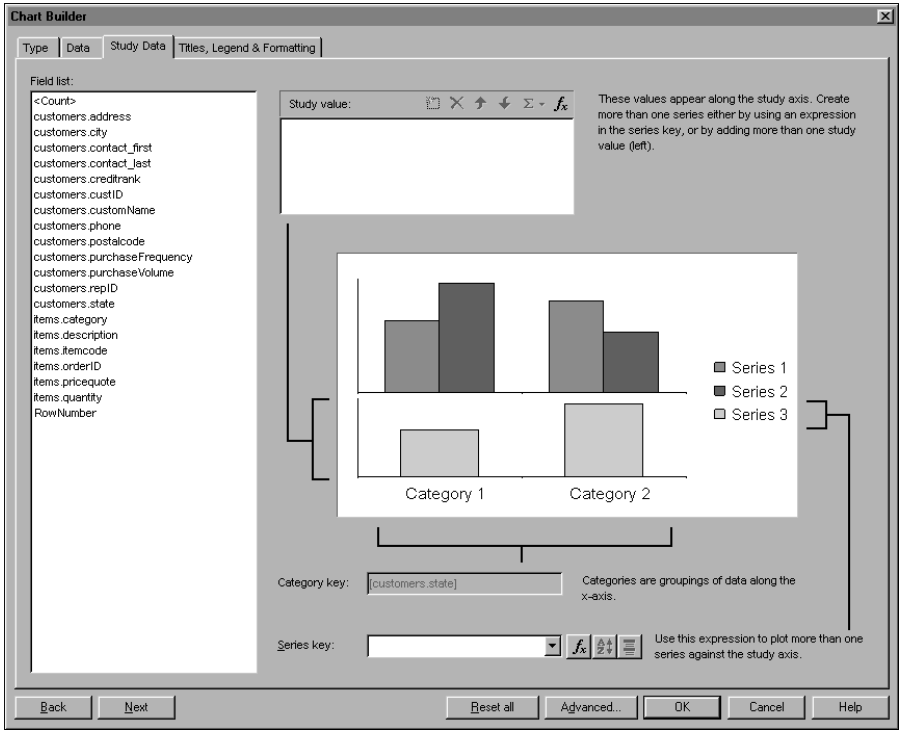


Figure 8-7 Chart Builder—Study Data

To include a field in the study chart layer, select a field and drag it to Study value. The field appears as part of a SUM expression in Study value. To change the value expression, complete one of the following tasks:

- Double-click the expression, then type the new expression text.
- Choose Summary options, then select a function from the drop-down list.
- Choose Expression Builder.

2 To use multiple study value expressions as series, repeat step 1 until you have added all the study value fields.

3 If you did not specify chart series in step 2, to use a series key, complete one of the following tasks:

- Select a field to use as a series key and drag it to Series key.
- Select a field from the Series key drop-down list.
- In Series key, type a series key expression.
- Choose Expression Builder.

You cannot use an aggregate expression as a series key.

4 To sort the series on the chart, choose Sorting.

5 To group the series on the chart, choose Grouping.

6 In Chart Builder—Data, complete one of the following steps:

- To use default chart text and formatting, choose OK. The chart control appears on the report design.
- To set basic formatting attributes, choose Next. Chart Builder—Titles, Legend, and Formatting appears.
- To set advanced formatting attributes, choose Advanced. Advanced Chart Options appears.

Sorting chart data

When you select data for a chart, you can use sorting to display chart data in a different order from the database order. You can also use Chart Builder to set up sorting for an existing chart. To open Chart Builder for a chart, double-click the chart control.

How to sort chart data

1 In Chart Builder—Data, to the right of the field to sort, choose Sorting.

2 In Sorting and Grouping—Sorting, select the sorting options:

- To sort the data, in Sorting, select Sort by <field name>, then select Ascending or Descending.

- To display data in the order in which it appears in the data source, in Sorting, select Keep original order. In Direction, select Forward.
- To display data in the reverse of the order in which it appears in the data source, in Sorting, select Keep original order. In Direction, select Reverse.

Choose OK.

Grouping chart data

You can group the data a chart displays. For example, you can group sales records to show sales totals by month. When you set up chart groups, you determine how to allocate data to a group. You can also set the range of data to use. For example, you can group order numbers between 200 and 800 in groups of 50. To include a group without data, you can add a missing group or create a static group.

You can set up grouping when you first select data for a chart. You can also use Chart Builder to set up grouping for an existing chart. To open Chart Builder for a chart, double-click the chart control.

Setting up a group

To allocate data to a group, you use the following settings:

- The data type of the expression to group determines the available group units and intervals.
- The group unit determines the size of the groups. For example, you can use a month unit for date-and-time data.
- The group interval determines when the range for each group begins and ends. For example, using a group unit of month, you can use a group interval of 2 to show data in two-month groups.

For example, the chart in Figure 8-8 shows the item counts of the orders placed by a group of sales representatives. The chart uses the number of orders as values, sales representatives as series, and the number of items per order as categories. Categories appear in groups of ten, using a group unit of integer and a group interval of ten. For example, the chart shows that Barajas had seven hundred orders that included between zero and nine items.

The chart in Figure 8-9 shows the number of orders of each status in two-month periods. This chart uses the number of orders as values, order status as series, and the order dates as categories. The categories appear in groups of two months, using a group unit of month and a group interval of two.

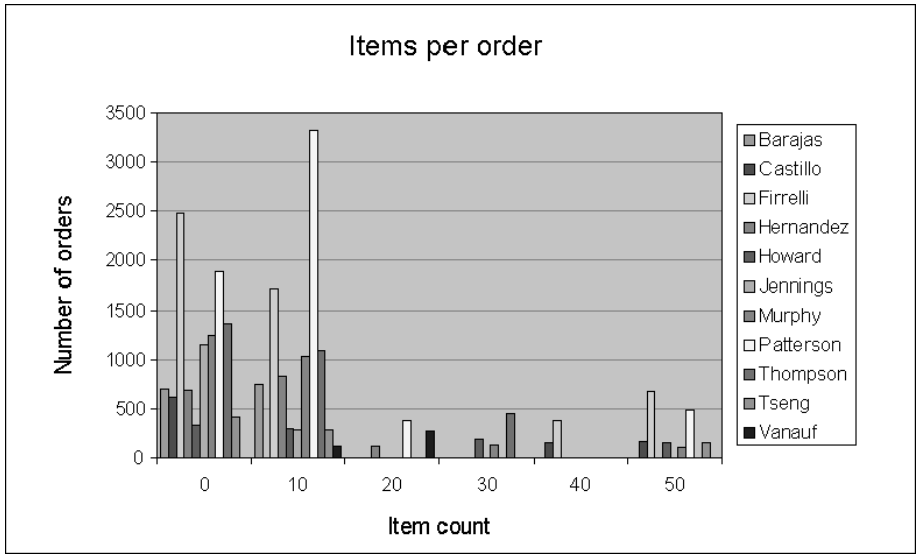


Figure 8-8 Allocating data to a group by item count

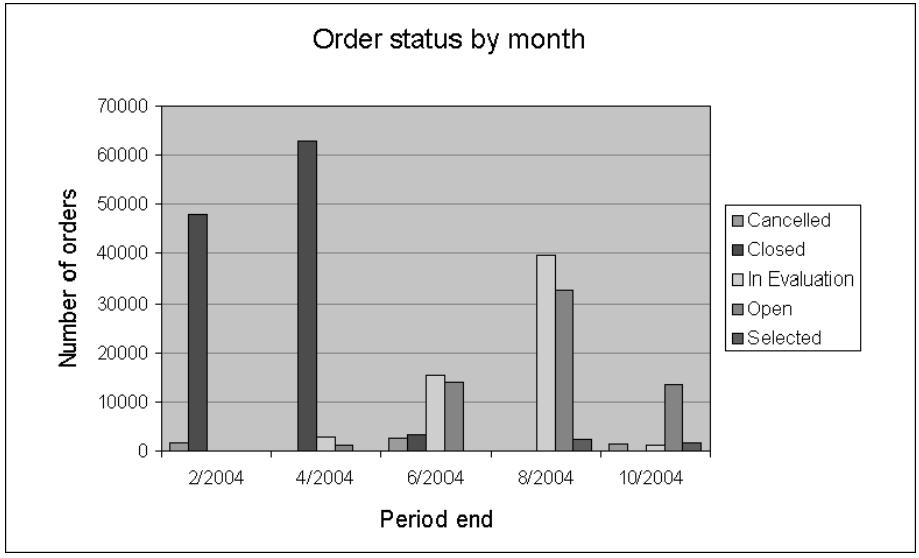


Figure 8-9 Allocating data to a group in a bi-monthly timeframe

How to group chart data

1 In Chart Builder—Data, to the right of the field to group, choose Grouping.

2 In Sorting and Grouping—Grouping, specify grouping settings:

- In Data type, select the data type of the expression to group.

- Select Group.
- In Unit, select a group unit. If you select Week, you can specify the day to use as the first day of the week.
- In Interval, select the group interval.

For example, to display data in five year intervals, use the Date and Time data type, the Year unit, and an interval of 5. Choose OK.

Adding missing groups

When you group chart data, you can have Chart Builder insert missing groups. Chart Builder adds missing groups based on date-and-time or counting logic. For example, if you group data by week of the year, Chart Builder can add a week for which there is no data, as shown in Figure 8-10.

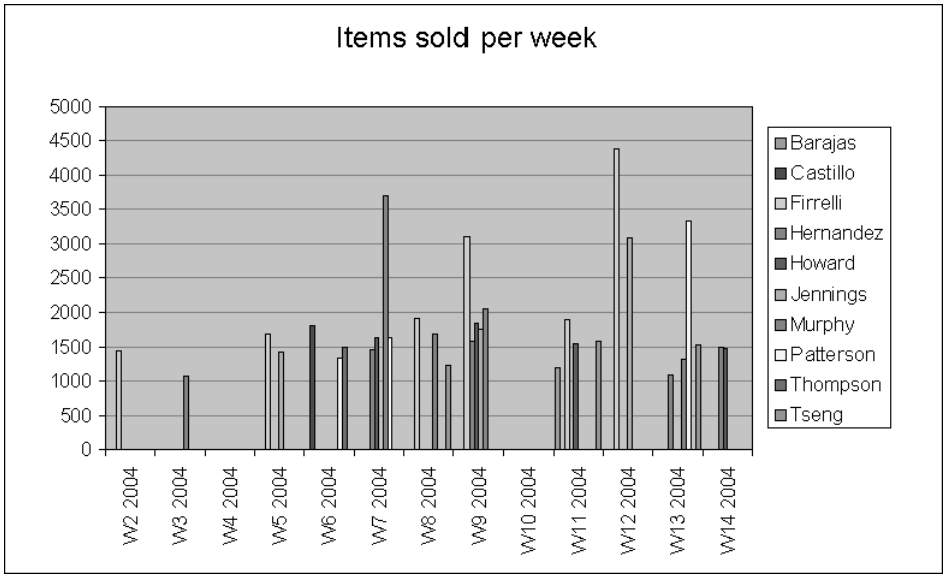


Figure 8-10 Adding missing groups to a chart

When you add missing groups, you can also set the range of data to group. To set the group range, you use round and span settings. If you do not specify round or span settings, the chart uses the data range from the report.

Round settings determine where to begin and end the range using standard start and end points. For example, the standard start and end points for a year are January and December. The standard start and end points for integer units of ten are zero and nine, ten and nineteen, and so on.

Span settings determine a minimum length for the range without using standard start and end points. When you set a group span, the range begins with the first

group for which there is data. The end point is either with the last group for which there is data or the last group the span determines. For example, if you set a span of 1000 for a range of customer IDs that begin with 450 and end with 1600, the range ends at 1600. If the customer IDs begin with 450 and end with 1300, the range ends at 1450.

To specify rounding or span, you use the following settings:

- The round or span unit determines the length of the range.
- The round or span interval determines the number of units to use to round or extend the range.

For example, the chart in Figure 8-11 shows the number of orders of each status by month. The chart uses the number of orders as values, order status as series, and the order dates as categories. The category key uses a round unit of quarter and an interval of one. The group range begins with the first month of the first quarter for which there is data, Q4 2003, and ends in the last month of the last quarter for which there is data, Q4 2004.

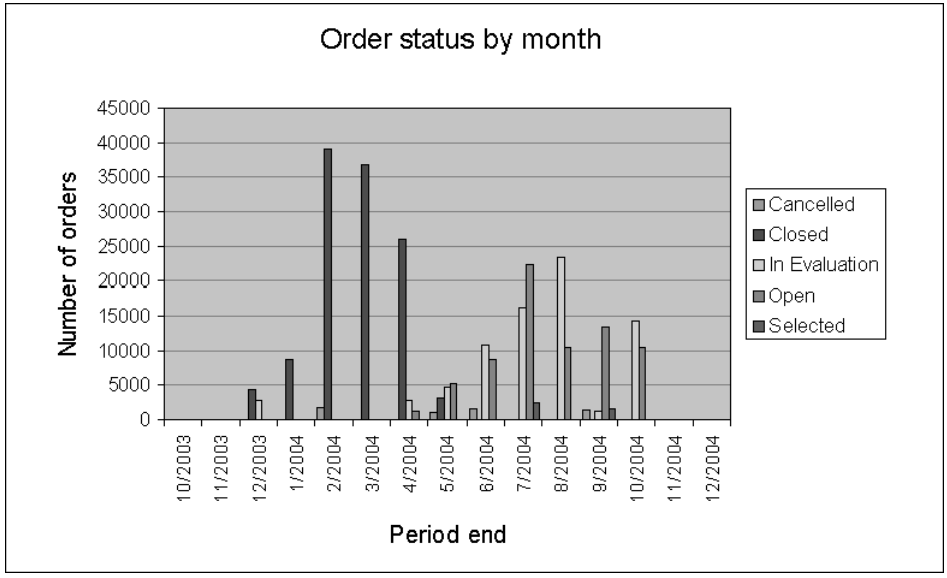


Figure 8-11 Bar chart showing order status by month

Figure 8-12 shows a similar chart using span settings. The category key uses a span unit of year and an interval of one. The group range begins with the first group for which there is data, 11/2003. Because the last group for which the chart displays data, 9/2004, is not at least a year from the beginning of the range, the range ends with 10/2004.

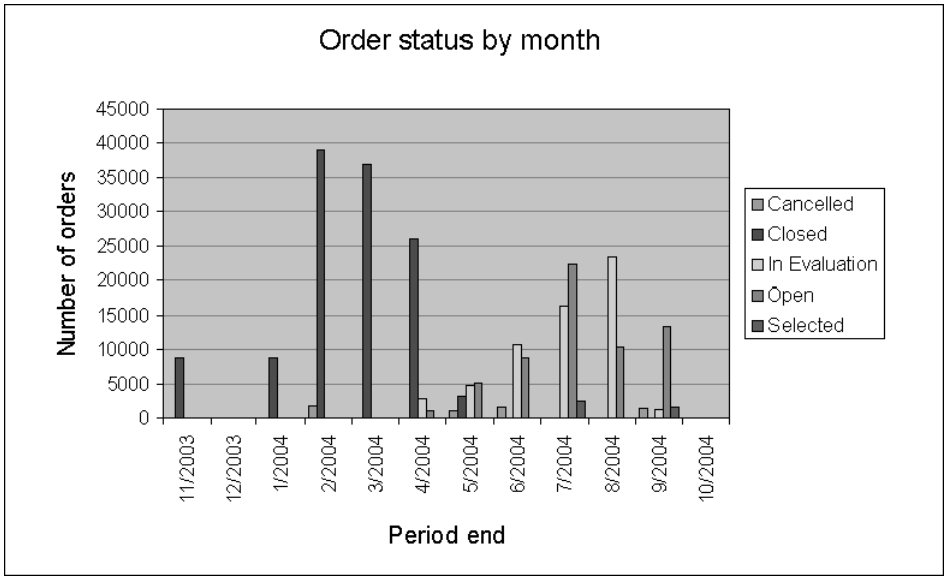


Figure 8-12 Bar chart using span settings

How to add missing groups

1 In Chart Builder—Data, to the right of the field to group, choose Grouping.

2 In Sorting and Grouping—Grouping, indicate how to treat missing values:

- To display a group even if no data for that group appears in the data source, select Add missing groups. For example, if your data source includes data for 1, 2, 3, and 5, you can use Add missing groups to display 4 as a group in the chart.
- To keep Chart Builder from adding a missing group, in Do not add these values, type the group to exclude. The chart displays the group only if data for that group appears in the data source.
- To exclude a day when you group data by day, in Do not add these days, select the day to exclude. For example, using the settings in Figure 8-13, the chart does not include data for Saturday or Sunday.

Choose OK.

3 To set the group range, choose Advanced Grouping.

4 In Sorting and Grouping—Advanced Grouping, to set the start and ending values for the group range:

- Select Round.
- To determine where to begin and end adding missing groups, select a rounding unit and a rounding interval. The unit and the interval must be

the same or larger than the group unit that you selected on Sorting and Grouping—Grouping.

For example, if you group data by month and use a rounding unit of Quarter and a rounding interval of 1, the first group is the first month of the first quarter for which data appears. The last group is the last month of the last quarter for which data appears. If you group data by month and use a rounding unit of Month and a rounding interval of 1, the first group is the first month for which data appears. The last group is the last month for which data appears.

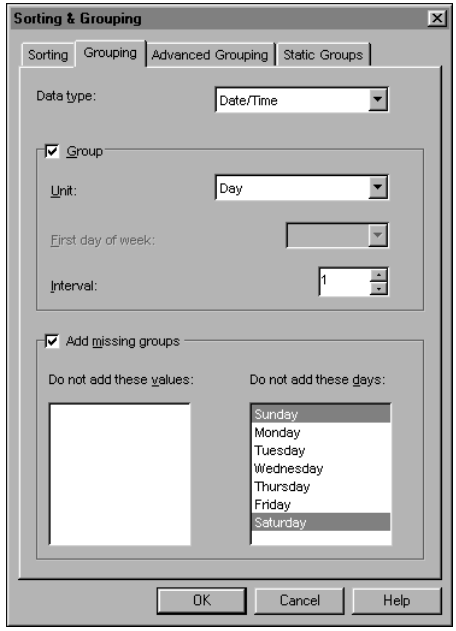


Figure 8-13 Adding missing groups to a chart

5 To determine the minimum span of the group range:

- Select Span.
- Select a span unit and a span interval. The span unit must be the same as or larger than the rounding unit that you selected in step 4. For example, if you group data by month and use a span unit of Year and a span interval of 1, the first group is the first month for which data appears. The last group is at least one year after the first group. Choose OK.

Adding a static group

You can add a static group to a chart. A static group always appears in the chart, even if the data source does not include data for that group. A static group is user-defined and is not based on date-and-time or counting logic. For example, in

a chart that shows order totals by office, as shown in Figure 8-14 you can include an office that does not show any orders.

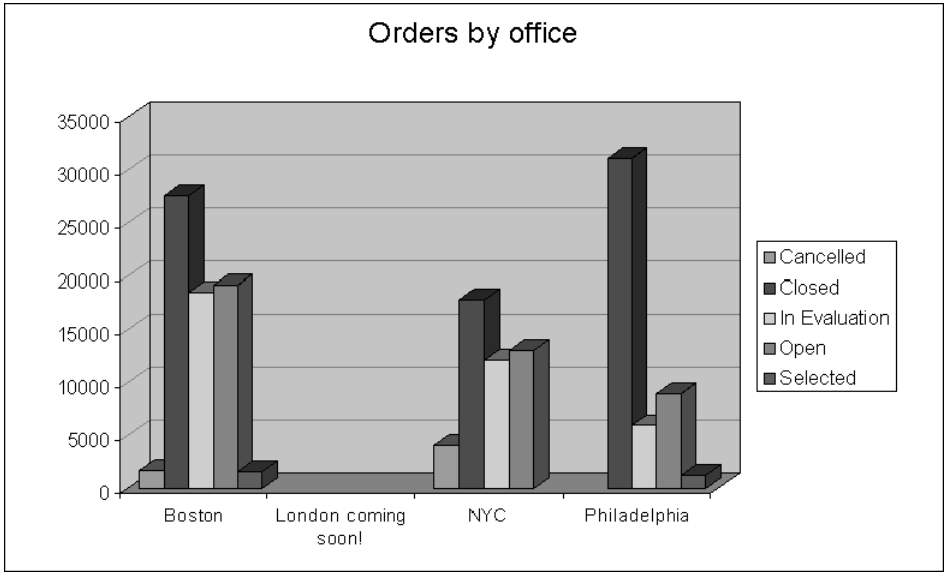


Figure 8-14 Bar chart showing order totals by office

Chart Wizard saves a static group name as a string value. You cannot apply a value format to a string label value. An axis label for a static group displays the group name that you define when you create the group.

How to create a static group

1. In Chart Builder—Data, to the right of the field to group, choose Grouping.
2. In Sorting and Grouping—Grouping, to specify a group that should always appear in the chart, even if it does not appear in the data source, choose Static Groups.
3. In Sorting and Grouping—Static Groups, specify a new group:
   - In Data type, select the data type of the expression to group.
   - In Value, type a value that uses the data type that you selected in Data type.

     To create a date-and-time group, you must type a value from the group in the relevant data format. For example, to create a 2002 group, you type a 2002 date in date-and-time format, such as 1/1/2002.
   - In Label, type the display name for the group. If you do not type label text, the displayed group name is the same as the text in Value.

   Choose OK.

Modifying a chart

After creating a chart, you can resize the chart control or modify chart formatting. Chart Builder adds some parts of a chart automatically, such as a legend. You can remove or modify these parts. You can also add other items to a chart, such as data labels or drop lines. You use Chart Builder to modify the following basic chart attributes:

- Titles. You can use default title text, provide new title text, or hide titles. You can format title text and background and set title positioning.
- Chart area. You can modify the appearance of the chart background or plot. In a chart with depth, you can change the appearance of the chart floor or walls.
- Legend. You can hide a legend or set where a legend appears on a chart. You can also change legend formatting. For example, you can modify legend outline style or change the text or the text style of legend items.

You use Advanced Chart Options to modify the following advanced chart attributes:

- Axes. You can modify axis line style or change the scale of an axis. You can add grid lines that extend from axis ticks and modify axis label text, formatting, and position. You can also specify where one axis intersects another.
- Series or category attributes. You can change how Chart Builder plots empty values in a series. You can also change how series data appears on the chart. For example, you can use point labels to identify data points or set the color or style of each series on the chart.
- Chart hyperlinks. You can create a link from a part of the chart to the associated chart data.
- Sample chart data. You can modify the data a chart displays in the layout window.

You can also use Chart—Properties to change chart component properties. Chart component properties are similar to the component properties of other e.Report Designer Professional controls.

How to resize a chart control

To resize a chart control, drag the handles on the chart control until it is the desired size.

How to modify an existing chart

1. To open Chart Builder for an existing chart, double-click the chart control.
2. In Chart Builder—Type, complete one of the following tasks:

- To modify the chart type, on Chart Builder—Type, make the necessary changes.
- To modify the chart data, choose Data. Chart Builder—Data appears.
- To modify the chart titles, legend, or chart area, choose Titles, Legend, and Formatting. Chart Builder—Titles, Legend, and Formatting appears. For information about modifying a title, a legend, or chart area formatting, see the relevant section, later in this chapter:
  - Working with chart titles
  - Formatting the chart area
  - Working with a chart legend
- To modify advanced chart attributes, choose Advanced. Advanced Chart Options appears.

3 In Chart Builder, choose OK.

How to modify chart component properties

1 In the layout window, right-click the chart control and choose Properties.

2 In the Properties page for the chart control, make the necessary changes. Then choose OK.

Working with chart titles

Chart Builder creates chart and axis titles based on the chart data. You can use the default title text or create new titles. You can also change title formatting and position.

When you create a chart, Chart Builder uses the chart data expressions to create chart titles. Chart Builder can create the following types of titles:

- *Y*-axis, overlay axis, and study axis titles. Chart Builder uses the value expression to create a default *y*-axis, study axis, or overlay axis title.
- *X*-axis titles. Chart Builder uses the category and series expressions to build a default *x*-axis title.
- Chart titles. Chart Builder uses the *x*-axis and *y*-axis titles to create a default chart title.

For example, if a chart uses orders.state as the category expression, orders.type as the series expression, and Sum([orders.pricequote]) as the value expression, Chart Builder creates the following default titles:

- An *x*-axis title, state and type
- A *y*-axis title, Sum of pricequote
- A chart title, Sum of pricequote / state and type

By default, Chart Builder displays a title for the chart and for each axis on the chart.

How to set up chart titles

1. In Chart Builder, choose Titles, Legend, and Formatting.
2. In Chart Builder—Titles, Legend, and Formatting, to hide a title, deselect the title checkbox. For example, to hide the chart title, deselect the Chart checkbox.
3. Specify chart text:
   - To use default title text, select Auto-text.
   - To type title text, deselect Auto-text and type the new title.
4. To format chart title text:
   1. Choose Style.
   2. In Chart Title, in Orientation, indicate how the title text appears:
      - To use default orientation, select Automatic.
      - To display title text vertically from top to bottom, select Vertical.
      - To specify another orientation angle, select Custom, then select an angle value.
   3. To change the title font, choose Font.
   4. In Font, specify title font attributes. Choose OK.
   5. To specify title background color, choose Background.
   6. In the color selector, select a background color or use Custom Color to create a background color.
   7. To specify title border attributes, choose Border.
   8. In Border, specify border attributes. Choose OK.
5. To modify another title, repeat steps 3 and 4 for the *x*-axis, *y*-axis, overlay *y*-axis, or study *y*-axis titles.
6. In Chart Builder—Titles, Legend, and Formatting, choose OK.

Formatting the chart area

In a chart without depth, you can modify the appearance of the chart background, chart border, plot background, or plot border. In a chart with depth, you can modify the appearance of the chart background, chart border, walls, or floor. When you use a pie chart, you can modify the chart background only.

How to format the chart area in a chart without depth

1. In Chart Builder, choose Titles, Legend, and Formatting.
2. In Chart Builder—Titles, Legend, and Formatting, modify the chart area background color:
   1. Choose Chart background.
   2. In the color selector, as shown in Figure 8-15, select a background color or use Custom Color to create a background color.

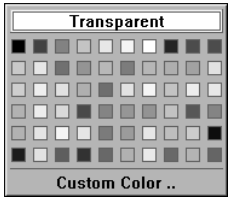


Figure 8-15 Color selector

3. To modify the chart border:
   1. Choose Chart border.
   2. In Border, as shown in Figure 8-16, specify chart border attributes.

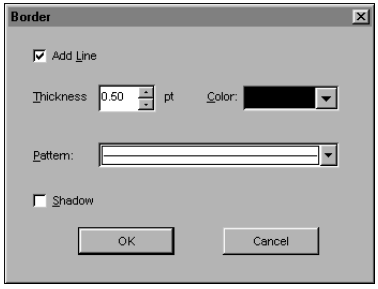


Figure 8-16 Border setup

   Choose OK.
4. To modify the plot background color:
   1. Choose Plot background.
   2. In the color selector, select a background color or use Custom Color to create a background color.
5. To modify the plot border:
   1. Choose Plot border.
   2. In Border, specify plot border attributes. Choose OK.
6. In Chart Builder—Titles, Legend, and Formatting, choose OK.

How to format the chart area of a chart with depth

1. In Chart Builder, choose Titles, Legend, and Formatting.
2. In Chart Builder—Titles, Legend, and Formatting, to modify the chart area background color:
    1. Choose Chart background.
    2. In the color selector, select a background color or use Custom Color to create a background color.
3. To modify the chart border:
    1. Choose Border.
    2. In Border, specify border attributes. Choose OK.
4. To modify the wall color of a 3-D chart or a chart with depth:
    1. Choose Walls color.
    2. In the color selector, elect a background color or use Custom Color to create a background color.
5. To modify the floor color of a 3-D chart or a chart with depth:
    1. Choose Floor color.
    2. In the color selector, select a background color or use Custom Color to create a background color.
6. In Chart Builder—Titles, Legend, and Formatting, choose OK.

Working with a chart legend

You can modify where a legend appears on a chart. You can also change legend font, background, and border attributes or select a number format for legend items.

You can modify the text that appears in a legend. If you use a series or a category key, you can build an expression from which to generate legend text. If you use multiple values as series or categories, you can create custom series or category names that appear in the legend.

Stock chart series names do not appear in a legend. If a chart that uses a stock chart as a base chart includes overlay and study data, the overlay and study series can appear in a legend.

How to place or format a chart legend

1. In Chart Builder, choose Titles, Legend, and Formatting.
2. In Chart Builder—Titles, Legend, and Formatting, deselect Legend to hide the legend.

3 To change where the legend appears on the chart, in Legend, select a legend position.

4 To change the legend font:

   1 In Formatting, choose Font.

   2 In Font, specify legend font attributes. Choose OK.

5 To modify the legend background color:

   1 In Formatting, choose Background.

   2 In the color selector, select a background color or use Custom Color to create a background color.

6 To modify the legend border:

   1 Choose Border.

   2 In Border, specify border attributes. Choose OK.

7 In Chart Builder—Titles, Legend, and Formatting, choose OK.

How to modify the number format of legend items

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Legend.

3 In Advanced Chart Options—Legend, to change the format of the base chart legend text, in Base layer legend text, select or type the new format. Figure 8-17 shows some of the format options.

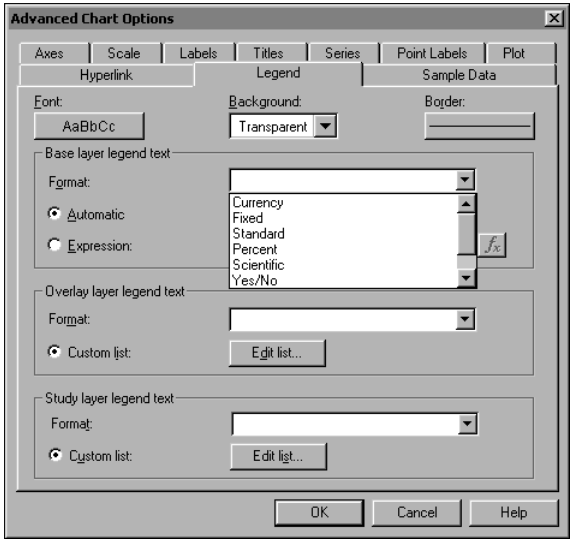


Figure 8-17 Advanced Chart Options—Legend

4 To change the format of the overlay or study chart layer, select or type the new format in Overlay layer legend text or Study layer legend text.

5 Choose OK.

6 In Chart Builder, choose OK.

How to build a label text expression

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Legend.

3 In Advanced Chart Options—Legend, to create an expression for the legend items for the overlay or study chart layer, select Expression. Then choose either Expression Builder in Overlay layer legend text or Study layer legend text.

4 Choose OK.

5 In Chart Builder, choose OK.

Working with axes

You can modify axis attributes, such as line style and tick mark frequency. You use axis scale to modify how data is placed along an axis. You can also add and modify axis labels. You use Advanced Chart Options to work with a chart axis.

About the *x*-axis

All types of charts except pie charts have an *x*-axis, which is the line along which the major divisions of data are represented. An overlay chart layer or study chart layer uses the base chart's *x*-axis formatting. Figure 8-18 shows the parts of the *x*-axis you can format.

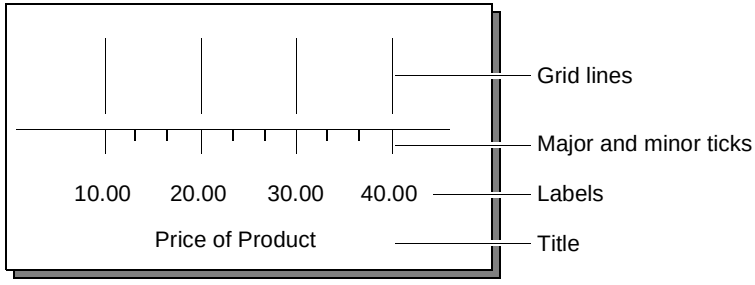


Figure 8-18 Formatting the *x*-axis

About the *y*-axis

All types of charts except pie charts have a *y*-axis, which is the line along which the data values are represented. Figure 8-19 shows the parts of the *y*-axis you can format.

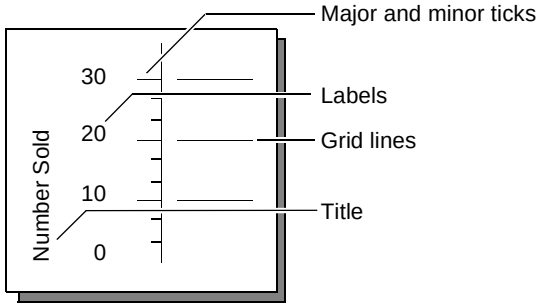


Figure 8-19 Formatting the *y*-axis

Formatting an axis

You can change the appearance of an axis line or the tick marks that appear on an axis. On charts without depth, you can set where an axis intersects another axis. All axes use major divisions. Value axes also use minor divisions.

You can add grid lines that extend from an axis at the same positions as the tick marks. For 2D charts, you can also add grid lines at arbitrary positions, for example to show where one fiscal year ends and another begins, or to show a threshold for acceptable sales. Adding grid lines at arbitrary positions requires coding. For an example of a chart that uses custom grid lines, see <eRDPro_HOME>\Examples\DesignAndLayout\Charts\CustomGridLines\CustomGridLines.rod.

How to modify major or minor tick marks

1. In Chart Builder, choose Advanced.
2. In Advanced chart Options, choose Axes.
3. In Advanced Chart Options—Axes, to set *y*-axis options, choose Y Axis.
4. In Advanced Chart Options—Axes—Y Axis, in Major, set major tick mark options:
    1. To hide major tick marks, deselect Ticks.
    2. To change major tick mark position, in Ticks, select a position option.
    3. To modify major axis grid lines, choose the line button below Grid.
    4. In Line Style, set grid line options, as shown in Figure 8-20.

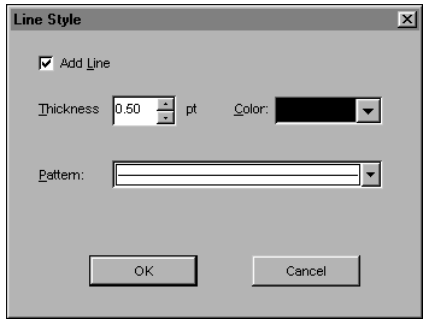


Figure 8-20 Formatting line style

Choose OK.

5 In Minor, set minor tick mark options:

   1 To hide minor tick marks, deselect Ticks.

   2 To change minor tick mark position, select an option in Ticks.

   3 To modify minor axis grid lines, choose the button below Grid.

   4 In Line Style, set grid line options, then choose OK.

6 To set the tick mark options on another axis, choose X Axis, Overlay, or Study, then repeat steps 4 and 5 for that axis.

7 Choose OK.

8 In Chart Builder, choose OK.

How to position an intersecting axis

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Axes.

3 In Advanced Chart Options—Axes, choose X Axis.

4 In Advanced Chart Options—Axes—X Axis, in Y Axis position, select a position option:

- To use the default *y*-axis position for the chart, select Auto.
- To display the *y*-axis to the left of the *x*-axis, select Left.
- To display the *y*-axis to the right of the *x*-axis, select Right.
- To specify an *x*-axis value at which the *y*-axis intersects, select Cross at X = and type an *x*-axis value.

5 To set the position of another axis, choose Y Axis, Overlay, or Study, then select a position option for that axis.

6 Choose OK.

7 In Chart Builder, choose OK.

How to change axis line style

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Axes.

3 In Advanced Chart Options—Axes, choose X Axis.

4 In Advanced Chart Options—Axes—X Axis, to change *x*-axis line style, choose the button to the right of Line style.

5 In Line Style, set line style options. Choose OK.

6 To change the line style of another axis, choose Y Axis, Overlay, or Study, then repeat steps 4 and 5 for that axis.

7 Choose OK.

8 In Chart Builder, choose OK.

Working with axis scale

To change the scale and placement of data relative to a value axis, modify the axis scale. You can set axis divisions and set the minimum or maximum value on an axis. You can also set the number of major divisions and the number of minor ticks that appear in each major division. You can set scale for each value axis on a chart. For example, if you use a scatter base chart layer with a scatter overlay chart layer, you can set scale options for the *x*-axis, the *y*-axis, and the overlay *y*-axis.

How to set major axis divisions

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Scale.

3 In Advanced Chart Options—Scale, choose Y Axis.

4 In Advanced Chart Options—Scale—Y Axis, in Interval, set the major axis interval options:

- To have Chart Builder set major axis divisions, select Auto.
- To specify a minimum interval to use, select At least, then type the minimum interval value.
- To specify an interval, select Exactly, then type the interval value.

5 In Count, specify how many interval marks to display on the axis:

- To specify a maximum number of interval marks, select No more than, then select the maximum number of interval marks.

- To specify the number of interval marks to use, select Exactly, then select the number of interval marks to use.

If Interval and Count settings conflict, a value that you select in Interval takes precedence over a value that you select in Count.

6 To set major axis divisions for another chart axis, choose X Axis, Overlay or Study, then repeat steps 4 and 5 for that axis.

7 Choose OK.

8 In Chart Builder, choose OK.

How to set a minimum or maximum value for an axis

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Scale.

3 In Advanced Chart Options—Scale, choose Y Axis.

4 In Advanced Chart Options—Scale—Y Axis, to set the minimum axis value, in Min and Max, select Min. Then select a minimum value.

5 To set a maximum value, in Min and Max, select Max. Then select a maximum value.

6 To set the minimum or maximum axis value for another axis, choose Overlay or Study, then repeat steps 4 and 5 for that axis.

7 Choose OK.

8 In Chart Builder, choose OK.

How to set the number of minor ticks on an axis

1 In Chart Builder, choose Advanced.

2 Advanced Chart Options, choose Scale.

3 In Advanced Chart Options—Scale, choose Y Axis.

4 In Advanced Chart Options—Scale—Y Axis, in Minor, select the number of tick marks to display in each major interval.

5 To set the number of minor ticks on another axis, choose Overlay or Study, then repeat step 4 for that axis.

6 Choose OK.

7 In Chart Builder, choose OK.

Working with axis labels

When you create a chart, chart axes display a label at each major interval. You can reposition axis labels. You can also format label text. You can change the font

characteristics or the value format of axis labels. For example, you can display axis labels as currency or percent values. You cannot apply a number format to a string label value, such as a label that displays a static group name.

You can also modify category axis label text. In a chart that uses a category key, you can build a new label text expression. In a chart that does not use categories or a chart that uses multiple category expressions, you can create custom label text.

How to arrange axis labels

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Labels.
3. In Advanced Chart Options—Labels, choose X Axis.
4. In Advanced Chart Options—Labels—X Axis, in Position, select where axis labels appear. To hide axis labels, select No labels.
5. In Orientation, select the angle at which to display axis label text. To set a custom label orientation angle, select Custom, then select an angle setting.
6. To set the position of labels on another axis, choose Y Axis, Overlay, or Study, then repeat steps 4 and 5 for that axis.
7. Choose OK.
8. In Chart Builder, choose OK.

How to format axis labels

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Labels.
3. In Advanced Chart Options—Labels, choose X Axis.
4. In Advanced Chart Options—Labels—X Axis, change axis label font attributes:
    1. Choose the button below Font.
    2. In Font, set font attributes. Then choose OK.
5. To change the axis label value format, in Format, select a format from the drop-down list or type a custom format string.
6. To format labels on another axis, choose Y Axis, Overlay, or Study, then repeat steps 4 and 5 for that axis.
7. Choose OK.
8. In Chart Builder, choose OK.

How to set axis label text for a chart that uses a category key

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Labels.
3. In Advanced Chart Options—Labels, choose X Axis.
4. In Advanced Chart Options—Labels—X Axis, in Label text, set label text options:
   - To use automatic label text, select Automatic.
   - To display a field value as label text, select Label expression. Then select a field from the drop-down list.
   - To create an expression to use as label text, select Label expression. Then choose Expression Builder.
5. To set label text on another axis, choose Y Axis, Overlay, or Study, then repeat step 4 for that axis.
6. Choose OK.
7. In Chart Builder, choose OK.

How to create custom labels for a chart that uses one or multiple category expressions

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Labels.
3. In Advanced Chart Options—Labels, to create custom labels for the *x*-axis, choose X Axis.
4. Advanced Chart Options—Labels—X Axis, in Label text, choose Edit list.
5. In Custom Categories Labels, modify the category labels, as shown in Figure 8-21:

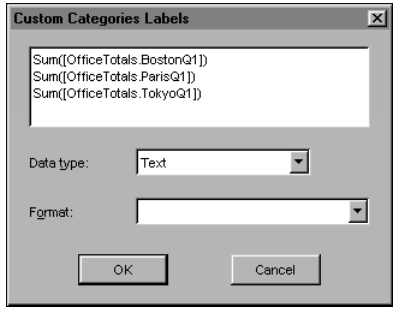


Figure 8-21 Modifying category labels

- To change the label for a category, select the category. Then type your changes.

- To change the data type of the labels, select a category. Then select a new data type.
- To change the number format of the labels, select a category. Then select the new format or type a format string.

Choose OK.

6 In Advanced Chart Options—Labels—X Axis, choose OK.

7 In Chart Builder, choose OK.

Working with series and categories

You can change the appearance of a chart series or category. For example, you can modify a series color or name or set the space between categories on a bar chart. You can change how Chart Builder plots empty values in a series. You can also add chart parts, such as point labels or high-low lines. In a line or an area chart, you can display data points and category labels in line with or between category axis divisions. You use Advanced Chart Options to work with series and categories.

Modifying series labels or series color

When you create a chart, Chart Builder creates series and category labels using the data that appears in the chart. If the chart uses one series or plots multiple value expressions as series, you can create new series label text. You can also change the data type of the label. If you select the date-and-time data type, you can select a number format for the label.

Chart Builder uses default colors for chart series. You can use Advanced Chart Options—Series to change the color of a chart series. You cannot change the color of the series in a stock chart. You can change the color of up or down bars in a stock chart.

How to modify a series label in an area, bar, or step chart

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Series.

3 In Advanced Chart Options—Series, to modify a series in the base chart, choose Y Axis. Figure 8-22 shows Advanced Chart Options—Series—Y Axis for a bar chart that plots four value expressions as series.

4 In Advanced Chart Options—Series—Y Axis, to edit a series label, select the series name, then type your changes.

5 To change a series color:

1 Choose Fill Color for that series row.

2 In the color selector, select a series color option. To use default colors for the series, select Automatic.

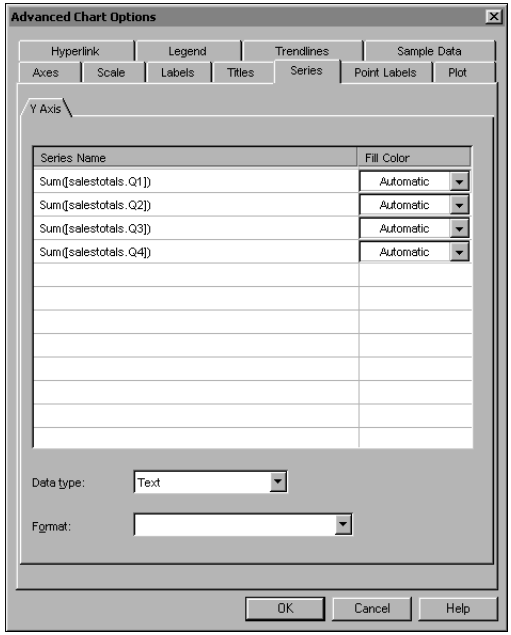


Figure 8-22 Advanced Chart Options—Series—Y Axis

6 To update the data type and format of the series:

1 In Data type, select a new data type for the series.

2 In Format, select a new number format for the series, or type a format string.

7 To modify a series in an overlay or study chart layer, choose Overlay or Study, then repeat steps 4 through 6 for that axis.

8 Choose OK.

9 In Chart Builder, choose OK.

How to modify a series label in a line or scatter chart

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Series.

3 In Advanced Chart Options—Series, to modify a series on the base chart, choose Y Axis.

4 In Advanced Chart Options—Series—Y Axis, to edit a series label, select the series name, then type your changes.

5 To change the appearance of a series line:

   1 Choose Line Style for that series.

   2 In Line Style, set line style options:

      - To use default line settings, select Automatic.
      - To hide lines, select No line.
      - To specify custom line style settings, choose Custom, then select line thickness, color, and pattern options.

   Choose OK.

6 To update the data type and format of the series:

   1 In Data Type, select a new data type for the series.

   2 In Format, select a new number format for the series or type a format string.

7 To modify a series in an overlay or study chart layer, choose Overlay or Study, then repeat steps 2 through 4 for that axis.

8 Choose OK.

9 In Chart Builder, choose OK.

Working with point labels

You can use point labels to display text that describes chart values. When you use point labels, the chart displays a label for each data point on the chart. You can determine how Chart Builder creates point label text. You can also format and position point labels. On a pie chart, you can use point label lines, which connect each data point to the associated point label.

How to set point label text for an area or step chart

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Point Labels.

3 In Advanced Chart Options—Point Labels, to set point label text on the *y*-axis, choose Y Axis.

4 In Advanced Chart Options—Point Labels—Y Axis, select Show point labels.

5 In Source, select a point label text option:

   - To display the point value, select Value.
   - To display the category to which the point belongs, select Category.
   - To display the series to which the point belongs, select Series.
   - To display the area value as a percentage of the whole, select Percentage.

- If you select Value, Category, or Series and you want to display the percentage in the label, select Append percentage.

6 Choose OK.

7 In Chart Builder, choose OK.

How to set point label text for a bar, line, or stock chart

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Point Labels.

3 In Advanced Chart Options—Point Labels, to set point label text on the *y*-axis, choose Y Axis.

4 In Advanced Chart Options—Point Labels—Y Axis, select Show point labels.

5 In Source, select a point label text option:

- To display the point value, select Value.
- To display the category to which the point belongs, select Category.
- To display the series to which the point belongs, select Series.

6 To set point label text for an overlay or study chart layer, choose Overlay or Study, then repeat steps 4 and 5 for that axis.

7 Choose OK.

8 In Chart Builder, choose OK.

How to set point label text for a scatter chart

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Point Labels.

3 In Advanced Chart Options—Point Labels, to set point label text on the *y*-axis, choose Y Axis.

4 In Advanced Chart Options—Point Labels—Y Axis, select Show point labels.

5 In Source, select a point label text option:

- To display the *y*-axis value of the data point, select Y value.
- To display the *x*-axis value of the data point, select X value.

6 To set point label text on another axis, choose Overlay or Study, then repeat steps 4 and 5 for that axis. Choose OK.

7 In Chart Builder, choose OK.

How to set point label text for a pie chart

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Point Labels.

3 In Advanced Chart Options—Point Labels, select Show point labels.

4 In Source, select a point label text option:

- To display the value of the sector, select Value.
- To display the name of the sector, select Slice name.
- To display the sector value as a percentage of the pie in which it appears, select Percentage.
- If you select Value or Slice name and you want to display the sector as a percentage of the pie in which it appears, select Append percentage. The percentage appears with the sector name or value.

5 Choose OK.

6 In Chart Builder, choose OK.

How to show point label lines on a pie chart

Point label lines are lines that point from the pie sector to its label.

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Point Labels.

3 In Advanced Chart Options—Point Labels, select Show point label lines. Choose OK.

4 In Chart Builder, choose OK.

How to format and position point labels

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Point Labels.

3 In Advanced Chart Options—Point Labels, to format point labels for a base chart, choose Y Axis.

4 In Advanced Chart Options—Point Labels—Y Axis, to change point label font attributes:

   1 Choose the button to the right of Font.

   2 In Font, set font attributes. Then choose OK.

5 To change the point label value format, in Format, select a format option.

6 Set point label position options:

- To change the angle at which point label text appears, in Orientation, select an option. To use a custom angle, select Custom, then select an angle value.

- To specify where on each series a point label appears, in Placement, select a position option. If Placement is inactive, the point label appears

7 To format and position point labels on another axis, choose Overlay or Study, then repeat steps 4 through 6 for that axis. Choose OK.

8 In Chart Builder, choose OK.

Positioning data points and axis labels on an *x*-axis

In an area or a line chart, you can align data points and *x*-axis labels with *x*-axis tick marks or display category markers and labels in between *x*-axis tick marks.

Figure 8-23 and Figure 8-24 show the difference between categories in line with tick marks and categories between tick marks on a sample line chart.

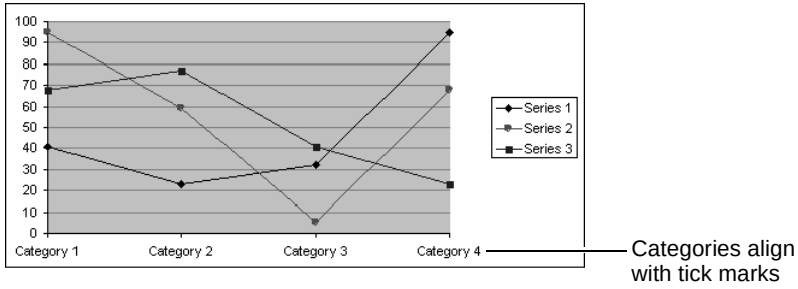


Figure 8-23 Categories in line with tick marks

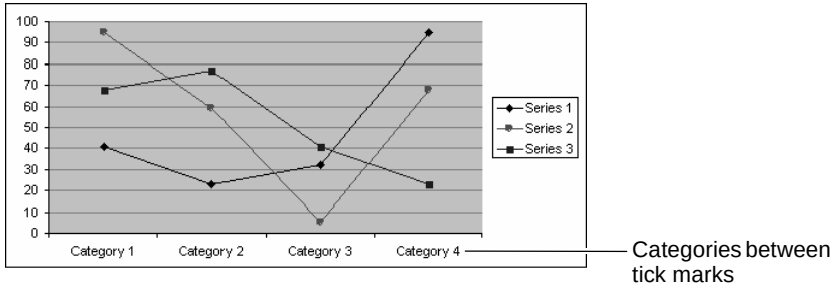


Figure 8-24 Categories between tick marks

How to set the position of category markers and labels

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Plot.

3 In Advanced Chart Options—Plot—Y Axis, set category marker and label position:

- To display category markers and labels between tick marks, select Categories are between ticks.

- To align category markers and labels with tick marks on an *x*-axis, deselect Categories are between ticks.

Choose OK.

4 In Chart Builder, choose OK.

Working with an empty value in a series

You can choose how you want Chart Builder to treat an empty value in chart data. Empty values do not appear in stock or pie charts.

Chart Builder offers three options for displaying empty values in charts:

- Break line. No data point appears on the chart, as shown in Figure 8-25. In a bar chart, this option appears as Do not plot. Area and step charts do not support the Break line option.

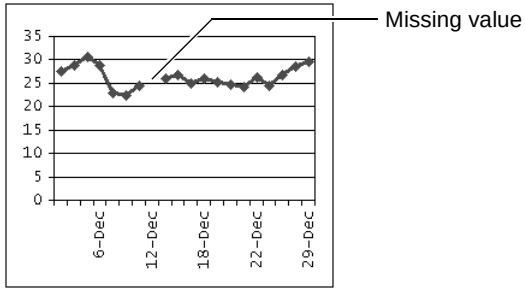


Figure 8-25 Break line in a chart

- Plot as zero value. Chart Builder plots the data point at 0 as shown in Figure 8-26.

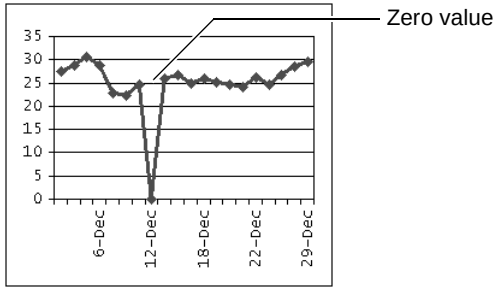


Figure 8-26 Plot as zero value

- Interpolated. Chart Builder calculates the missing value based on the surrounding data points, as shown in Figure 8-27. Bar and stock charts do not support the Interpolated option.

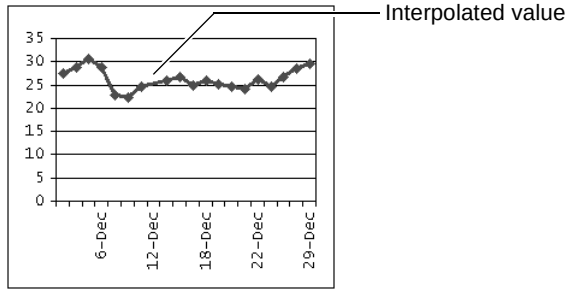


Figure 8-27 Interpolated

How to set how a chart uses empty values

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Plot.
3. In Advanced Chart Options—Plot, to set how the *y*-axis plots empty values, choose Y Axis.
4. In Advanced Chart Options—Plot—Y Axis, in Empty Values, select an option.
5. To set how an overlay or study chart layer uses empty values, choose Overlay or Study, then repeat step 4 for that axis. Choose OK.
6. In Chart Builder, choose OK.

Displaying data point markers or series lines

When you use a line or a scatter chart, you can use data point markers to identify data points. When you use data point markers, the chart displays a marker for each data point on the chart.

You can also use a line to connect the data points in each series. When you use series lines, the chart displays lines for all series in a chart, except series for which you define a custom series line style that does not show lines.

How to display data point markers or series lines

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Plot.
3. In Advanced Chart Options—Plot, to set data point markers or series lines on the *y*-axis, choose Y Axis.
4. In Advanced Chart Options—Plot—Y Axis, to display data point markers, select Show markers.
5. To display series lines, select Show lines.

6 To display markers on an overlay or study chart layer, choose Overlay or Study, then repeat steps 4 and 5 for that axis. Choose OK.

7 In Chart Builder, choose OK.

Working with up or down bars

An up or down bar is a bar on a line or a candlestick stock chart that extends from the first series in a category to the last series in a category. If the last series value is higher than the first series value, the bar is called an up bar. If the first series value is higher than the last series value, the box is called a down bar.

When you use up or down bars, the chart displays one up or down bar for each category in the chart. By default, up bars appear white and down bars appear black. You can change the color of the up or down bars in a chart.

How to add or modify up or down bars

1 In Chart Builder, choose Advanced.

2 Choose Plot.

3 In Advanced Chart Options—Plot, to add or modify up or down bars on the *y*-axis, choose Y Axis.

4 In Advanced Chart Options—Plot—Y Axis, select Up-down bars, as shown in Figure 8-28.

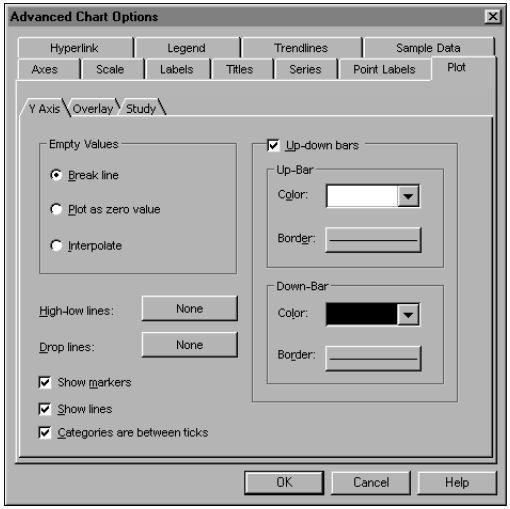


Figure 8-28 Selecting Up-down bars

5 In Up-bar, set attributes for bars for which the last series value is higher than the first series value:

1 In Color, select an up bar color.

2. To change the up bar border, choose Border.
3. In Border, select border options. Choose OK.

6. In Down-bar, set attributes for bars for which the first series value is higher than the last series value:
   1. In Color, select a down bar color.
   2. To change the down bar border, choose Border.
   3. In Border, select border options. Choose OK.
7. To add or modify up or down bars on an overlay or study chart layer, choose Overlay or Study, then repeat steps 4 through 6 for that axis.
8. Choose OK.
9. In Chart Builder, choose OK.

Working with high-low lines

A high-low line is a line on a line or stock chart that extends from the highest value in a category to the lowest value in a category. When you use high-low lines, the chart displays one high-low line for each category in the chart.

How to add high-low lines to a chart

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Plot.
3. In Advanced Chart Options—Plot, to add or modify high-low lines on the *y*-axis, choose Y Axis.
4. In Advanced Chart Options—Plot—Y Axis, choose the button to the right of High-low lines.
5. In Line Style, set line style options. Then choose OK.
6. To add or modify high-low lines on an overlay or study chart layer, choose Overlay or Study, then repeat steps 4 and 5 for that axis.
7. Choose OK.
8. In Chart Builder, choose OK.

Working with drop lines

A drop line is a line on a line or stock chart that extends from a data point to the x-axis. When you use drop lines, the chart displays one drop line for each data point on the chart.

How to add drop lines to a chart

1. In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Plot.

3 In Advanced Chart Options—Plot, to add or modify drop lines on the *y*-axis, choose Y Axis.

4 In Advanced Chart Options—Plot—Y Axis, choose the button to the right of Drop lines.

5 In Line Style, set line style options. Then choose OK.

6 To add or modify high-low lines on an overlay or study chart layer, choose Overlay or Study, then repeat steps 4 through 5 for that axis.

7 In Advanced Chart Options—Plot, choose OK.

8 In Chart Builder, choose OK.

Working with trendlines

Trendlines are used to graphically display trends in data. This type of analysis is also called regression analysis.

You can add trendlines to data series in unstacked 2-D area, bar, line, stock, scatter, and step charts. You cannot add trendlines to data series in 3-D, stacked, or pie charts, or charts with depth. If a chart does not support trendlines, the Trendlines tab does not appear in Advanced Chart Options.

When you add a trendline to a chart, you can choose any of the six different trend or regression types. The type of data that you have determines the type of trendline you should use. For an example of a chart that uses a trendline, see <eRDPro_HOME>\Examples\DesignAndLayout\Charts\Trendlines\Trendlines.rod.

About linear trendlines

A linear trendline is a best-fit straight line that is used with simple linear data sets. Your data is linear if the pattern in its data points resembles a line. A linear trendline usually shows that something is increasing or decreasing at a steady rate. Figure 8-29 shows a linear trendline clearly illustrating that refrigerator sales have consistently risen over a 13-year period.

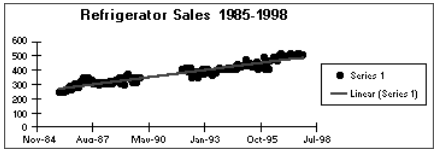


Figure 8-29 Linear trendline

This option creates a trendline by using the following linear equation to calculate the least squares fit for a line:

$y\ 0\ mx + b$

where m is the slope and b is the y-intercept.

About logarithmic trendlines

A logarithmic trendline is a best-fit curved line that is most useful when the rate of change in the data increases or decreases quickly and then levels out. A logarithmic trendline can use negative and/or positive values. Figure 8-30 uses a logarithmic trendline to illustrate predicted population growth for animals in a fixed-space area, where population levels out as space for the animals decreases.

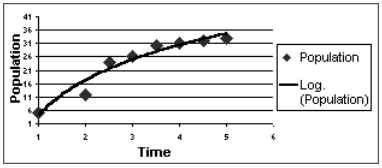


Figure 8-30 Logarithmic trendline

This option creates a logarithmic trendline by using the following equation to calculate the least squares fit through points:

$y = c \ln x + b$

where c and b are constants, and ln is the natural logarithm function.

About polynomial trendlines

A polynomial trendline is a curved line that is used when data fluctuates. It is useful, for example, for analyzing gains and losses over a large data set. The order of the polynomial can be determined by the number of fluctuations in the data or by how many bends, or hills and valleys, appear in the curve. An Order 2 polynomial trendline generally has only one hill or valley. Order 3 generally has one or two hills or valleys. Order 4 generally has up to three. Setting the order of a polynomial too high for the number of data points usually results in wild oscillations.

Figure 8-31 shows an Order 2 polynomial trendline, one hill, to illustrate the relationship between speed and gasoline consumption.

This option creates a polynomial or curvilinear trendline by using the following equation to calculate the least squares fit through points:

$y = b + c_1x + c_2x^2 + c_3x^3 + \ldots + c_6x^6$

where b and $c_1 \ldots c_6$ are constants.

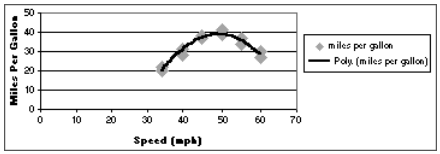


Figure 8-31 Order 2 polynomial trendline

About power trendlines

A power trendline is a curved line that is best used with data that increases at a specific rate, for example the speed of a race car at 1-second intervals. You cannot create a power trendline if your data contains zero or negative values. In the following example, the chart plots distance traveled in meters against time elapsed in seconds. The power trendline illustrates acceleration, as shown in Figure 8-32.

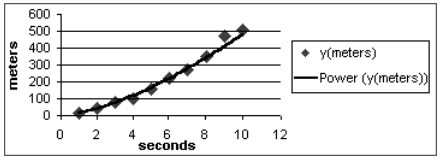


Figure 8-32 Power trendline

This option creates a power trendline by using the following equation to calculate the least squares fit through points:

$$y = cx^b$$

where c and b are constants.

About exponential trendlines

An exponential trendline is a curved line that is most useful when data values rise or fall at increasingly higher rates. You cannot create an exponential trendline if your data contains zero or negative values. In Figure 8-33 an exponential trendline is used to illustrate the decreasing amount of carbon 14 in an object as it ages.

This option creates an exponential trendline by using the following equation to calculate the least squares fit through points:

$$y = ce^{bx}$$

where c and b are constants, and e is the base of the natural logarithm.

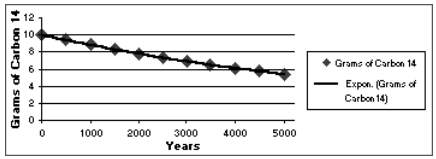


Figure 8-33 Exponential trendline

About moving average trendlines

A moving average trendline smooths out fluctuations in data to show a pattern or trend more clearly. A moving average uses a specific number of data points (set by the Period option), averages them, and uses the average value as a point in the line. The number of points in a moving average trendline equals the total number of points in the series less the number that you specify for the period. If Period is set to 2, for example, then the average of the first two data points is used as the first point in the moving average trendline. The average of the second and third data points is used as the second point in the trendline, and so on. In Figure 8-34 a moving average trendline shows a pattern in number of homes sold over a 26-week period.

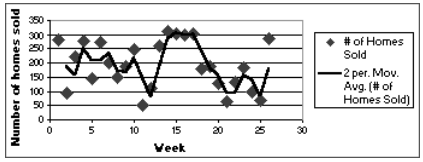


Figure 8-34 Moving average trendlines

If you add a moving average to a scatter chart, the moving average is based on the order of the x values plotted in the chart. To get the result that you want, you might need to sort the x values before adding a moving average.

This option creates a moving average trendline by using the following equation:

$$F_t = \frac{A_t + A_{t-1} + \ldots A_{t-n+1}}{n}$$

How to add trendlines to a chart

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Trendlines.
3. In Advanced Chart Options—Trendlines, to add a trendline for a *y*-axis series, choose Y Axis.

4. In Advanced Chart Options—Trendlines—Y Axis, choose New.
5. In Trendline Properties:
   1. In Series, choose the series for which you want to create a trendline.

2 In Name, type a name for the trendline. This name appears in the chart's legend.

3 In Type, choose the trendline's type. If you choose Moving Average, type the period in Period. If you choose Polynomial, type the order in Order.

4 To specify the *y*-intercept for a linear, polynomial, or exponential trendline, select Intercept and type the value. The *y*-intercept is included in the calculation of the trendline. The trendline does not intersect the *y*-axis.

5 To specify the thickness, color, and pattern for the trendline, choose the line next to Line Style and make your selections, as shown in Figure 8-35. Choose OK.

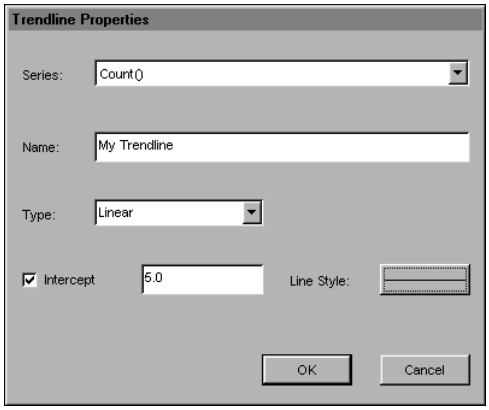


Figure 8-35 Trendline Properties

6 Choose OK. The names of the trendline and the corresponding series appear in Advanced Chart Options—Trendlines—Y Axis, as shown in Figure 8-36.

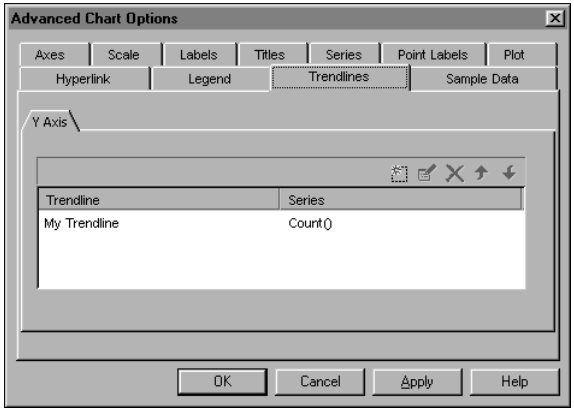


Figure 8-36 Advanced Chart Options

6 In Advanced Chart Options—Trendlines, choose OK.

7 In Chart Builder, choose OK.

Working with bars in a chart

You can change the appearance or position of bars in a bar chart or a bar chart with depth. You can modify the bar shape or outline style or display each bar as a line. In a stacked or percent bar chart, you can use a series line to connect the tops of the bars in a series.

You can also determine how the chart arranges bars relative to one another. You can set the following bar spacing options:

- Category gap. The space that appears between categories. By default, the category gap is 100, meaning that the space between categories is 100% of the width of a bar.
- Overlap gap. In a bar chart without depth, the space that appears between bars in each category. A positive overlap gap value means bars overlap. A negative overlap value means space appears between bars. The default overlap gap value for a chart in which bars appear next to one another is 0, which means there is no space between the bars. The default overlap gap value for a stacked bar chart or a stacked percent bar chart is 100, which means the bars appear on top of one another.

For example, the chart in Figure 8-37 uses a category gap of 100 and an overlap gap of 0.

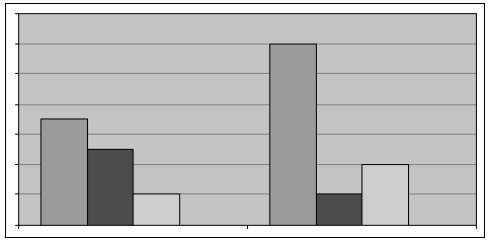


Figure 8-37 Example 1 of category gap and overlap gap

The chart in Figure 8-38 uses a category gap of 50 and an overlap gap of 25.

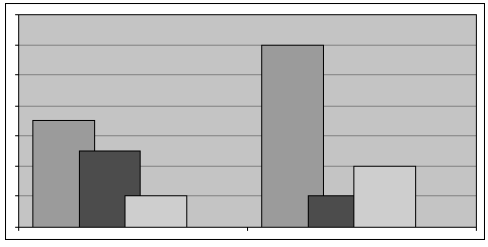


Figure 8-38 Example 2 of category gap and overlap gap

If you use an overlay or study chart layer, the spacing that you select for the base chart applies to the overlay or study chart layer as well.

How to format bars in a bar chart

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Plot.
3. In Advanced Chart Options—Plot, to format bars on the *y*-axis, choose Y Axis.
4. In Advanced Chart Options—Plot—Y Axis, to modify the bar border:
    1. Choose the button to the right of Bar outline style.
    2. In Border, to display bars without outlines, select No line.
5. To use a series line to connect the top of each bar in a series, select Series lines.
6. To display each bar as a line, select Show bars as lines.
7. To modify the bars on an overlay or study chart layer, choose Overlay or Study, then repeat steps 4 through 6 for that axis.
8. Choose OK.
9. In Chart Builder, choose OK.

How to format bars in a bar chart with depth

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Plot.
3. In Advanced Chart Options—Plot, to format bars on the *y*-axis, choose Y Axis.
4. In Advanced Chart Options—Plot—Y Axis, to change the shape of the bars, in Bar top shape, select an option:
    - To display each bar as a rectangle, select Square.
    - To display each bar as a cylinder, select Round.
    - To display each bar as a pyramid, select Triangle.
    - To display bars without depth, select Flat.
5. To modify the bar border:
    1. Choose the button below Bar outline style.
    2. In Border, specify border attributes. Choose OK.
6. To modify the bars on an overlay or study chart layer, choose Overlay or Study, then repeat steps 4 and 5 for that axis.
7. Choose OK.
8. In Chart Builder, choose OK.

How to position bars on a bar chart

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Plot.
3. In Advanced Chart Options—Plot—Y Axis, set the distance between categories by selecting a percent value in Category gap.
4. To overlap the bars in each category in a bar chart without depth, select a percent value in Overlap.
5. Choose OK.
6. In Chart Builder, choose OK.

Working with pie chart sectors

In a pie chart, each category appears as a sector of the pie. You can change the following sector attributes:

- Sector color and name. You can set the color of each sector in a pie chart. You can also modify the name of a sector.
- Sector explosion. You can display pie sectors touching one another, or you can display one or more of the pie sectors pulled away from the rest of the pie. Pulling one or more sectors of the pie from the rest of the pie is called exploding the sectors.
- Sector start angle. Chart Builder displays pie sectors clockwise in the same order as they appear in the chart's data source. By default, the edge of the first sector starts at 0°, or the 12 o'clock position. You can change the start angle for sectors on a pie chart.
- Sector outline style. You can change the thickness, color, and pattern of sector borders, or display a border shadow.

How to change the color or name of a sector in a pie chart

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Colors.
3. In Advanced Chart Options—Colors, to modify a sector label, select the sector name, then type your changes. Choose OK.
4. To change a sector color:
    1. Choose Fill Color for that sector row.
    2. In the color selector, select a sector color.
5. To change the data type or format of the sectors:
    - Select a new data type.

- Select a new number format, or type a format string.

6 In Chart Builder, choose OK.

How to explode a pie sector

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options—Colors, choose Plot.

3 In Advanced Chart Options—Plot, select Explode.

4 In Amount, specify the percent of the pie's radius to pull each exploded sector away from the pie. You can use a value between 0 and 40%.

5 To explode each sector equally, select Explode all slices.

6 To explode only pie sectors that occupy a certain percent of the pie area:

   1 Select Explode slices whose value is.

   2 Select an operator, select a percent value to which to compare the pie sector, then select Percent.

7 To explode pie sectors based on a comparison to a certain value:

   1 Select Explode slices whose value is.

   2 Select an operator, select a value to which to compare the pie sector, then deselect Percent.

   For example, using the settings in Figure 8-39, the pie chart explodes all slices to 15% of the pie's radius.

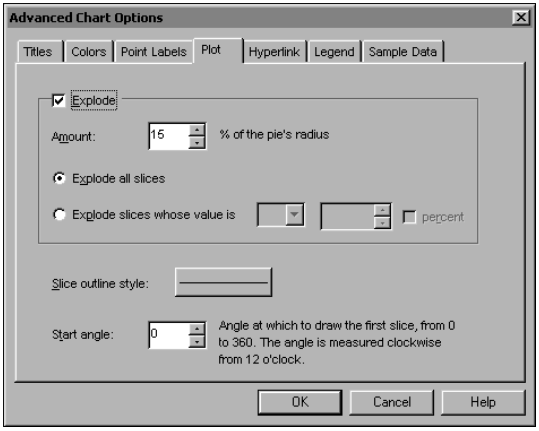


Figure 8-39 Advanced Chart Options for a pie chart

Choose OK.

8 In Chart Builder, choose OK.

How to set sector outline style

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Plot.
3. In Advanced Chart Options—Plot, choose the button to the right of Slice outline style.
4. In Border, set border attributes. Choose OK.
5. In Advanced Chart Options—Plot, choose OK.
6. In Chart Builder, choose OK.

How to set the pie sector start angle

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Plot.
3. In Advanced Chart Options—Plot, in Start angle, select the angle value at which to display the first sector in the chart. Choose OK.
4. In Chart Builder, choose OK.

Using a chart hyperlink

You can use a chart hyperlink to link from a part of a chart to the associated report data. You can type a hyperlink value or use Hyperlink Builder to create a hyperlink.

How to create a chart hyperlink

1. In Chart Builder, choose Advanced.
2. In Advanced Chart Options, choose Hyperlink.
3. In Advanced Chart Options—Hyperlink:
   - To use a link from any part of a chart, in Chart link, type the link URL, or choose Builder to use Hyperlink Builder to create the link.
   - To use a link from any of the data points in a category, in Category link, type the link URL, or choose Builder to use Hyperlink Builder to create the link.
   - To use a link from any of the data points in a series, in Series link, type the link URL, or choose Builder to use Hyperlink Builder to create the link.
   - To use a link from any data point, in Point link, type the link URL, or choose Builder to use Hyperlink Builder to create the link.
   - To display the link location in a different window from the active window, in Target window name, type the window name.

4 Choose OK.

5 In Chart Builder, choose OK.

How to use Hyperlink Builder to create a link

1 In Advanced Chart Options—Hyperlink, choose Builder.

2 In Hyperlink Builder, in URL, type a link expression. Choose OK.

Working with sample chart data

You can change the number of series or categories a chart control displays in the layout window. If you use multiple value expressions as series, the chart design always displays a number of series equal to the number of series expressions. If you use multiple value expressions as categories, the chart design always displays a number of categories equal to the number of category expressions.

The procedure you use to modify the sample chart data for a scatter chart is slightly different from the procedure that you use to modify the sample chart data for other chart types. You use Advanced Chart Options to work with sample chart data.

How to modify sample chart data for an area, bar, line, pie, step, or stock chart

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Sample Data.

3 In Advanced Chart Options—Sample Data:

- In Number of categories, select the number of categories to display in the chart control.
- In Number of series, select the number of series to display in the chart control.

4 An overlay or study chart layer always displays the same number of sample chart categories as the base chart layer. To change the number of series an overlay or study chart layer displays, choose Overlay or Study, then repeat step 3 for that axis.

5 Choose OK.

6 In Chart Builder, choose OK.

How to modify sample chart data for a scatter chart

1 In Chart Builder, choose Advanced.

2 In Advanced Chart Options, choose Sample Data.

3 In Advanced Chart Options—Sample Data:

- In Number of points per series, select the number of markers to display for each series in the chart control.
- In Number of series, select the number of series to display in the chart control.

4. An overlay or study chart layer always displays the same number of points per series as the base chart layer. To change the number of series an overlay or study chart layer displays, choose Overlay or Study, then repeat step 3 for that axis.
5. Choose OK.
6. In Chart Builder, choose OK.

Rendering a chart on Windows and UNIX

If you use both Windows and UNIX platforms to deploy reports, choose fonts that are available on both Windows and UNIX when you design the report. Unavailable fonts can adversely affect the appearance of report object instance (.roi) files.

For the charting engine to render the fonts, the referenced fonts must be available to the JVM used for the chart classes. To make fonts available on your platform, complete the following tasks:

- On Windows, install referenced fonts in the standard fonts folder for the operating system, such as \Winnt\Fonts.
- On UNIX, install referenced fonts in $JAVA_HOME/jre/jre/lib/fonts.

Actuate uses default fonts for those parts of a chart that use a font not found on the operating system. The default font that Actuate uses is dependent on the locale specified when loading the chart classes. For each locale, you must have at least one font. For UNIX and Windows output to look identical, the fonts listed in the following table must be available to the UNIX JVM.

To view charts on UNIX, you need an x-server. You can use the virtual x-server Xvfb that Actuate ships. Xvfb supports a number of fonts. Alternatively, you might need to install your own fonts.

If a specific font is not available, you can use one of the fonts listed in Table 8-1.

Table 8-1 UNIX font table

Locale specified	Default fonts
For all languages not specified later in this table	arial.ttf arialbd.ttf arialbi.ttf ariali.ttf
Chinese, simplified	simsun.ttc
Chinese, traditional	mingliu.ttc
Japanese	msgothic.ttc msmincho.ttc
Korean	gulim.ttc batang.ttc

Chapter

9

Drawing with SVG

This chapter contains the following topics:

- Working with SVG
- Supporting SVG in e.Report Designer Professional
- Examining the SVG code in a drawing control
- Debugging an SVG program

Working with SVG

Scalable Vector Graphics, or SVG, is a language for describing two-dimensional graphics in XML. SVG is a W3C standard which builds on other standards such as CSS. With SVG you can combine complex gradients with filters and bitmap graphics to create both realistic and abstract images. The W3C specification for SVG can be found at http://www.w3.org/TR/SVG11/index.html. The CSS online specification is at http://www.w3.org/TR/REC-CSS2/cover.html. SVG has a rich variety of graphic abilities. e.Report Designer Professional provides several examples of SVG applications. There are also many resources available to help you learn this language. Figure 9-1, shows images that have been created with SVG.

Figure 9-1 Images created with SVG

SVG is an XML-based language. The following SVG example draws a rectangle:

```
<svg>
   <rect
      x='10%' y='10%'
      width='80%' height='80%'
      fill='red'
      stroke='black' stroke-width='4pt'
   />
</svg>
```

While SVG is used to describe graphics, it is also a language that is intended to work in conjunction with other languages, in that it exposes graphical objects so that other applications can use them. It is an open standard that has numerous tools emerging to support it.

In a report design, SVG images can be placed in a chart, drawing or browser scripting control. You send SVG to a browser scripting control when you want a browser that supports SVG to render the image. e.Report Designer Professional supports the following SVG primitives:

- Lines, rectangles, polygons, circles, ellipses, arcs, B-spline curves
- Text
- Fills, patterns, textures

- Gradients, filters, lighting effects
- Any combination of the above

For example, you can use SVG to create rotated text for column headers, corporate logos, diagrams, annotations on charts, specialized chart types and dashboard components.

SVG is supported in several web browsers. SVG viewing applications, which allow you to view SVG images, are available on the web.

Supporting SVG in e.Report Designer Professional

Support of SVG consists of several Actuate Basic functions and an AFC class, AcDrawingSVGPlane. These work in conjunction with the methods of a drawing or chart control. The Actuate Basic functions provide a string in the locale-neutral format that SVG requires, and are described in Table 9-1. You use these functions to create attributes or parameters in the proper format as required for SVG. If you are using a non-neutral locale, you must use these functions to make your SVG program function properly. In addition, some functions also simplify the creation of SVG code. For example, SVGFontStyle returns the correct string as required by SVG given the specified font parameters. The following code segments are equivalent:

```
svg = SVGFontStyle( "Label", pointLabelFont,
+                    "text-rendering:geometricPrecision;" )
```

and:

```
svg = "<style type='text/css'>" &
+       "<![CDATA[.Label{font-family:Arial;" &
+       "font-size:10.0px;font-weight:normal;" &
+       "font-style:normal;" &
+       "text-decoration:none;fill:#000000;" &
+       "stroke:none;text-rendering:geometricPrecision;}]]>" &
+       "</style>"
```

For a full description on the use of these functions, refer to *Programming with Actuate Basic*.

Table 9-1 Actuate Basic SVG functions

Function	Use
SVGAttr	Returns a string for setting attributes
SVGColorAttr	Returns a string for setting color attributes
SVGDbl	Returns a string for setting a numeric value

(continues)

Table 9-1 Actuate Basic SVG functions (continued)

Function	Use
SVGFontStyle	Returns a string for setting font values
SVGStr	Returns a string for setting a string value
SVGStyle	Returns a string for setting a CSS style

SVG drawings are created dynamically when the report is generated. They are rendered as a bitmap during viewing, and scale to the appropriate zoom factor or resolution. Users do not need an SVG viewer to view SVG drawings in a report. If you want to send SVG to a browser that supports SVG it must be embedded within a browser scripting control.

On Windows systems with a large size dpi setting, when you view a report within e.Report Designer Professional, the results may appear differently from how it appears when viewing the output in DHTML.

e.Report Designer Professional does not include an SVG editing tool, however various tools and plug-ins are available on the web. SVG animation and interactivity are not supported. Consult the examples installed with e.Report Designer Professional in <eRDPro_HOME>\Examples\DesignAndLayout \Charts and <eRDPro_HOME>\Examples\DesignAndLayout\Drawings. The chart examples that use SVG have names that begin with the word Drawing.

Understanding measurements and scaling

You use the viewBox attribute of the svg tag to apply a scale to an SVG drawing plane. This allows you to specify the default units. When you use viewBox, all of the absolute units of measures, such as inches, are scaled in the drawing. Use viewBox to scale the default units to points. Do not use explicit units.

The following code retrieves the size of the drawing, and then sets the default units in the SVG code to points. It then creates the SVG code to draw a rectangle, and sets the string containing the code into an AcDrawingSVGPlane.

```
' Get the size of the drawing in points
   Dim w As Double
   w = Size.Width / OnePoint
   Dim h As Double
   h = Size.Height / OnePoint

   Dim svg As String
```

```
   ' Scale the drawing to use points as the default units
   svg = "<svg version='1.1'"
+     ' Standard SVG 1.1 namespaces
+     & " xmlns='http://www.w3.org/2000/svg'"
+     & " xmlns:xlink='http://www.w3.org/1999/xlink'"
+     ' Do not collapse whitespace in text
+     & " xml:space='preserve'"
+     ' Scale the SVG to use points as the default units
+     & " viewBox='0 0 " & SVGDbl( w ) & " "
+     & SVGDbl( h ) & "'>"
    svg = svg
+     & "<rect x='10%' y='10%'"
+     & " width='" & (w * 0.8) & "'"
+     & " height='" & (h * 0.8) & "'"
+     & " fill='green' stroke='black' stroke-width='4'/>"
+     & "</svg>"

   Dim svgPlane As AcDrawingSVGPlane
   Set svgPlane = AddDrawingPlane( DrawingPlaneTypeSVG )
   svgPlane.SetSVG( svg )
```

Using drawing planes

SVG code is placed within drawing planes. A drawing may contain multiple drawing planes which are drawn on top of each other. Each drawing plane has its own content.

You must explicitly insert or append drawing planes in your code. There is no default drawing plane in a drawing. Any drawing plane can be positioned, sized, or hidden. You can delete drawing planes, except for a drawing plane that contains a chart.

Drawing planes are typically used in the Start( ), OnRow( ), or Finish( ) methods of a drawing control, or DrawOnChart( ) for a chart control. To create an SVG image, you place all the SVG into one string variable, and then set that into a drawing plane as shown in the following code:

```
Sub Finish( )
     Dim svg As String

' create SVG code here
     svg = …

     Dim svgPlane As AcDrawingSVGPlane
     Set svgPlane = AddDrawingPlane( DrawingPlaneTypeSVG )
     svgPlane.SetSVG( svg )

     Super::Finish( )
End Sub
```

For more information on the AcDrawing and AcDrawingSVGPlane classes, see *Programming with Actuate Foundation Classes.*

How to use SVG in a report design

1. Determine what SVG elements and filters you will use.
2. Drag a drawing or chart control from Toolbox into a frame.
3. Copy code from one of the examples and modify it as necessary. Report designs that use SVG install with e.Report Designer Professional in <eRDPro_HOME>\Examples\DesignAndLayout\Charts and <eRDPro_HOME>\Examples\DesignAndLayout\Drawings. The chart examples that use SVG have names that begin with the word Drawing.
4. Build up your SVG in stages, making sure you have a valid drawing at each stage.

If you run into problems, add code to dump your SVG to a file and use a tool like XML-SPY to validate your SVG. You may also try to view your SVG using an SVG viewer. See "Debugging an SVG program," later in this chapter, for more information on how to dump your SVG code to various locations.

Drawing on charts

AcChart is a subclass of AcDrawing. By default, a chart creates a single special drawing plane that displays the chart. This plane does not use SVG. You can insert additional drawing planes behind and in front of the chart plane. Drawing on charts may not appear 100% accurate when a report is viewed within e.Report Designer Professional.

You can hide the chart plane, but not delete it. To draw on a chart, you override the AcChart::DrawOnChart( ) method. In addition, you can use chart method calls to get the position and size of the plot areas or pie in a chart.

In e.Report Designer Professional, open <eRDPro_HOME>\Examples\DesignAndLayout\Charts\DrawingRadialPieLabels\DrawingRadialPieLabels.rod.

This example report shows how to modify a chart to make room for SVG drawn labels on a radial pie chart. When you open the report, it contains a single frame with a chart control in it. The control's DrawOnChart( ) method is overridden as follows:

```
Sub DrawOnChart( baseLayer As AcChartLayer, overlayLayer As
   AcChartLayer, studyLayers() As AcChartLayer )
   ' Shrink and recenter the pie to make room for the labels
   Dim pieScale As Double
   pieScale = 0.95
   Dim chartPlane As AcDrawingChartPlane
   Set chartPlane = GetChartDrawingPlane( )
```

```
   chartPlane.SetSize( Size.Width * pieScale,
+                      Size.Height * pieScale)
   chartPlane.SetPosition( (Size.Width * (1 - pieScale)) / 2,
+                          (Size.Height * (1 - pieScale)) / 2)

   ' Get the center and radius of the pie (in twips)
   DescribeLayout( )
   Dim pieCenter As AcPoint
   pieCenter = baseLayer.GetPieCenter( )
   Dim pieRadius As AcTwips
   pieRadius = baseLayer.GetPieRadius( )

   ' Get the width, height, and pie center in points
   Dim w As Double
   w = Size.Width / OnePoint
   Dim h As Double
   h = Size.Height / OnePoint
   Dim cX As Double
   cX = pieCenter.X / OnePoint
   Dim cY As Double
   cY = pieCenter.Y / OnePoint

   ' The labels will be spaced slightly outside the pie
   Dim labelRadius As Double
   labelRadius = (pieRadius / OnePoint) + 8

   Dim svg As String
   svg = "<svg version='1.1'"
+     ' Standard SVG 1.1 namespaces
+     & " xmlns='http://www.w3.org/2000/svg'"
+     & " xmlns:xlink='http://www.w3.org/1999/xlink'"
+     ' Do not collapse whitespace in text
+     & " xml:space='preserve'"
+     ' Scale the SVG to use points as the default units
+     & " viewBox='0 0 " & SVGDbl( w ) & " "
+     & SVGDbl( h ) & "'>"

   ' Define the point label font style; use geometric
   ' precision text rendering to get consistent appearance
   ' regardless of angle
   Dim pointLabelFont As AcFont
      pointLabelFont = baseLayer.GetSeriesStyle
+                   ( 1 ).GetPointLabelStyle( ).Font
   svg = svg
+     & "<defs>"
+     &   SVGFontStyle( "Label", pointLabelFont,
+                       "text-rendering:geometricPrecision;" )
+     & "</defs>"
```

```
   ' Get the height of the label font
Dim labelFontHeight As Double
labelFontHeight = GetFontDisplayHeight( pointLabelFont )
+                              / OnePoint

   Dim series As AcChartSeries
   Set series = baseLayer.GetSeries( 1 )
   Dim numberOfCategories As Integer
   numberOfCategories = baseLayer.GetNumberOfCategories( )

   Dim totalValue As Double
   totalValue = series.GetSumOfSliceValues( )

   Dim cumulativeValue As Double

   ' Set the coordinate origin to the pie center
   svg = svg
+    & "<g transform='"
+    & " translate(" & SVGDbl( cX ) & "," & SVGDbl( cY )
+    & ")" & "'>"

   Dim i As Integer
   For i = 1 To numberOfCategories
      Dim category As AcChartCategory
      Set category = baseLayer.GetCategory( i )
      Dim labelText As String
      labelText = category.GetLabelText( )Dim point As AcChartPoint
      set point = series.GetPoint( i )
      Dim pointValue As Double
      pointValue = point.GetYValue( )

      Dim sliceAngle As Double
      sliceAngle = (pointValue / totalValue) * 360

      Dim labelAngle As Double
      labelAngle = ((cumulativeValue / totalValue) * 360) +
+                    (sliceAngle / 2) - 90

      Dim r As Double
      Dim textAnchor As String
      If (labelAngle <= 90) Then
         r = labelRadius
         ' Text on the right is left-aligned
            textAnchor = " text-anchor='start'"
      Else
         ' Flip the labels on the left side
         labelAngle = labelAngle + 180
         r = -labelRadius
```

```
         ' Text on the left is right-aligned
         textAnchor = " text-anchor='end'"
      End If

      ' Draw the label rotated and offset to center
      ' it on the slice
      svg = svg
+        & "<text class='Label'"
+        & " transform='"
+        &  " rotate(" & SVGDbl( labelAngle ) & ")"
+        &  " translate(" & SVGDbl( r ) & ","
+        & SVGDbl( labelFontHeight * 0.31 ) & ")"
+        &  "'"
+        &   textAnchor
+        & ">"
+        &   SVGStr( labelText )
+        & "</text>"
      cumulativeValue = cumulativeValue + pointValue
   Next i

   svg = svg
+     & "</g>"
+     & "</svg>"

   Dim svgPlane As AcDrawingSVGPlane
   Set svgPlane = AddDrawingPlane( DrawingPlaneTypeSVG )
   svgPlane.SetSVG( svg )
End Sub
```

Utilizing chart information

To be able to draw on charts effectively, you must know the size and shape of the chart you are drawing on, or be able to modify the chart so that your drawings are appropriate for the chart. In the example code, the chart needs to be made smaller. The code to do that follows:

```
   Dim pieScale As Double
   pieScale = 0.95
   Dim chartPlane As AcDrawingChartPlane
   Set chartPlane = GetChartDrawingPlane( )
   chartPlane.SetSize( Size.Width * pieScale, Size.Height *
+                  pieScale)
   chartPlane.SetPosition( (Size.Width * (1 - pieScale)) / 2,
+                   (Size.Height * (1 - pieScale)) / 2)
```

This section of code retrieves the chart drawing plane. It then resets the size properties of the chart to be 95% of its original size by calling SetSize. It also then modifies the position of the chart so that it is again recentered.

Once the chart has been resized and recentered, we retrieve the layout of the chart as follows:

```
DescribeLayout( )
Dim pieCenter As AcPoint
pieCenter = baseLayer.GetPieCenter( )
Dim pieRadius As AcTwips
pieRadius = baseLayer.GetPieRadius( )
```

DescribeLayout is a method in AcChart that computes the layout of a chart without rendering it. You can get information about the chart's layout by calling DescribeLayout( ) then calling methods such as GetPieCenter to retrieve information about the chart being drawn. These methods can only be called from DrawOnChart( ). The available chart information methods are listed in Table 9-2.

Table 9-2 Chart layout methods

Method	Use
GetPieCenter	Returns the position of the center of a pie chart relative to the top left corner of its parent chart's chart drawing plane. You can use this method only for two-dimensional pie charts.
GetPieRadius	Returns the radius of a pie chart. You can use this method only for two-dimensional pie charts.
GetPlotAreaPosition	Returns the position of a chart layer's plot area relative to the top left corner of its parent chart's chart drawing plane. You can use this method only for two-dimensional charts that are not pie charts.
GetPlotAreaSize	Returns the size of a chart layer's plot area. You can use this method only for two-dimensional charts that are not pie charts.

A section of code that retrieves and uses chart information follows:

```
Dim series As AcChartSeries
Set series = baseLayer.GetSeries( 1 )
Dim numberOfCategories As Integer
numberOfCategories = baseLayer.GetNumberOfCategories( )
Dim totalValue As Double
totalValue = series.GetSumOfSliceValues( )
```

GetSeries retrieves information about a particular series within a layer. In this section of code, we retrieve the number of categories and the sum of the slice values for use later in the method for text positioning.

Finally, we retrieve the text and its position on the chart, as shown in the following code:

```
Dim category As AcChartCategory
Set category = baseLayer.GetCategory( i )
Dim labelText As String
labelText = category.GetLabelText( )

Dim point As AcChartPoint
set point = series.GetPoint( i )
```

For more information on the AcChart class, see *Programming with Actuate Foundation Classes.*

Using text metrics

In the previous example, GetFontDisplayHeight is used to properly size and place the text onto the chart. Text metric methods in Actuate Basic are available to get text sizing information which simplifies the task of placing text on drawings or charts. These functions return their values in twips, and should be converted to points for use in SVG. The available text metric methods are described in Table 9-3.

Table 9-3 Text metric methods

Method	Use
GetDisplayHeight	Returns the height required to display a given text string without truncating it
GetFontAverageCharWidth	Returns the average width of a character in a specified font
GetFontDisplayHeight	Returns the height needed to display a character in a specified font
GetTextWidth	Returns the width of a text string

For more information about text metric functions, and examples of text metric function use, see *Programming with Actuate Basic.*

Examining the SVG code in a drawing control

In e.Report Designer Professional, open <eRDPro_HOME>\Examples \DesignAndLayout\Drawings\3DEffect\3DEffect.rod.

This sample application shows how you can use SVG filters within a drawing to create a 3D Effect. When you open the report, it contains a single frame with a drawing control in it. The control's Finish( ) method is overridden as follows:

```
Sub Finish( )
   ' Get the size of the drawing in points
   Dim w As Double
   w = Size.Width / OnePoint

Dim h As Double
   h = Size.Height / OnePoint
   Dim svg As String
' Begin header
     svg = "<svg version='1.1'"
+       ' Standard SVG 1.1 namespaces
+       & " xmlns='http://www.w3.org/2000/svg'"
+       & " xmlns:xlink='http://www.w3.org/1999/xlink'"
+       ' Do not collapse whitespace in text
+       & " xml:space='preserve'"
+       ' Scale the SVG to use points as the default units
+       & " viewBox='0 0 " & SVGDbl( w ) & " " & SVGDbl( h )
+       & "'>"
' End header

' Begin defined reference section
    ' Define a combined drop shadow and specular highlight
    ' filter
     svg = svg
+       & "<defs>"
+       & "<filter id='3DFilter' filterUnits='userSpaceOnUse'"
+       & "width='100%' height='100%'>"
+       & "<feGaussianBlur in='SourceAlpha' "
+       & "stdDeviation='3' result='blur'/>"
+       & "<feOffset in='blur' dx='5' dy='5' "
+       & "result='offsetBlur'/>"
+       & "<feSpecularLighting in='blur' surfaceScale='5' "
+       & "specularConstant='0.75' "
+       & "specularExponent='20' lighting-color='#F8D8D0' "
+       & "result='specOut'>"
+       & "<fePointLight x='-2000' y='-4000' z='8000'/>"
+       & "</feSpecularLighting>"
+       & "<feComposite in='specOut' in2='SourceAlpha' "
```

```
+       & "operator='in' result='specOut'/>"
+       & "<feComposite in='SourceGraphic' in2='specOut' "
+       & "operator='arithmetic' k1='0' k2='1' k3='1' "
+       & "k4='0' result='litPaint'/>"
+       & "<feMerge>"
+       & "<feMergeNode in='offsetBlur'/>"
+       & "<feMergeNode in='litPaint'/>"
+       & "</feMerge>"
+       & "</filter>"
+       & "</defs>"
' End defined reference section

' Begin drawing functions
   ' Background rectangle
   svg = svg
+     & "<rect x='0.5' y='0.5'"
+     &   SVGAttr( "width", w - 1 )
+     &   SVGAttr( "height", h - 1 )
+     & " fill='#808080' stroke='#404040' stroke-width='1'/>"

   ' Offset and apply the filter
   svg = svg
+     & "<g transform='translate(20,20)'>"
+     &   "<g filter='url(#3DFilter)'>"

   ' Draw the outline
   svg = svg
+     & "<path fill='none' stroke='#B00000' "
+     & "stroke-width='10' "
+     & "d='M50,100 a50,50 0 1,1 0,-100 l130,"
+     & "0 a50,50 0 1,1 0,100 Z'/>"

   ' Draw the button
   svg = svg
+     & "<path fill='#B00000' stroke='none' "
+     & "d='M50,80 a30,30 0 1,1 0,-60 l130,0 a30,"
+     & "30 0 1,1 0,60 Z'/>"
+     & "</g>"

   ' Draw the text unfiltered
   svg = svg
+     & "<g font-size='36' font-family='Verdana' "
+     & "fill='#602020' stroke='black' "
+     & "stroke-width='0.67'>"
+     & "<text x='40' y='60'>Drawing</text>"
+     & "</g>"
+     & "</g>"
```

```
   svg = svg
+     & "</svg>"
' End drawing functions

   Dim svgPlane As AcDrawingSVGPlane
   Set svgPlane = AddDrawingPlane( DrawingPlaneTypeSVG )
   svgPlane.SetSVG( svg )

   Super::Finish( )
End Sub
```

The overridden Finish( ) method generates a string containing SVG commands and sets it into an SVG drawing plane. It then calls Super::Finish( ) to render the image onto the display.

The SVG string is in several sections, all concatenated together. This example has a header, defined reference, and drawing function sections.

Examining the SVG header information

The following SVG header contains information about name spaces and the coordinate system to be used.

```
svg = "<svg version='1.1'"
+  ' Standard SVG 1.1 namespaces
+  & " xmlns='http://www.w3.org/2000/svg'"
+  & " xmlns:xlink='http://www.w3.org/1999/xlink'"
+  ' Do not collapse whitespace in text
+  & " xml:space='preserve'"
+  ' Scale the SVG to use points as the default units
+  & " viewBox='0 0 " & SVGDbl( w ) & " " & SVGDbl( h )
+  & "'>"
```

This example uses two name spaces. The default namespace that identifies SVG, www.w3.org/2000/svg, and a linking namespace for creating references to existing SVG code, www.w3.org/1999/xlink.

Xml:space is an attribute that helps applications determine whether they should pay attention to white space. In this case, we want to preserve white space, and not collapse it.

The viewBox defines the internal coordinate system used by anything within the width and height of the SVG element. The values represent the coordinates of the left top corner, followed by the width and height of the bounding box. SVGDbl is called to convert the values of the size of the control to those appropriate to SVG.

Examining defined referenced elements

The defs element is a container element for referenced elements. These elements are not displayed when they are defined, but are used in references in later sections.

The following part of the example uses the defs section to create a chained filter that creates a drop shadow and lighting effects:

```
' Define a combined drop shadow and specular highlight filter
   svg = svg
+     & "<defs>"
+     & "<filter id='3DFilter' filterUnits='userSpaceOnUse'
           width='100%' height='100%'>"
+     & "<feGaussianBlur in='SourceAlpha' "
+     & "stdDeviation='3' result='blur'/>"
+     & "<feOffset in='blur' dx='5' dy='5' "
+     & "result='offsetBlur'/>"
+     & "<feSpecularLighting in='blur' surfaceScale='5' "
+     & "specularConstant='0.75' "
+     & "specularExponent='20' lighting-color='#F8D8D0' "
+     & "result='specOut'>"
+     & "<fePointLight x='-2000' y='-4000' z='8000'/>"
+     & "</feSpecularLighting>"
+     & "<feComposite in='specOut' in2='SourceAlpha' "
+     & "operator='in' result='specOut'/>"
+     & "<feComposite in='SourceGraphic' in2='specOut' "
+     & "operator='arithmetic' k1='0' k2='1' k3='1' "
+     & "k4='0' result='litPaint'/>"
+     & "<feMerge>"
+     & "<feMergeNode in='offsetBlur'/>"
+     & "<feMergeNode in='litPaint'/>"
+     & "</feMerge>"
+     & "</filter>"
+     & "</defs>"
```

A filter is a transformation that changes an image prior to it being rendered. A filter is prefixed with the characters fe which stands for filter effect. SVG has several different primitive filters that modify the image in various ways. They can be chained together, causing their effects to be cumulative. By applying these filters in different combinations and chains you can create a much larger set of usable filters. The filters used in this example are feGaussianBlur, feOffset, fePointLight, feSpecularLighting, feComposite, and feMerge.

The filter element itself appears as follows:

```
<filter id='3DFilter' filterUnits='userSpaceOnUse'
   width='100%' height='100%'>
```

The id attribute specifies the name of the filter, 3DFilter. This name is used to reference the filter in other sections of the SVG code. userSpaceOnUse specifies that the filter uses the coordinate system that is in effect when the filter is being called. Width and height specify how large an area the filter covers.

Following the filter tag, a set of filter primitives are used to modify the image, starting with feGaussianBlur. The key attributes for chaining filters are in and result. In is assigned the name of the image to modify, and result is where the transformed image is placed. The resulting image can then be used as input into another transformation. For example, feGaussianBlur takes SourceAlpha as its in value, and has blur as its result value. The filter feOffset then takes the blur result from feGaussianBlur as its in value, and produces its result in offsetBlur. The chain can continue on in this way until the final desired effect is achieved.

Two special values are used for the original source image for in values. They are SourceGraphic and SourceAlpha. SourceGraphic represents the original image that is being transformed, and SourceAlpha represents the alpha channel, or mask, of that image. A mask is a grayscale bitmap image. The pure white areas in the image represent the portions of your original image that are 100% protected when combined with a second image. The pure black portions of the image represent the areas of your original image that are completely masked out, or erased. Masks support the creation of soft fades, decorative edges, and translucent effects.

Examining drawing functions

After setting values for the header, and defining the filter, the elements are placed into the image. The first image to be drawn is a gray rectangle, as shown in the following snippet:

```
' Background rectangle
   svg = svg
+     & "<rect x='0.5' y='0.5'"
+     & SVGAttr( "width", w - 1 )
+     & SVGAttr( "height", h - 1 )
+     & " fill='#808080' stroke='#404040' stroke-width='1'/>"
```

There are several shape primitives in SVG, including rect, line, circle, ellipse, path, and polygon. A path represents the outline of a shape which can be filled, stroked, used as a clipping path, or any combination of the three. Once the background rectangle is drawn, two groups of elements are created with the g tag, as shown in the following snippet:

```
svg = svg
+  & "<g transform='translate(20,20)'>"
+  & "<g filter='url(#3DFilter)'>"
```

The first grouping translates all the elements within the group by 20 in both the x and y directions. The second grouping utilizes the 3DFilter created in the defs section. All geometry created within this grouping has the chained filter 3DFilter

applied to it. After the two path elements are created which draw the outline and the button, the section closes the grouping that has the filter applied to it.

```
' Draw the outline
svg = svg
+  & "<path fill='none' stroke='#B00000' "
+  & "stroke-width='10' "
+  & "d='M50,100 a50,50 0 1,1 0,-100 l130,"
+  & "0 a50,50 0 1,1 0,100 Z'/>"

' Draw the button
svg = svg
+  & "<path fill='#B00000' stroke='none' "
+  & "d='M50,80 a30,30 0 1,1 0,-60 l130,"
+  & "0 a30,30 0 1,1 0,60 Z'/>"
+  & "</g>"
```

Since the filter grouping is closed, the text that is drawn next is unfiltered, but still translated. However, before the text is drawn, a new group is created that specifies a font. The text is drawn in that font, and then the font group is closed. The translation font is next closed. Finally, the SVG tag is closed in the following snippet:

```
' Draw the text unfiltered
svg = svg
+  & "<g font-size='36' font-family='Verdana' "
+  & "fill='#602020' stroke='black' "
+  & "stroke-width='0.67'>"
+  & "<text x='40' y='60'>Drawing</text>"
+  & "</g>"
+  & "</g>"

svg = svg
+ & "</svg>"
```

Once the SVG string has been completed, an AcDrawingSVGPlane class is created. It is added to the drawing plane, and the SVG text is set into it. Super::Finish( ) is called to render the image in the following snippet:

```
Dim svgPlane As AcDrawingSVGPlane
Set svgPlane = AddDrawingPlane( DrawingPlaneTypeSVG )
svgPlane.SetSVG( svg )

Super::Finish( )
```

Debugging an SVG program

DebugSVG.rol is included in the DebuggingSVG.rod, located at <eRDPro_HOME>\Examples\DesignAndLayout\Drawings\DebuggingSVG.

DebugSVG.rol contains a control named DebugSVG. This control is a subclass of AcTextControl which has been customized to extract the SVG text from a drawing plane and display it. In addition, the control can send the SVG text to e.Report Designer Professional's Output window or dump it out to a text file. The behavior of the control can be selected based on certain properties, as shown in Table 9-4.

Table 9-4 DebugSVG properties

Property	Use
DrawingClassName	Set this property to the name of the drawing control that you want to debug.
DrawingPlaneIndex	Set this property to the index of the drawing plane that you want to debug.
DumpFileName	Set this property to the name of the file to which you want to dump the SVG text. If this property is blank, the SVG text is not dumped to a file. A sequence number and the .svg extension are added automatically to the file name you specify.
DumpFileFixSize	Set this property to true to add absolute size information to the SVG text which is dumped. If this property is false, the SVG in the dump file scales automatically when you view it using an external SVG viewer.
ShowInOutputPane	Set this property to true to display the SVG text in e.Report Designer Professional's output window.

Examining the DebugSVG control

To view the DebugSVG control, open <eRDPro_HOME>\Examples \DesignAndLayout\Drawings\DebuggingSVG\DebuggingSVG.rod. The Finish( ) method for the DebugSVG text control is overridden as follows:

```
Sub Finish( )
   Static s As Integer

   ' Find the drawing
   Dim drawing As AcDrawing
   Set drawing = GetContainer( ).FindContentByClass
+                  DrawingClassName )

   ' Get the SVG drawing plane
   Dim svgPlane As AcDrawingSVGPlane
   Set svgPlane = drawing.GetDrawingPlane( DrawingPlaneIndex )
```

```
   ' Extract the SVG
   Dim svg As String
   svg = svgPlane.GetSVG( )

   ' Display the SVG in this control
   DataValue = svg

   ' Dump the SVG to a file
   If (DumpFileName > "") Then
      Dim f As Integer
      f = FreeFile( )
      s = s + 1
      Open DumpFileName & Format( s, "_000" ) & ".svg"
+          For Output As f

      If DumpFileFixSize Then
         ' Adjust the SVG to specify absolute size so that
         ' the SVG debug file will display without scaling
         Dim svgElementEnd As Integer
         svgElementEnd = InStr( svg, ">" )
         svg = Left( svg, svgElementEnd - 1 )
+                & " width='" & SVGDbl( svgPlane.Size.Width /
+                          OnePoint ) & "pt'"
+                & " height='" & SVGDbl( svgPlane.Size.Height /
+                          OnePoint ) & "pt'"
+                & Mid( svg, svgElementEnd )
      End If
      Print #f, svg
      Close f
   End If

   ' Display the SVG in eRD Pro's Output pane
   If ShowInOutputPane And (GetAppContext( ) = DWBContext)
   Then
      ShowFactoryStatus( svg )
   End If

   Super::Finish( )
End Sub
```

The function retrieves the SVG code from the control, and displays it in one of several ways, based on property settings of the control. The first section of the following code retrieves the drawing control as specified by the DrawingClassName property:

```
   Dim drawing As AcDrawing
   Set drawing = GetContainer( ).FindContentByClass(
+                  DrawingClassName )
```

The next section retrieves the SVG drawing plane based on the DrawingPlaneIndex property.

```
Dim svgPlane As AcDrawingSVGPlane
Set svgPlane = drawing.GetDrawingPlane( DrawingPlaneIndex )
```

Once the SVG drawing plane is retrieved, GetSVG is called.

```
Dim svg As String
svg = svgPlane.GetSVG( )
```

After the SVG code is retrieved, the property settings of the DebugSVG control determine where to place the output.

Using the DebugSVG control

DebuggingSVG.rod consists of one frame containing a drawing control and a DebugSVG control, a subclass of the DebugSVG library component. It appears in Figure 9-2. The SVG generating code written in Actuate Basic is located within the Finish( ) method of the drawing control.

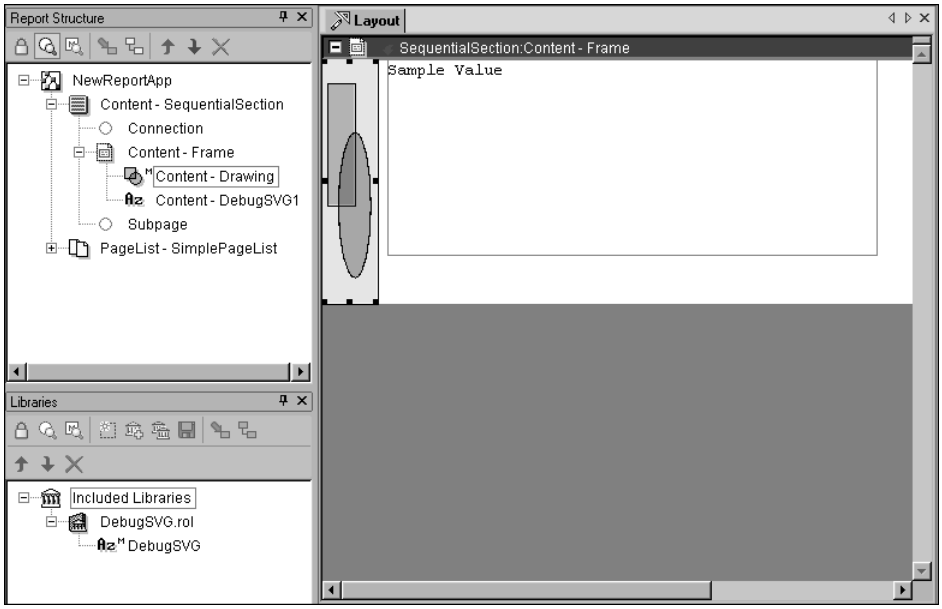


Figure 9-2 DebugSVG report design

How to examine the drawing control's SVG generating code

1 Select the drawing control in Layout.

2 Go to Properties—Methods. Open Sub Finish( ). Drawing::Finish appears, as shown below:

```
Sub Finish( )
   ' Get the size of the drawing in points
   Dim w As Double
   w = Size.Width / OnePoint
   Dim h As Double
   h = Size.Height / OnePoint

   Dim svg As String
   svg = "<svg version='1.1'"
+     ' Standard SVG 1.1 namespaces
+     & " xmlns='http://www.w3.org/2000/svg'"
+     & " xmlns:xlink='http://www.w3.org/1999/xlink'"
+     ' Do not collapse whitespace in text
+     & " xml:space='preserve'"
+     ' Scale the SVG to use points as the default units
+     & " viewBox='0 0 " & SVGDbl( w ) & " " & SVGDbl( h )
+     & "'>"

   ' Rotate and shift the coordinate system
   svg = svg
+     & "<g"
+     & " transform='"
+     & " translate(" & SVGDbl( w - 20 ) & "," &
+     & SVGDbl(h - 6 ) & ")"
+     & " rotate(90)"
+     & "'>"

   ' Place some text
   svg = svg
+     & "<text font-family='Arial' font-size='20' "
+     & "text-anchor='end'>Hello World!</text>"

   svg = svg
+     & "</g>"
+     & "</svg>"

   Dim svgPlane As AcDrawingSVGPlane
   Set svgPlane = AddDrawingPlane( DrawingPlaneTypeSVG )
   svgPlane.SetSVG( svg )

   Super::Finish( )
End Sub
```

The SVG generated by this code creates the phrase Hello World! rotated 90 degrees. When you run the report, it appears as in Figure 9-3.

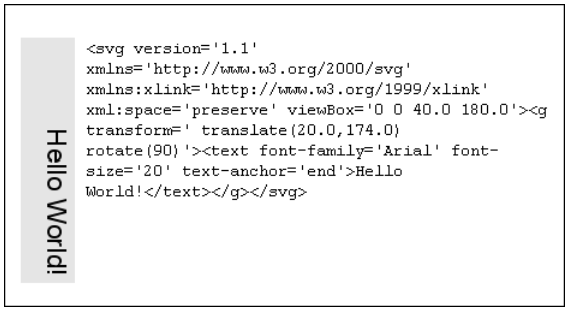


Figure 9-3 Report showing SVG code

The SVG code in the Finish( ) method generated the image. The SVG code that generated the image is displayed to the right of the image in the DebugSVG control.

At this point, you have the ability to observe the code and the resulting image created from it. Depending on the values you have set for the DebugSVG properties, you may also export the SVG code to other software tools to do further analysis.

Chapter

10

Presenting data in a cross tab

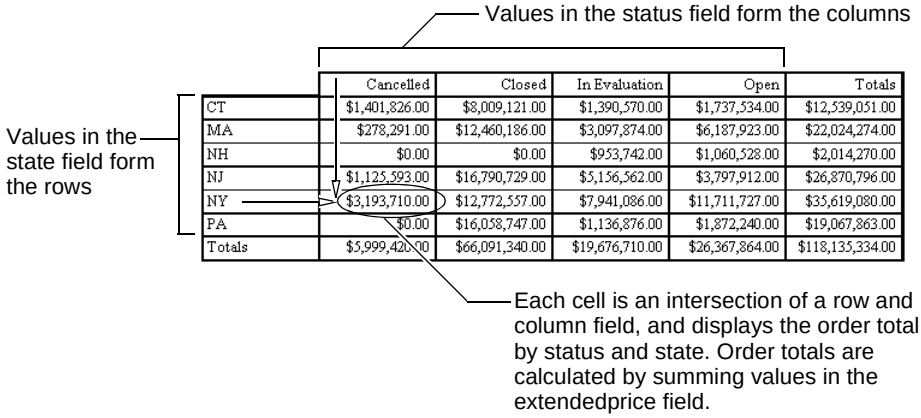
This chapter contains the following topics:

- About cross tabs
- Placing a cross tab in a report
- Building a cross tab
- Customizing the appearance of a cross tab
- Understanding cross tab sizing
- Printing a cross-tab report

About cross tabs

A cross tab displays data in a row-and-column matrix that looks similar to a spreadsheet. Like a spreadsheet, the cross tab is ideal for summarizing data in a compact and concise format. It displays summary, or aggregate, values such as sums, counts, or averages. The cross tab groups these values by one set of data listed down the left side of the matrix and another set of data listed across the top of the matrix.

Figure 10-1 shows a cross tab that displays order totals by status and by state. It uses data from three fields, status, state, and extendedprice.



	Cancelled	Closed	In Evaluation	Open	Totals
CT	$1,401,826.00	$8,009,121.00	$1,390,570.00	$1,737,534.00	$12,539,051.00
MA	$278,291.00	$12,460,186.00	$3,097,874.00	$6,187,923.00	$22,024,274.00
NH	$0.00	$0.00	$953,742.00	$1,060,528.00	$2,014,270.00
NJ	$1,125,593.00	$16,790,729.00	$5,156,562.00	$3,797,912.00	$26,870,796.00
NY	$3,193,710.00	$12,772,557.00	$7,941,086.00	$11,711,727.00	$35,619,080.00
PA	$0.00	$16,058,747.00	$1,136,876.00	$1,872,240.00	$19,067,863.00
Totals	$5,999,420.00	$66,091,340.00	$19,676,710.00	$26,367,864.00	$118,135,334.00

Figure 10-1 Cross tab displaying totals by status and state

As Figure 10-1 shows, a cross tab has the following characteristics:

- It requires at least three fields from a data source. These fields can be from one or multiple tables. Like most reports, cross tabs typically use data from multiple tables.
- One field populates the column headings in the cross tab. There is one column for each unique value in the field. In this example, there are four values in the status field, Cancelled, Close, In Evaluation, and Open. Each value is a column.
- One field populates the row headings in the cross tab. There is one row for each unique value in the field. In this example, there are six values in the state field, CT, MA, NH, NJ, NY, and PA. Each value is a row.
- One field's values are aggregated and these values populate the cells of the cross tab. In this example, each cell displays the order total by status and by state. The order total is calculated from the following aggregate expression:

```
Sum([items.extendedprice])
```

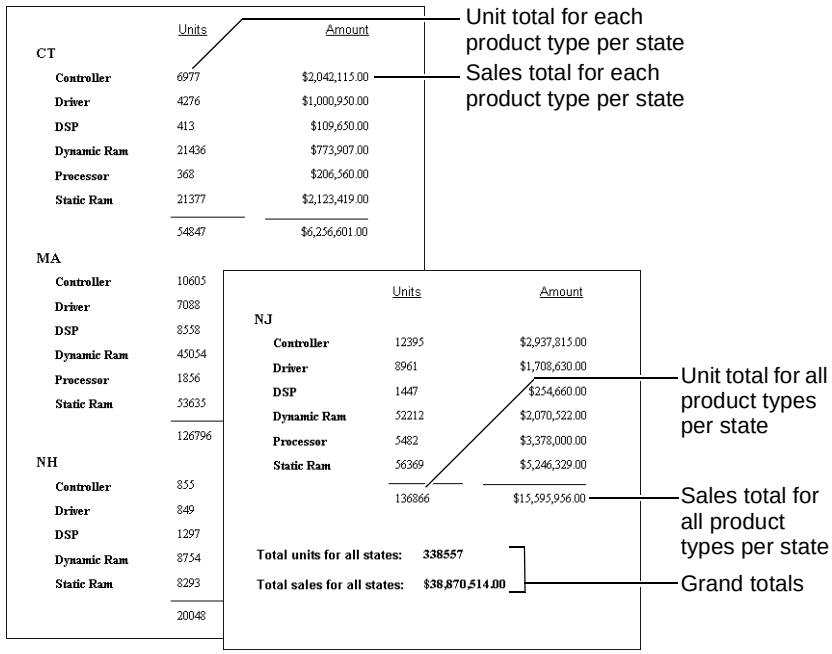
- The cells in the Totals column and Totals row display grand totals. In this example, the cells in the Totals column displays order totals for all statuses for each state. The cells in the Totals row displays order totals for all states for each status.
- Cross tabs are for numeric analyses only. While you can use aggregate functions, such as Max( ) and Min( ), with non-numeric data, the resulting text data is meaningless.

This chapter discusses how to create, modify, and print a cross-tab report. For information about exporting a cross-tab report to Excel, see *Working with Actuate e.Reports.*

Comparing a cross tab to a listing report

A cross tab is not the only way to present summary information. You can also present summary information in a listing report. Examine the reports in Figure 10-2 and Figure 10-3. Both reports display two aggregate values, unit totals and sales totals, grouped by product type and state.

Report 1: Listing report shows total units and total sales grouped by state and product type



	Units	Amount
CT		
Controller	6977	$2,042,115.00
Driver	4276	$1,000,950.00
DSP	413	$109,650.00
Dynamic Ram	21436	$773,907.00
Processor	368	$206,560.00
Static Ram	21377	$2,123,419.00
	54847	$6,256,601.00
MA		
Controller	10605	
Driver	7088	
DSP	8558	
Dynamic Ram	45054	
Processor	1856	
Static Ram	53635	
	126796	
NH		
Controller	855	
Driver	849	
DSP	1297	
Dynamic Ram	8754	
Static Ram	8293	
	20048	
NJ		
Controller	12395	$2,937,815.00
Driver	8961	$1,708,630.00
DSP	1447	$254,660.00
Dynamic Ram	52212	$2,070,522.00
Processor	5482	$3,378,000.00
Static Ram	56369	$5,246,329.00
	136866	$15,595,956.00

Figure 10-2 Listing report with total units and total sales grouped by state and product type

The listing report shows the same information as the cross tab, except that, without programming, it cannot calculate and display the overall totals for each product type. The listing report cannot show these totals easily because product type is the inner group and the listing report calculates overall totals only for the top-level group.

With the listing report, it is difficult to compare, for example, total sales for CT and NJ because the information is on different pages. It is also difficult to compare state totals with the grand totals because the grand totals appear at the end of the report.

The cross tab, on the other hand, has a compact format that is ideal for comparing and analyzing summary information. Whenever a report user requests summary information to be shown by one criterion and also by another, the cross tab is the perfect solution.

Report 2: Cross tab shows total units and total sales grouped by state and product type

The listing report cannot show these totals easily

	CT		MA		NH		NJ		Totals	
	Units	Amount	Units	Amount	Units	Amount	Units	Amount	Units	Amount
Controller	6,977.00	$2,042,115.00	10,605.00	$2,763,291.00	855.00	$188,955.00	12,395.00	$2,937,815.00	30,832.00	7,932,176.00
DSP	413.00	$109,650.00	8,558.00	$2,510,560.00	1,297.00	$440,980.00	1,447.00	$254,660.00	11,715.00	3,315,850.00
Driver	4,276.00	$1,000,950.00	7,088.00	$1,677,410.00	849.00	$271,680.00	8,961.00	$1,708,630.00	21,174.00	4,658,670.00
Dynamic Ram	21,436.00	$773,907.00	45,054.00	$1,693,978.00	8,754.00	$426,169.00	52,212.00	$2,070,522.00	127,456.00	4,964,576.00
Processor	368.00	$206,560.00	1,856.00	$1,066,320.00	0.00	$0.00	5,482.00	$3,378,000.00	7,706.00	4,650,880.00
Static Ram	21,377.00	$2,123,419.00	53,635.00	$5,153,836.00	8,293.00	$824,778.00	56,369.00	$5,246,329.00	139,674.00	13,348,362.00
Totals	54,847.00	6,256,601.00	126,796.00	14,865,395.00	20,048.00	2,152,562.00	136,866.00	15,595,956.00	338,557.00	38,870,514.00

Figure 10-3 Cross tab with total units and total sales grouped by state and product type

Understanding cross tab limitations

In Actuate e.Report Designer Professional, cross-tab controls have the following limitations:

- If you modify a cross-tab control, you cannot choose Undo to reverse the change. Actuate e.Report Designer Professional does not support Undo and Redo for a cross-tab control.
- When viewing a generated cross-tab report, you cannot select an individual cross-tab cell or search the contents of the cross tab.
- A cross-tab control uses a set of Actuate Foundation Classes. If you subclass a cross-tab control, the subclass inherits only the properties and methods of the AcCrosstab class. The cross-tab subclass copies all other information at the time of its creation, including the visual properties of its parent cross tab, such as background colors and fonts of heading labels.
- Actuate e.Report Designer Professional does not support modifying the behavior of a cross-tab control by overriding methods.

Placing a cross tab in a report

A cross tab looks like a standalone report. It is, however, just a component in a report, like a text object, date object, or chart. You can build a report that displays only a cross tab or you can build a report that presents data in a cross tab, as well as in other layouts. For example, a report design can contain a listing report and a cross tab, where the listing report displays detailed data and the cross tab displays summary data.

To display a cross tab, you place a cross-tab control in the report design the same way that you place other data controls. Use one of the following ways:

- Choose Insert➛Crosstab and place the cross-tab control in a frame.
- Drag a cross-tab control from Toolbox—Controls and drop it in a frame.

The frame and section in which you place a cross tab is very important. You see different behavior and data depending on where you place the cross tab. The following sections discuss these issues.

Understanding the effects of placing a cross tab in different frames

You almost always place a cross-tab control in an After frame. e.Report Designer Professional does not prevent you from placing it in a Before or Content frame, but the following are practical reasons for placing a cross-tab control in an After frame:

- When you run the report, the cross tab expands dynamically to accommodate all the rows and columns it needs to display. Remember, each unique value in the fields becomes a row or a column, so it is difficult to predict the exact size of the generated cross tab. The After frame expands dynamically to contain its contents, whereas the Before frame's size is fixed at design time. If you place a cross-tab control in the Before frame, and the cross tab expands to a size that is larger than the frame, the cross tab is truncated.
- The cross tab displays aggregate information, and controls that display aggregate information are placed either in a Before or After frame. If you place a cross-tab control in a Content frame, the report cannot properly display other data in the same frame.

Understanding the effects of placing a cross tab in different sections

The section in which you place a cross tab determines where in the report it appears, as well as, the data it displays. Consider this example. You create a cross tab that shows total orders by city and status. If you place this cross tab in a report

section, the cross tab displays data for every city and every status. If, however, you place this same cross tab in a group section that is based on the city field, the cross tab displays data for every status, but for only one city. This result occurs because the group section has already grouped the report data by city.

Another difference is that a cross tab placed in a report section appears only once in the report. A cross tab placed in a group section appears once in each group and, therefore, multiple times in the report.

Figure 10-4 and Figure 10-5 show a report design and the output when the same cross tab is placed in a report section and in a group section. The cross tab and group section data are grouped by the same field.

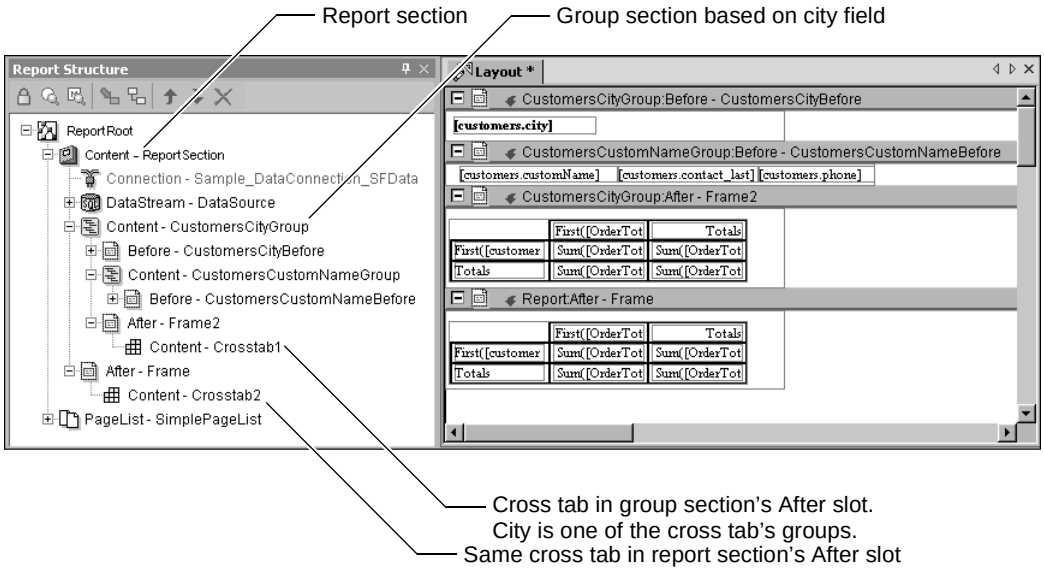


Figure 10-4 Report design with cross tab placed in report and group sections

As you can see, the cross tab in the group section displays only two rows, and one of them is the Totals row, which displays the same data as the row above. This use of a cross tab is not a particularly good one. There are better ways to display the same data.

You can place a cross tab in a group section. You should consider carefully the fields to use in the cross tab, however. Do not use the same field that the grouped section uses as its grouping field.

For example, instead of grouping the cross-tab data in the previous example by city and by status, you can group the data by customer and status, as shown in Figure 10-6.

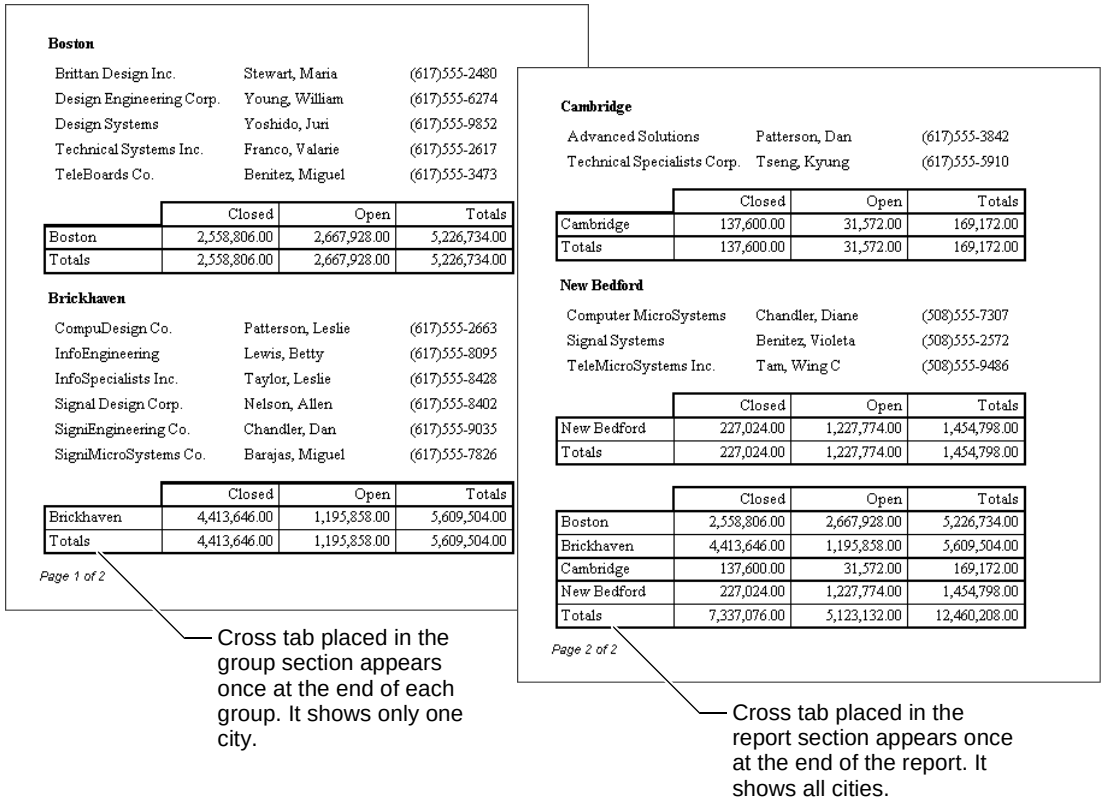


Figure 10-5 Report output showing cross tab placed in report and group sections

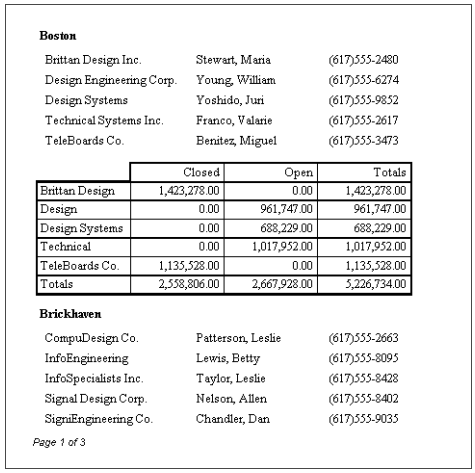


Figure 10-6 Grouping cross-tab data by customer and status

Placing multiple cross tabs in a frame

If you wish, you can place multiple cross-tab controls in a frame to display a series of cross tabs. Bear in mind, however, that in this case, all the cross tabs use the same query, so make sure the query specifies sufficient data for presenting different types of information.

Figure 10-7 and Figure 10-8 show a report design that includes two cross-tab controls in an After frame, and the output.

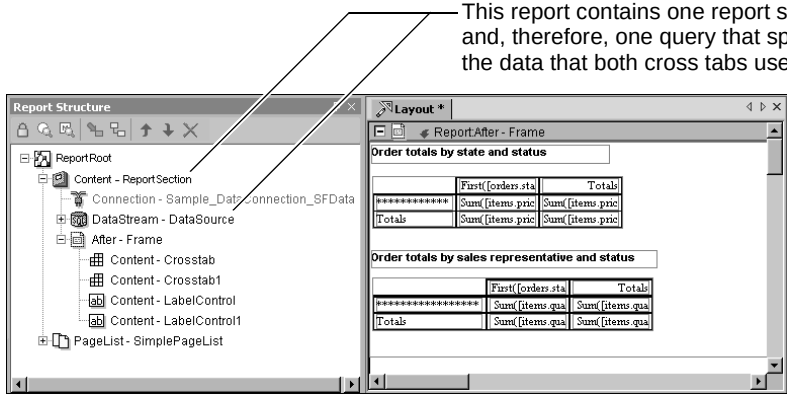


Figure 10-7 Report design that includes multiple cross tabs

Order totals by state and status

	Cancelled	Closed	In Evaluation	Open	Selected	Totals
CT	$119,526.00	$4,287,421.00	$962,220.00	$887,434.00	$0.00	$6,256,601.00
MA	$278,291.00	$6,943,650.00	$3,097,874.00	$4,545,580.00	$0.00	$14,865,395.00
NH	$0.00	$0.00	$1,092,034.00	$1,060,528.00	$0.00	$2,152,562.00
NJ	$619,993.00	$9,408,029.00	$2,234,522.00	$3,333,412.00	$0.00	$15,595,956.00
NY	$2,011,710.00	$4,149,207.00	$7,253,836.00	$3,673,327.00	$1,010,287.00	$18,098,367.00
PA	$0.00	$12,901,913.00	$91,776.00	$1,680,590.00	$364,917.00	$15,039,196.00
Totals	$3,029,520.00	$37,690,220.00	$14,732,262.00	$15,180,871.00	$1,375,204.00	$72,008,077.00

Order totals by sales representative and status

	Cancelled	Closed	In Evaluation	Open	Selected	Totals
Allen Thompson	0.00	2,875,117.00	745,840.00	571,019.00	0.00	4,191,976.00
Betty Howard	0.00	4,261,055.00	0.00	1,417,640.00	0.00	5,678,695.00
Diane Murphy	278,291.00	4,688,009.00	1,248,420.00	2,206,145.00	1,010,287.00	9,431,152.00
Foon Yue Tseng	2,011,710.00	699,236.00	3,143,459.00	0.00	0.00	5,854,405.00
George Vanauf	0.00	0.00	297,356.00	362,386.00	0.00	659,742.00
Jeff Firrelli	0.00	364,624.00	788,926.00	1,128,593.00	0.00	2,282,143.00
John Hernandez	619,993.00	8,130,110.00	309,734.00	1,116,435.00	0.00	10,176,272.00
Julie Firrelli	0.00	5,131,218.00	91,776.00	262,950.00	364,917.00	5,850,861.00
Leslie Jennings	0.00	1,277,919.00	1,053,133.00	1,956,703.00	0.00	4,287,755.00
Leslie Thompson	0.00	770,054.00	0.00	560,228.00	0.00	1,330,282.00
Maria Barajas	119,526.00	1,777,221.00	975,051.00	1,696,934.00	0.00	4,568,732.00
Mary Patterson	0.00	1,684,017.00	4,146,384.00	1,420,435.00	0.00	7,250,836.00
Pamela Castillo	0.00	2,558,806.00	1,060,528.00	2,221,129.00	0.00	5,840,463.00
Steve Patterson	0.00	3,472,834.00	0.00	0.00	0.00	3,472,834.00
William Patterson	0.00	0.00	871,655.00	260,274.00	0.00	1,131,929.00
Totals	3,029,520.00	37,690,220.00	14,732,262.00	15,180,871.00	1,375,204.00	72,008,077.00

Figure 10-8 Report output showing multiple cross tabs

Building a cross tab

The following are the basic steps for building a cross tab:

- Build the query.
- Insert a cross-tab control into an After frame.
- Specify the data to display in the cross tab:
  - Specify the field whose values form the cross tab columns.
  - Specify the field whose values form the cross tab rows.
  - Specify an aggregate expression to calculate the values that appear in the cross tab cells.
- Optionally, customize the appearance of the cross tab and the formats of data.

When you insert a cross-tab control, e.Report Designer Professional launches Crosstab Builder, a tool that helps you provide the necessary field information for a cross tab. After you build a cross tab, you can use the Crosstab Builder at any time to make changes to the fields.

This section provides step-by-step instructions for building a cross tab that displays total sales by state and product category. The report uses data from the customers, orders, and items tables in the Sfdata sample database. Figure 10-9 shows the cross tab in the finished report.

	CT	MA	NH	NJ	NY	PA	Totals
Controller	2,042,115.00	2,763,291.00	188,955.00	2,937,815.00	2,783,108.00	2,113,483.00	12,828,767.00
DSP	109,650.00	2,510,560.00	440,980.00	254,660.00	988,040.00	1,686,400.00	5,990,290.00
Driver	1,000,950.00	1,677,410.00	271,680.00	1,708,630.00	2,137,110.00	1,688,100.00	8,483,880.00
Dynamic Ram	773,907.00	1,693,978.00	426,169.00	2,070,522.00	1,512,057.00	2,195,404.00	8,672,037.00
Processor	206,560.00	1,066,320.00	0.00	3,378,000.00	4,631,680.00	2,893,160.00	12,175,720.00
Static Ram	2,123,419.00	5,153,836.00	824,778.00	5,246,329.00	6,046,372.00	4,462,649.00	23,857,383.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	15,595,956.00	18,098,367.00	15,039,196.00	72,008,077.00

Figure 10-9 Cross tab displaying total sales by state and product category

How to create a new report and build the query

1. Choose File➛New.
2. In Create New Report, select Blank Report. Choose OK. A basic report design appears. It contains a default connection to the Sfdata database, which this report uses.
3. Build the query:
   1. Choose View➛Data.
   2. In Database login, choose OK. You do not need a user name or password.
   3. In Database Browser, drag the following tables to Query Editor:

- customers
- orders
- items

The tables and their fields appear in Query Editor.

4 Add the following fields to the query by dragging them from the tables and dropping them into Column Name on the Columns page:

- customers.state
- items.category
- items.pricequote
- items.quantity

Figure 10-10 shows all your selections in Query Editor.

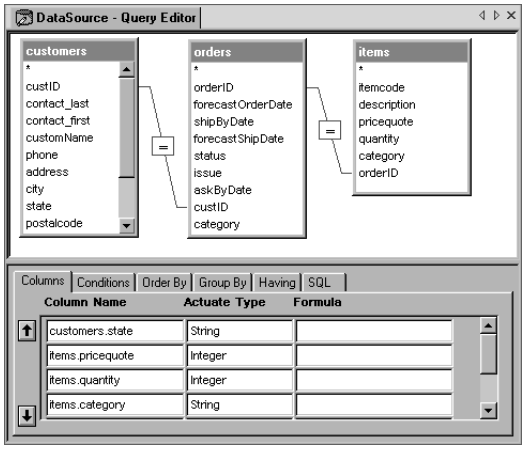


Figure 10-10 Building a query

4 Choose View➛Design to return to the design perspective.

5 Save the report.

How to insert a cross-tab control

1 In Report Structure, expand Content—ReportSection. By default, a frame is placed in Content.

2 Move this frame from Content to After. A cross-tab control should be placed in an After frame.

3 Choose Insert➛Crosstab. Drop the control into the After frame. Crosstab Builder appears, as shown in Figure 10-11.

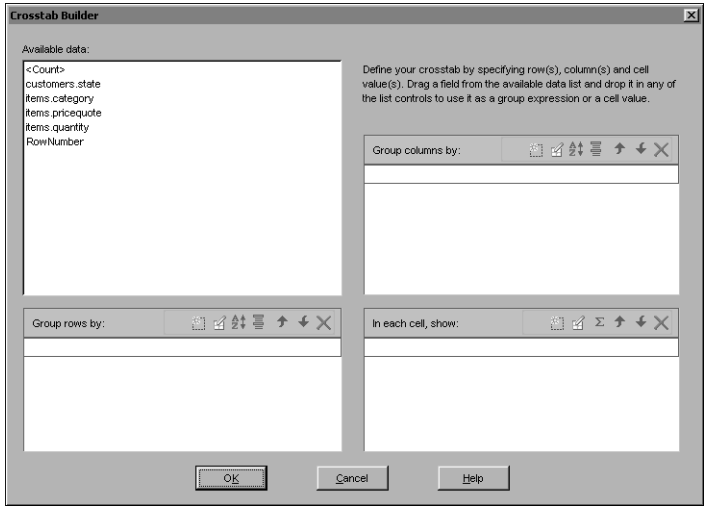


Figure 10-11 Crosstab Builder

Crosstab Builder displays in Available Data the fields that you selected for the query. Discounting the Available Data portion, the layout of Crosstab Builder itself mimics the layout of data in a cross tab, as shown in Figure 10-12. This makes it easy to visualize the data layout that you want as you work in Crosstab Builder.

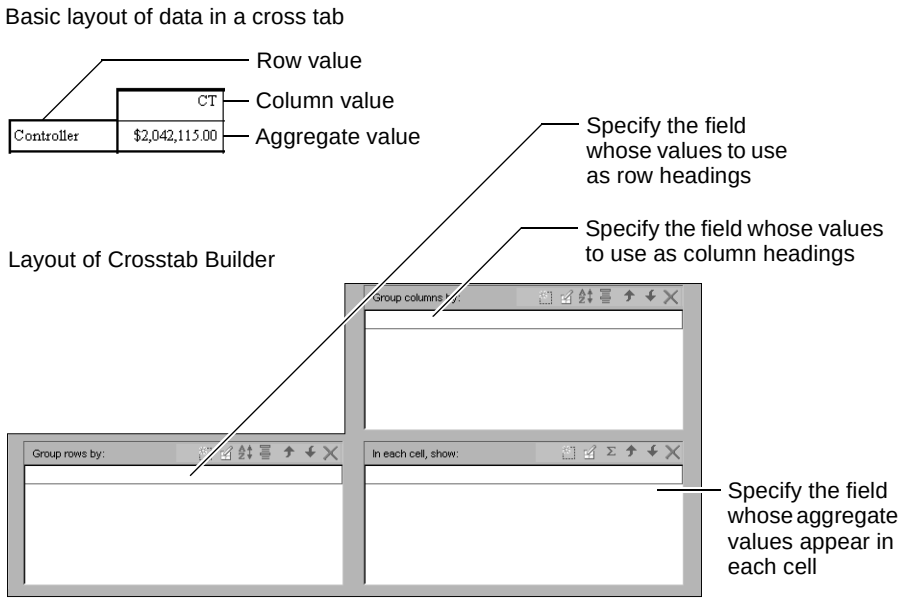


Figure 10-12 Crosstab Builder layout

How to specify the data to display in the cross tab

1 To specify the field whose values populate the cross tab column headings, drag the customers.state field from Available Data and drop it into Group columns by, as shown in Figure 10-13.

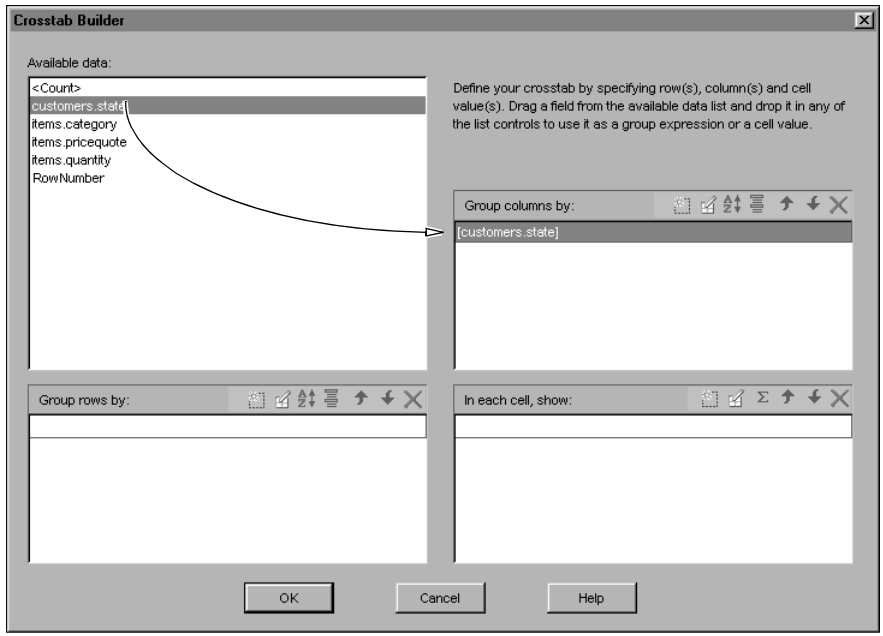


Figure 10-13 Specifying cross tab column data

2 To specify the field whose values populate the cross tab row headings, drag the items.category field from Available Data and drop it into Group rows by.

3 To specify the aggregate expression to calculate values for the cross tab cells, type the following in the first line in In each cell, show:

```
Sum([items.quantity] * [items.pricequote])
```

[items.quantity] * [items.pricequote] calculates the total amount for an order. The Sum aggregate function sums the total amounts of all orders.

4 Check the values that you specified. Crosstab Builder should appear as in Figure 10-14.

5 Choose OK. The cross-tab control appears in the After frame in the layout window, as shown in Figure 10-15.

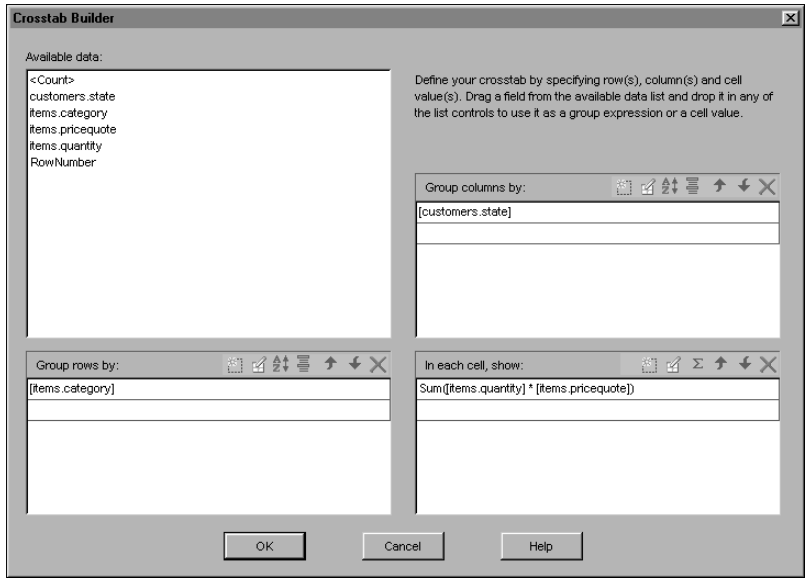


Figure 10-14 Specifying cross tab row and cell data

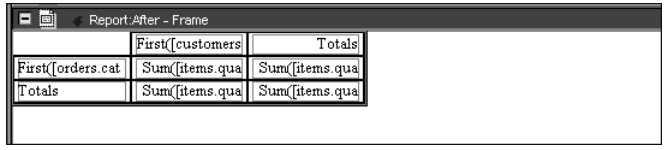


Figure 10-15 Cross-tab control in the layout window

6 Choose Report➛Build and Run. e.Report Designer Professional cycles through the data several times to calculate the totals for all row and column combinations. Report generation can take more time than usual.

The report displays the cross tab, as shown in Figure 10-16.

	CT	MA	NH	NJ	NY	PA	Totals
Controller	2,042,115.00	2,763,291.00	188,955.00	2,937,815.00	2,783,108.00	2,113,483.00	12,828,767.00
DSP	109,650.00	2,510,560.00	440,980.00	254,660.00	988,040.00	1,686,400.00	5,990,290.00
Driver	1,000,950.00	1,677,410.00	271,680.00	1,708,630.00	2,137,110.00	1,688,100.00	8,483,880.00
Dynamic Ram	773,907.00	1,693,978.00	426,169.00	2,070,522.00	1,512,057.00	2,195,404.00	8,672,037.00
Processor	206,560.00	1,066,320.00	0.00	3,378,000.00	4,631,680.00	2,893,160.00	12,175,720.00
Static Ram	2,123,419.00	5,153,836.00	824,778.00	5,246,329.00	6,046,372.00	4,462,649.00	23,857,383.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	15,595,956.00	18,098,367.00	15,039,196.00	72,008,077.00

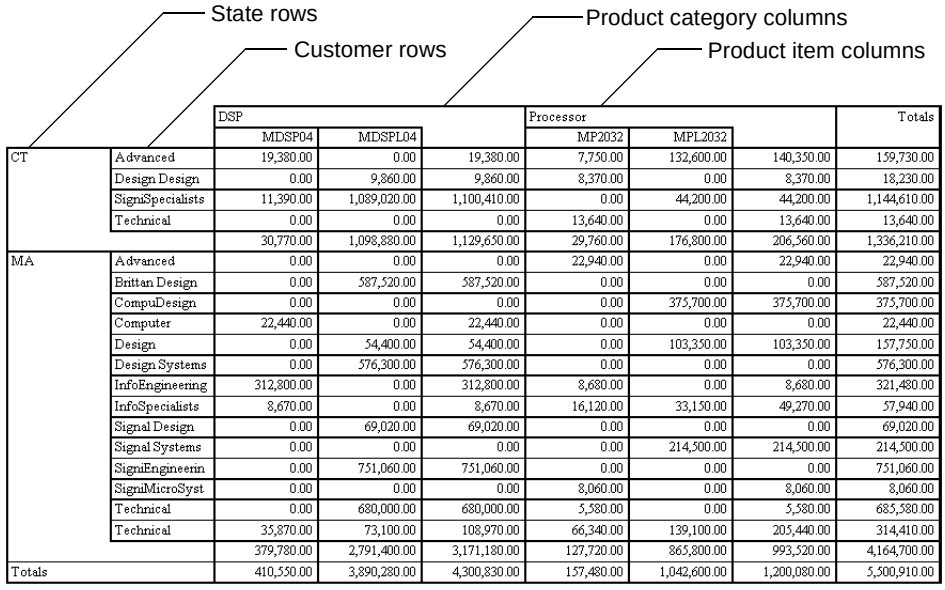
Figure 10-16 Cross tab output

You just learned how to build a basic cross tab that displays data using three fields, the minimum number. The rest of this chapter describes how to build cross tabs that use more fields and display more data, how to specify custom headings, how to sort and group data by intervals, and how to improve the appearance of a cross tab.

Grouping data by more than two fields

When you build a cross tab, you group the aggregated data by at least two fields, for example, total order amounts by state and status, or total units sold by product type and customer. If you wish, you can show aggregate information as it relates to more criteria or fields. For example, you can show total order amounts by three fields, state, customer, and product category, or by four fields, state, customer, product category, and product item.

Figure 10-17 shows a cross tab that shows the total order amount by state, customer, product category, and product item.



		DSP			Processor			Totals
		MDSP04	MDSPL04		MP2032	MPL2032		
CT	Advanced	19,380.00	0.00	19,380.00	7,750.00	132,600.00	140,350.00	159,730.00
	Design Design	0.00	9,860.00	9,860.00	8,370.00	0.00	8,370.00	18,230.00
	SigniSpecialists	11,390.00	1,089,020.00	1,100,410.00	0.00	44,200.00	44,200.00	1,144,610.00
	Technical	0.00	0.00	0.00	13,640.00	0.00	13,640.00	13,640.00
		30,770.00	1,098,880.00	1,129,650.00	29,760.00	176,800.00	206,560.00	1,336,210.00
MA	Advanced	0.00	0.00	0.00	22,940.00	0.00	22,940.00	22,940.00
	Brittan Design	0.00	587,520.00	587,520.00	0.00	0.00	0.00	587,520.00
	CompuDesign	0.00	0.00	0.00	0.00	375,700.00	375,700.00	375,700.00
	Computer	22,440.00	0.00	22,440.00	0.00	0.00	0.00	22,440.00
	Design	0.00	54,400.00	54,400.00	0.00	103,350.00	103,350.00	157,750.00
	Design Systems	0.00	576,300.00	576,300.00	0.00	0.00	0.00	576,300.00
	InfoEngineering	312,800.00	0.00	312,800.00	8,680.00	0.00	8,680.00	321,480.00
	InfoSpecialists	8,670.00	0.00	8,670.00	16,120.00	33,150.00	49,270.00	57,940.00
	Signal Design	0.00	69,020.00	69,020.00	0.00	0.00	0.00	69,020.00
	Signal Systems	0.00	0.00	0.00	0.00	214,500.00	214,500.00	214,500.00
	SigniEngineerin	0.00	751,060.00	751,060.00	0.00	0.00	0.00	751,060.00
	SigniMicroSyst	0.00	0.00	0.00	8,060.00	0.00	8,060.00	8,060.00
	Technical	0.00	680,000.00	680,000.00	5,580.00	0.00	5,580.00	685,580.00
	Technical	35,870.00	73,100.00	108,970.00	66,340.00	139,100.00	205,440.00	314,410.00
		379,780.00	2,791,400.00	3,171,180.00	127,720.00	865,800.00	993,520.00	4,164,700.00
Totals		410,550.00	3,890,280.00	4,300,830.00	157,480.00	1,042,600.00	1,200,080.00	5,500,910.00

Figure 10-17 Cross tab showing data grouped by multiple fields

As Figure 10-17 shows, each additional field that you group data by appears as a subcolumn or subrow. The customer rows appear within the state rows. The product item columns appear within the product category columns. This type of grouping is similar to creating nested groups in a listing report.

Each additional field that you group by provides a more comprehensive and detailed view of the data. Just as there is no limit to the number of groups that you can create in a listing report, there is no limit to the number of grouping levels that you can create in a cross tab. Practically, however, a cross tab that contains more than two or three grouping levels is difficult to read and probably does not make much sense.

To create multiple groups, you specify multiple fields in Group columns by and Group rows by in Crosstab Builder. Figure 10-18 shows the fields defined in Crosstab Builder to create the cross tab shown in Figure 10-17.

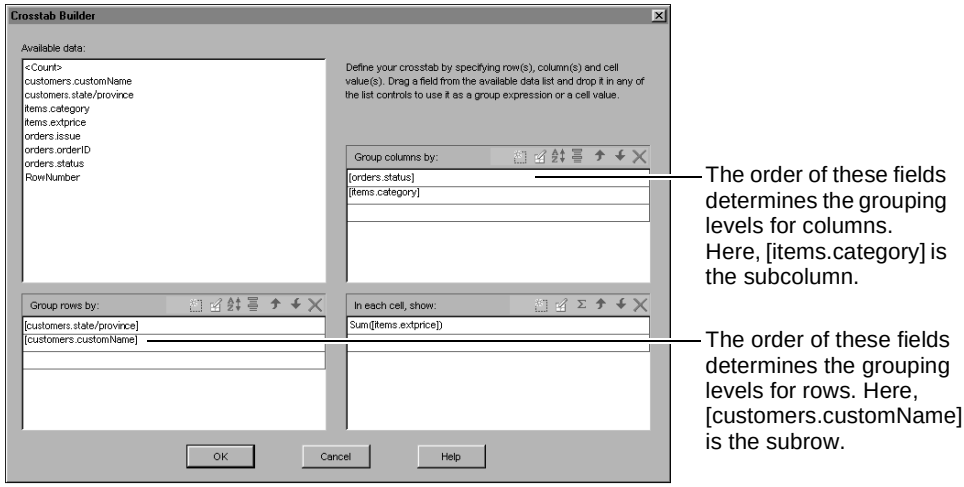


Figure 10-18 Specifying multiple fields to create multiple groups

If you plan to create multiple row or column fields, choose fields that have a logical one-to-many relationship. Figure 10-19 shows some of the following examples of fields to use as row and subrow, or column and subcolumn:

- City and Customers
- State and City
- Product category and Product names
- Country and Region
- Customer and Orders

Rows grouped by city and customers

Albany	Advanced
	CompuEngineeri
	Signal
	Technical
Allentown	Advanced
	CompuBoards
	InfoBoards
	Technical
	TekniMicroSyst

Rows grouped by state and city

CT	Bridgewater
	Glendale
	New Haven
MA	Boston
	Brickhaven
	Cambridge
	New Bedford

Columns grouped by product category and names

Controller		
MP1608	MP1608s	MP1608x

Figure 10-19 Commonly grouped fields

Displaying multiple aggregate fields

You can specify that cross tab cells display multiple aggregate values. For example, you can display in each cell the sum of order amounts and the average order amounts, or total units sold and total order amounts.

Figure 10-20 shows a cross tab that displays total units sold and total order amounts.

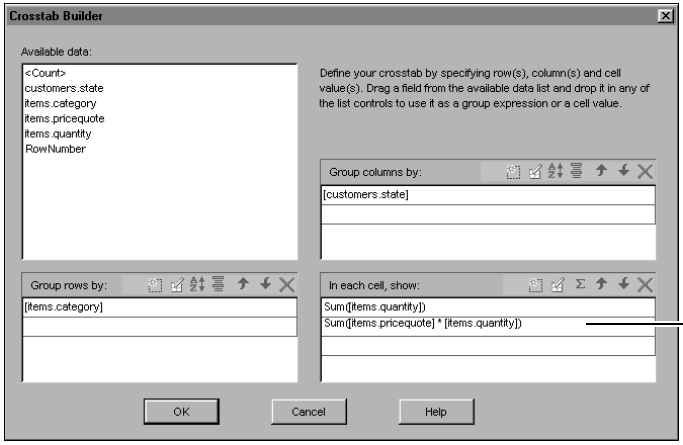
Each cell displays two aggregate values, side by side

	CT		MA		NH		NJ		Totals	
	Units	Amount	Units	Amount	Units	Amount	Units	Amount	Units	Amount
Controller	6,977.00	$2,042,115.00	10,605.00	$2,763,291.00	855.00	$188,955.00	12,395.00	$2,937,815.00	30,832.00	7,932,176.00
DSP	413.00	$109,650.00	8,558.00	$2,510,560.00	1,297.00	$440,980.00	1,447.00	$254,660.00	11,715.00	3,315,850.00
Driver	4,276.00	$1,000,950.00	7,088.00	$1,677,410.00	849.00	$271,680.00	8,961.00	$1,708,630.00	21,174.00	4,658,670.00
Dynamic Ram	21,436.00	$773,907.00	45,054.00	$1,693,978.00	8,754.00	$426,169.00	52,212.00	$2,070,522.00	127,456.00	4,964,576.00
Processor	368.00	$206,560.00	1,856.00	$1,066,320.00	0.00	$0.00	5,482.00	$3,378,000.00	7,706.00	4,650,880.00
Static Ram	21,377.00	$2,123,419.00	53,635.00	$5,153,836.00	8,293.00	$824,778.00	56,369.00	$5,246,329.00	139,674.00	13,348,362.00
Totals	54,847.00	6,256,601.00	126,796.00	14,865,395.00	20,048.00	2,152,562.00	136,866.00	15,595,956.00	338,557.00	38,870,514.00

Figure 10-20 Cross tab displaying multiple aggregate fields

To display multiple values in each cell, you specify multiple fields in the In each cell, show: part of Crosstab Builder.

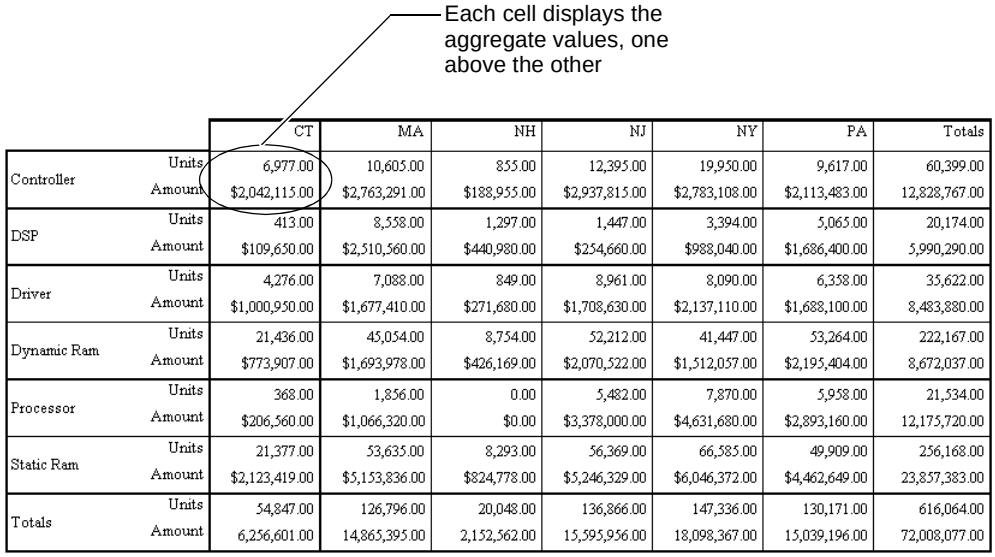
Figure 10-21 shows the fields defined in Crosstab Builder to create the cross tab shown in Figure 10-20.



The order of these aggregate fields determines the order in which the values appear in the cells

Figure 10-21 Specifying multiple aggregate fields

The default setting is for a cross tab to display values horizontally in a cell. You can specify that the cross tab display them vertically instead, as shown in Figure 10-22.



		CT	MA	NH	NJ	NY	PA	Totals
Controller	Units	6,977.00	10,605.00	855.00	12,395.00	19,950.00	9,617.00	60,399.00
	Amount	$2,042,115.00	$2,763,291.00	$188,955.00	$2,937,815.00	$2,783,108.00	$2,113,483.00	12,828,767.00
DSP	Units	413.00	8,558.00	1,297.00	1,447.00	3,394.00	5,065.00	20,174.00
	Amount	$109,650.00	$2,510,560.00	$440,980.00	$254,660.00	$988,040.00	$1,686,400.00	5,990,290.00
Driver	Units	4,276.00	7,088.00	849.00	8,961.00	8,090.00	6,358.00	35,622.00
	Amount	$1,000,950.00	$1,677,410.00	$271,680.00	$1,708,630.00	$2,137,110.00	$1,688,100.00	8,483,880.00
Dynamic Ram	Units	21,436.00	45,054.00	8,754.00	52,212.00	41,447.00	53,264.00	222,167.00
	Amount	$773,907.00	$1,693,978.00	$426,169.00	$2,070,522.00	$1,512,057.00	$2,195,404.00	8,672,037.00
Processor	Units	368.00	1,856.00	0.00	5,482.00	7,870.00	5,958.00	21,534.00
	Amount	$206,560.00	$1,066,320.00	$0.00	$3,378,000.00	$4,631,680.00	$2,893,160.00	12,175,720.00
Static Ram	Units	21,377.00	53,635.00	8,293.00	56,369.00	66,585.00	49,909.00	256,168.00
	Amount	$2,123,419.00	$5,153,836.00	$824,778.00	$5,246,329.00	$6,046,372.00	$4,462,649.00	23,857,383.00
Totals	Units	54,847.00	126,796.00	20,048.00	136,866.00	147,336.00	130,171.00	616,064.00
	Amount	6,256,601.00	14,865,395.00	2,152,562.00	15,595,956.00	18,098,367.00	15,039,196.00	72,008,077.00

Figure 10-22 Displaying cross tab cell values vertically

If you compare Figure 10-22 with Figure 10-20, you see that displaying cell values vertically reduces the width of the cross tab and increases its height. So, besides your preference for reading values horizontally or vertically, you can use this layout option to adjust the height and width of a cross tab.

How to display cell values vertically

1. In Report Structure, right-click the cross-tab control, then choose Properties.
2. In the Properties page for the cross tab, set Crosstab Layout➤ValuePlacement to ValuesVertical, as shown in Figure 10-23.

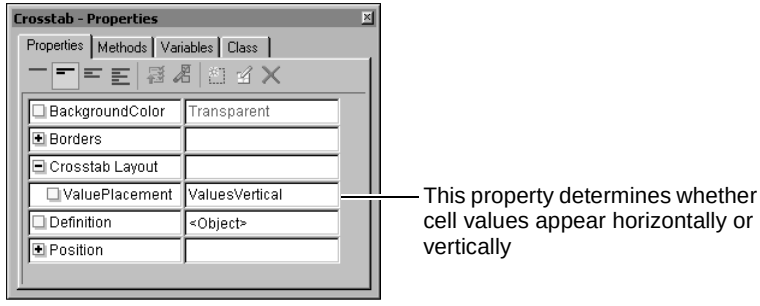


Figure 10-23 Specifying vertical placement of cell values

Displaying computed values in row and column headings

So far, all the examples show row and column headings displaying values exactly as they appear in the data source fields. If you wish, you can use a computed field to display values that are different from those in the data source.

For example, you want to display the full names of sales representatives in the row headings. The data source, however, stores the last names and first names in two separate fields. To display the full names, you specify the following expression in Group Row By in Crosstab Builder to concatenate the values from the two fields:

```
[salesreps.first] & " " & [salesreps.last]
```

You can also specify an expression to format data. For example, you can use the following expression to display the name of a month, such as January, instead of a date, such as 01/19/04:

```
Format$([orders.shipByDate], "mmmm")
```

You can use Expression Builder to construct and verify expressions.

Displaying static text in row and column headings

You can add static headings to a cross tab. A static heading is user-defined and always appears in the cross tab. You typically use static headings to accomplish the following:

- Display data with different text. For example, rather than display abbreviated state names as they are stored in the data source, you can specify that NJ appear as New Jersey.
- Add missing text data, such as the name of a state or order status. For example, in a cross tab that shows order totals by state, you can include a state that does not contain any order totals, as shown in Figure 10-24.

Static heading

	Cancelled	Closed	Open	Totals
CA - Coming Soon	$0.00	$0.00	$0.00	$0.00
CT	$1,401,826.00	$8,009,121.00	$1,737,534.00	$11,148,481.00
NJ	$1,125,593.00	$16,790,729.00	$3,797,912.00	$21,714,234.00
NY	$3,193,710.00	$12,772,557.00	$11,711,727.00	$27,677,994.00
PA	$0.00	$16,058,747.00	$1,872,240.00	$17,930,987.00
WA	$0.00	$393,900.00	$0.00	$393,900.00
Totals	$5,721,129.00	$54,025,054.00	$19,119,413.00	$78,865,596.00

Figure 10-24 Using static text

How to add a static value

1 In Crosstab Builder, select the row or column field to which you want to add a static value, as shown in Figure 10-25.

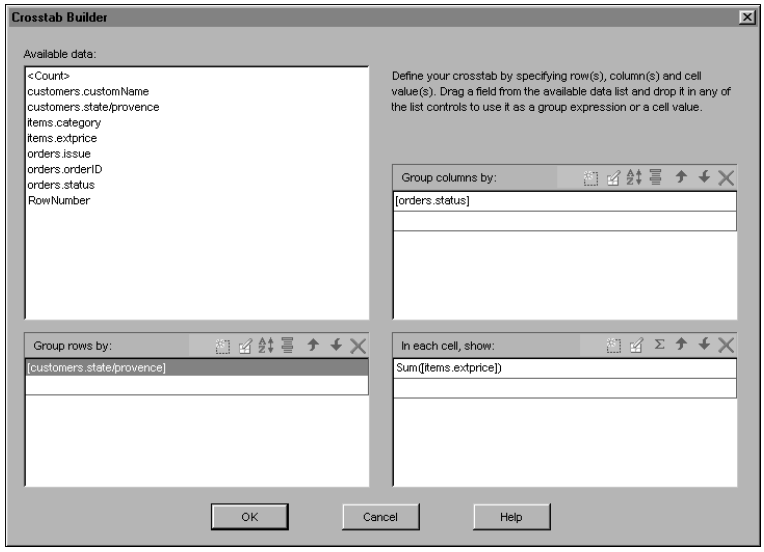


Figure 10-25 Adding a static value to a row or column field

2 Choose Grouping.

3 In Sorting and Grouping—Grouping, choose Static Groups. Sorting and Grouping—Static Group appears, as shown in Figure 10-26.

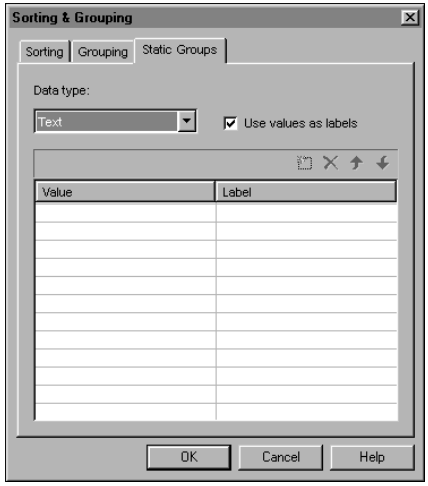


Figure 10-26 Specifying static groups

4 Specify the following:

- In Data type, select the data type of the static value or values to add.
- In Value, type a value that uses the data type that you selected in Data type.
  - To add a value, type the text that you want to display in the cross tab.
  - To replace a data source value, type the value that you want to replace.
- In Label, type the text to display in the cross tab. If you leave this option blank, the cross tab displays the text that you typed in Value.

Choose OK.

Figure 10-27 shows how to add a new value, California--Coming Soon, to the cross tab. It also shows how to display data source values NJ and NH as New Jersey and New Hampshire.

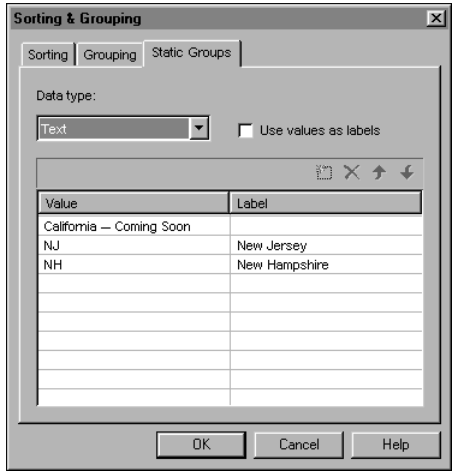


Figure 10-27 Adding a new static value to the cross tab

Figure 10-28 shows how the static values appear in the cross tab.

	Cancelled	Closed	In Evaluation	Open	Totals
California -- Coming Soon	$0.00	$0.00	$0.00	$0.00	$0.00
New Hampshire	$0.00	$0.00	$953,742.00	$1,060,528.00	$2,014,270.00
New Jersey	$1,125,593.00	$16,790,729.00	$5,156,562.00	$3,797,912.00	$26,870,796.00
Totals	$1,125,593.00	$16,790,729.00	$6,110,304.00	$4,858,440.00	$28,885,066.00

Figure 10-28 Static value output

Limiting the amount of data the cross tab displays

When you generate a report that contains a cross tab, e.Report Designer Professional creates one column or one row in the cross tab for each unique value of the data source fields, as shown in Figure 10-29.

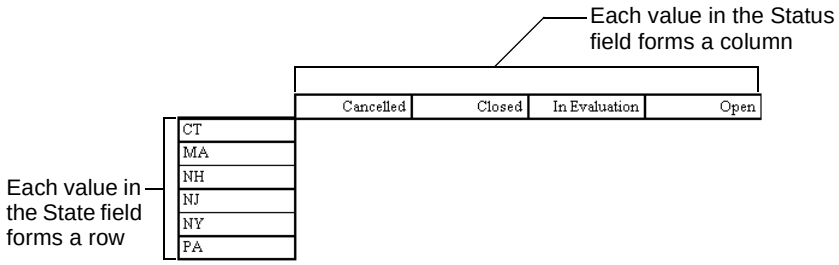


Figure 10-29 Limiting data

If a field used to populate the cross tab columns contains tens of values, the cross tab is very wide. Similarly, if a field used to populate the cross tab rows contains many values, the cross tab is very tall. For example, if you want to use the values in a customers field to populate the cross tab rows, and there are 1000 customer names, the generated cross tab contains 1000 rows.

While there is no limit to the amount of data a cross tab can display, you improve the usability of the data if you limit the amount. After all, the main advantage of presenting data in a cross tab is the ability to compare and analyze information, preferably on a single page or screen.

As you do with any type of report, you narrow the scope of data to retrieve through the query. If you select a customer field, you can use the Conditions page in Query Editor to retrieve only customers in certain regions, or with certain credit rankings, or with orders exceeding a specified amount. For more information about filtering data, see *Accessing Data using e.Report Designer Professional*.

You can also create parameters to prompt report users to specify what data they want to view.

Sorting cross tab data

By default, data displayed in the row and column headings is sorted in ascending order. Alternatively, you can:

- Sort data in descending order.
- Sort data in its database order, and specify whether it is sorted from the first record to the last, or the reverse.

How to sort cross tab data

1 In Crosstab Builder, select the field to sort. You can sort only fields in Group columns by and Group rows by.

2 Choose Sorting. Sorting and Grouping—Sorting appears, as shown in Figure 10-30.

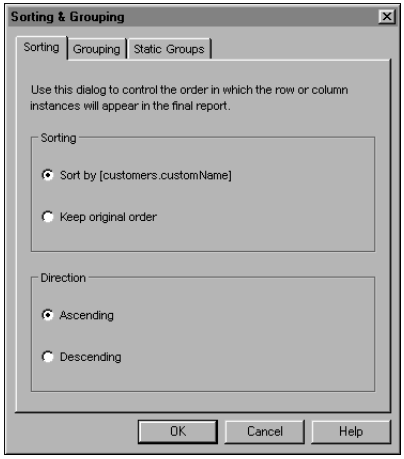


Figure 10-30 Sorting and Grouping window

3 Select the sorting options:

- To display data in descending order, in Direction, select Descending.
- To display data in the order in which it appears in the data source, in Sorting, select Keep original order.

  The options in Direction change from Ascending and Descending to Forward and Reverse.
- To display data in the reverse of the order in which it appears in the data source, select Reverse.

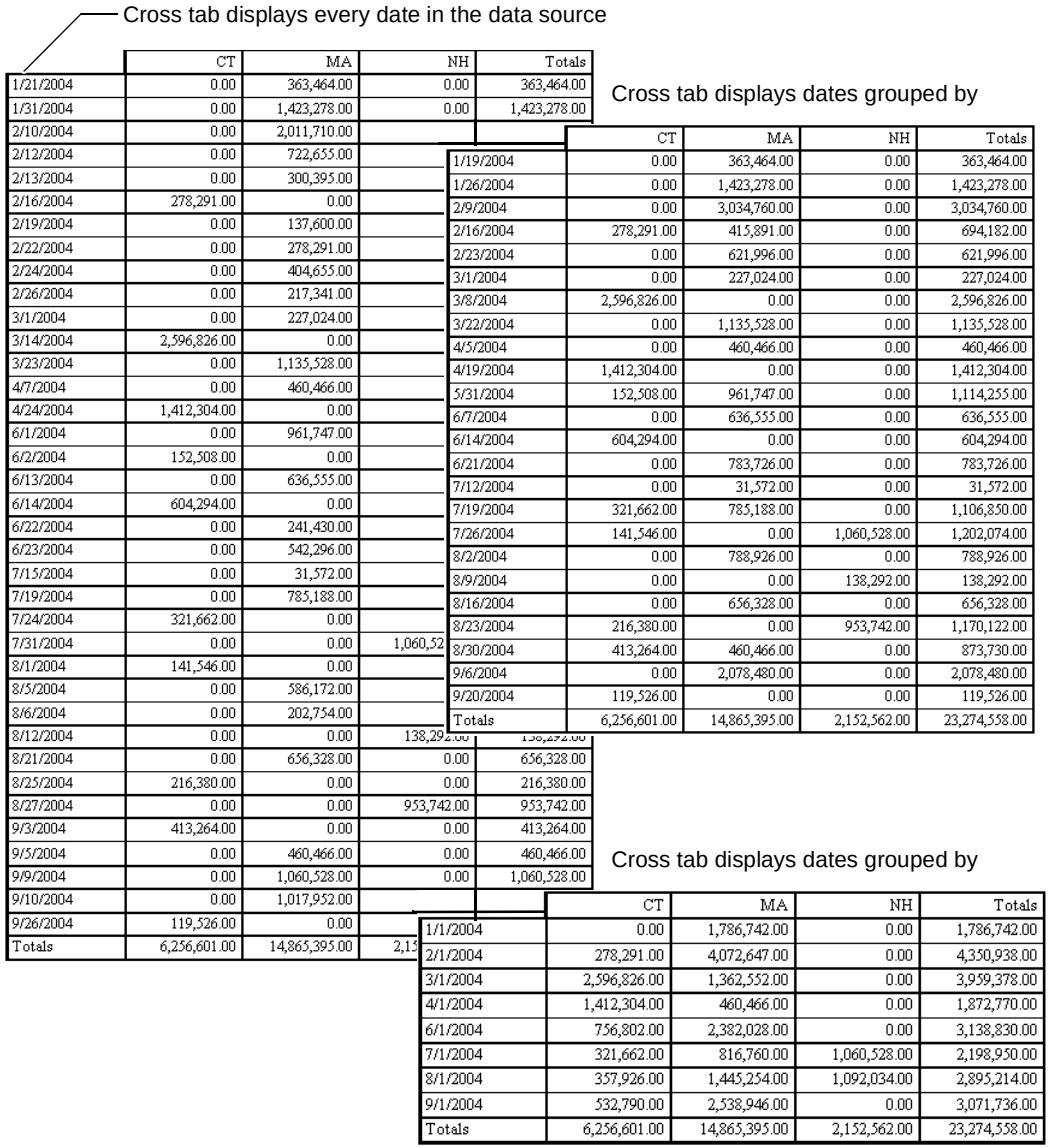
Choose OK.

Grouping cross tab data by intervals

When you specify the fields whose values form the row or column headings, e.Report Designer Professional creates a row or column for each value. Sometimes, however, it is more useful to display values grouped by interval instead of by individual values. For example, you can group date values by weeks or months instead of by the dates stored in the data source. This functionality is similar to the functionality for creating groups in a listing report and a chart. Unlike the grouping functionality in a listing report, a cross tab can group only numeric data and date-and-time data by intervals. A cross tab cannot group text data by intervals.

Figure 10-31 shows three cross tabs. The first cross tab displays every shipByDate field value for CT, MA, and NH in the row headings. The second cross tab

displays the values grouped by weeks. The third cross tab displays the values grouped by months.



	CT	MA	NH	Totals
1/21/2004	0.00	363,464.00	0.00	363,464.00
1/31/2004	0.00	1,423,278.00	0.00	1,423,278.00
2/10/2004	0.00	2,011,710.00		
2/12/2004	0.00	722,655.00		
2/13/2004	0.00	300,395.00		
2/16/2004	278,291.00	0.00		
2/19/2004	0.00	137,600.00		
2/22/2004	0.00	278,291.00		
2/24/2004	0.00	404,655.00		
2/26/2004	0.00	217,341.00		
3/1/2004	0.00	227,024.00		
3/14/2004	2,596,826.00	0.00		
3/23/2004	0.00	1,135,528.00		
4/7/2004	0.00	460,466.00		
4/24/2004	1,412,304.00	0.00		
6/1/2004	0.00	961,747.00		
6/2/2004	152,508.00	0.00		
6/13/2004	0.00	636,555.00		
6/14/2004	604,294.00	0.00		
6/22/2004	0.00	241,430.00		
6/23/2004	0.00	542,296.00		
7/15/2004	0.00	31,572.00		
7/19/2004	0.00	785,188.00		
7/24/2004	321,662.00	0.00		
7/31/2004	0.00	0.00	1,060,52	
8/1/2004	141,546.00	0.00		
8/5/2004	0.00	586,172.00		
8/6/2004	0.00	202,754.00		
8/12/2004	0.00	0.00	138,29	
8/21/2004	0.00	656,328.00	0.00	656,328.00
8/25/2004	216,380.00	0.00	0.00	216,380.00
8/27/2004	0.00	0.00	953,742.00	953,742.00
9/3/2004	413,264.00	0.00	0.00	413,264.00
9/5/2004	0.00	460,466.00	0.00	460,466.00
9/9/2004	0.00	1,060,528.00	0.00	1,060,528.00
9/10/2004	0.00	1,017,952.00		
9/26/2004	119,526.00	0.00		
Totals	6,256,601.00	14,865,395.00	2,15	

	CT	MA	NH	Totals
1/19/2004	0.00	363,464.00	0.00	363,464.00
1/26/2004	0.00	1,423,278.00	0.00	1,423,278.00
2/9/2004	0.00	3,034,760.00	0.00	3,034,760.00
2/16/2004	278,291.00	415,891.00	0.00	694,182.00
2/23/2004	0.00	621,996.00	0.00	621,996.00
3/1/2004	0.00	227,024.00	0.00	227,024.00
3/8/2004	2,596,826.00	0.00	0.00	2,596,826.00
3/22/2004	0.00	1,135,528.00	0.00	1,135,528.00
4/5/2004	0.00	460,466.00	0.00	460,466.00
4/19/2004	1,412,304.00	0.00	0.00	1,412,304.00
5/31/2004	152,508.00	961,747.00	0.00	1,114,255.00
6/7/2004	0.00	636,555.00	0.00	636,555.00
6/14/2004	604,294.00	0.00	0.00	604,294.00
6/21/2004	0.00	783,726.00	0.00	783,726.00
7/12/2004	0.00	31,572.00	0.00	31,572.00
7/19/2004	321,662.00	785,188.00	0.00	1,106,850.00
7/26/2004	141,546.00	0.00	1,060,528.00	1,202,074.00
8/2/2004	0.00	788,926.00	0.00	788,926.00
8/9/2004	0.00	0.00	138,292.00	138,292.00
8/16/2004	0.00	656,328.00	0.00	656,328.00
8/23/2004	216,380.00	0.00	953,742.00	1,170,122.00
8/30/2004	413,264.00	460,466.00	0.00	873,730.00
9/6/2004	0.00	2,078,480.00	0.00	2,078,480.00
9/20/2004	119,526.00	0.00	0.00	119,526.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	23,274,558.00

	CT	MA	NH	Totals
1/1/2004	0.00	1,786,742.00	0.00	1,786,742.00
2/1/2004	278,291.00	4,072,647.00	0.00	4,350,938.00
3/1/2004	2,596,826.00	1,362,552.00	0.00	3,959,378.00
4/1/2004	1,412,304.00	460,466.00	0.00	1,872,770.00
6/1/2004	756,802.00	2,382,028.00	0.00	3,138,830.00
7/1/2004	321,662.00	816,760.00	1,060,528.00	2,198,950.00
8/1/2004	357,926.00	1,445,254.00	1,092,034.00	2,895,214.00
9/1/2004	532,790.00	2,538,946.00	0.00	3,071,736.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	23,274,558.00

Figure 10-31 Cross tab output showing data grouped by intervals

As the examples show, grouping values by interval summarizes data further for more effective analysis.

How to group cross tab data by intervals

1 In Crosstab Builder, select the field to group by interval.

2 Choose Grouping. Sorting and Grouping—Grouping appears, as shown in Figure 10-32.

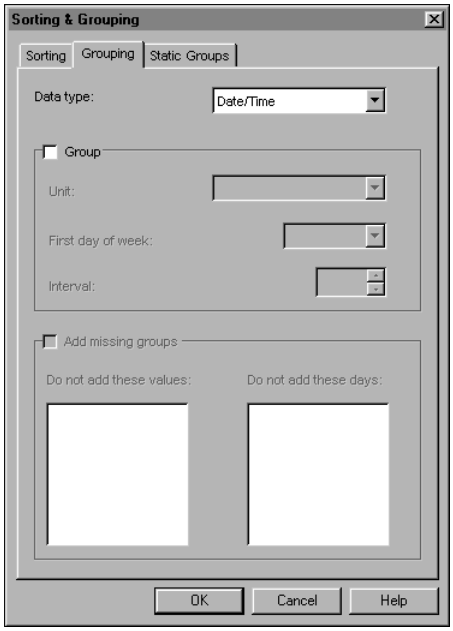


Figure 10-32 Grouping cross tab data by intervals

3 Specify the grouping options:

- In Data type, select the data type of the values to group.
- Select Group.
- In Unit, select the grouping unit. If you select Week, you can specify the day to use as the first day of the week.
- In Interval, specify the group interval.

For example, to display data in two-month periods, specify the Date and Time data type, the month unit, and an interval of 2.

4 To specify that Crosstab Builder display values for missing groups, select Add missing groups. Choose OK.

Adding missing groups

When you group data in row or column headings by interval, the cross tab displays only groups for which there is data. The cross tab in Figure 10-33 groups

dates by month. Notice that it skips 5/1/2004 because there is no data for that month.

	CT	MA	NH	Totals
1/1/2004	0.00	1,786,742.00	0.00	1,786,742.00
2/1/2004	278,291.00	4,072,647.00	0.00	4,350,938.00
3/1/2004	2,596,826.00	1,362,552.00	0.00	3,959,378.00
4/1/2004	1,412,304.00	460,466.00	0.00	1,872,770.00
6/1/2004	756,802.00	2,382,028.00	0.00	3,138,830.00
7/1/2004	321,662.00	816,760.00	1,060,528.00	2,198,950.00
8/1/2004	357,926.00	1,445,254.00	1,092,034.00	2,895,214.00
9/1/2004	532,790.00	2,538,946.00	0.00	3,071,736.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	23,274,558.00

5/1/2004 is missing

Figure 10-33 Cross tab with missing group

You can have Crosstab Builder add missing groups, as shown in Figure 10-34.

	CT	MA	NH	Totals
1/1/2004	0.00	1,786,742.00	0.00	1,786,742.00
2/1/2004	278,291.00	4,072,647.00	0.00	4,350,938.00
3/1/2004	2,596,826.00	1,362,552.00	0.00	3,959,378.00
4/1/2004	1,412,304.00	460,466.00	0.00	1,872,770.00
5/1/2004	0.00	0.00	0.00	0.00
6/1/2004	756,802.00	2,382,028.00	0.00	3,138,830.00
7/1/2004	321,662.00	816,760.00	1,060,528.00	2,198,950.00
8/1/2004	357,926.00	1,445,254.00	1,092,034.00	2,895,214.00
9/1/2004	532,790.00	2,538,946.00	0.00	3,071,736.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	23,274,558.00

5/1/2004 added

Figure 10-34 Cross tab with missing group added

How to add missing groups

1. In Crosstab Builder, select the field to group.
2. Choose Grouping.
3. In Sorting and Grouping—Grouping, if you have not already done so, specify the grouping unit and interval.
4. Select Add missing groups. This option is not available for text data. To add missing text data, such as an office or state for which there is no summary data, you can specify static group values.
5. If you wish, specify the conditions for adding missing groups.
   - To keep Crosstab Builder from adding certain missing values, specify the group to exclude in Do not add these values.
   - To exclude a day when you group data by day, select the day or days to exclude in Do not add these days.

   In Figure 10-35, date data is grouped by day. If there is no data for certain days, e.Report Designer Professional adds all the missing days, except for Saturday and Sunday.

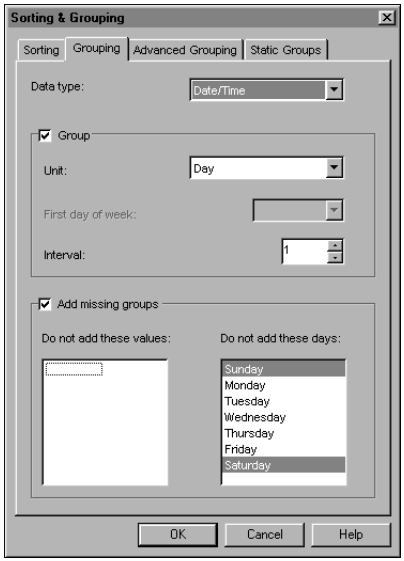


Figure 10-35 Adding missing groups

Figure 10-36 shows two cross tabs. The first cross tab does not display missing groups. The second cross tab shows the results of specifying the grouping options, as shown in Figure 10-35.

Cross tab displays groups for which there is data

	MA	PA	Totals
1/5/2004	0.00	518,937.00	518,937.00
1/10/2004	0.00	1,777,989.00	1,777,989.00
1/21/2004	363,464.00	0.00	363,464.00
1/22/2004	0.00	897,359.00	897,359.00
1/29/2004	0.00	742,515.00	742,515.00
1/31/2004	1,423,278.00	1,508,016.00	2,931,294.00
Totals	1,786,742.00	5,444,816.00	7,231,558.00

Cross tab displays missing groups

	MA	PA	Totals
1/5/2004	0.00	518,937.00	518,937.00
1/6/2004	0.00	0.00	0.00
1/7/2004	0.00	0.00	0.00
1/8/2004	0.00	0.00	0.00
1/9/2004	0.00	0.00	0.00
1/10/2004	0.00	1,777,989.00	1,777,989.00
1/12/2004	0.00	0.00	0.00
1/13/2004	0.00	0.00	0.00
1/14/2004	0.00	0.00	0.00
1/15/2004	0.00	0.00	0.00
1/16/2004	0.00	0.00	0.00
1/19/2004	0.00	0.00	0.00
1/20/2004	0.00	0.00	0.00
1/21/2004	363,464.00	0.00	363,464.00
1/22/2004	0.00	897,359.00	897,359.00
1/23/2004	0.00	0.00	0.00
1/26/2004	0.00	0.00	0.00
1/27/2004	0.00	0.00	0.00
1/28/2004	0.00	0.00	0.00
1/29/2004	0.00	742,515.00	742,515.00
1/30/2004	0.00	0.00	0.00
1/31/2004	1,423,278.00	1,508,016.00	2,931,294.00
Totals	1,786,742.00	5,444,816.00	7,231,558.00

Figure 10-36 Not displaying and displaying missing groups

Specifying the range of data to use when adding missing groups

When you add missing groups, you can also specify the range of data to group. If you do not specify the range of data, the cross tab uses the data range from the data source. In an earlier example that grouped data by month, the shipByDate field contained only dates from January to September, minus May, so 1/1/2004 and 9/1/2004 were the minimum and maximum values. If you want the cross tab to display all the months in the year, you need to specify that the range of data to group by is a year.

Figure 10-37 shows two cross tabs. The first cross tab adds missing groups, but the range of data is limited to the dates in the shipByDate field. The second cross tab adds missing groups and sets the range of data to one year.

	CT	MA	NH	Totals
1/1/2004	0.00	1,786,742.00	0.00	1,786,742.00
2/1/2004	278,291.00	4,072,647.00	0.00	4,350,938.00
3/1/2004	2,596,826.00	1,362,552.00	0.00	3,959,378.00
4/1/2004	1,412,304.00	460,466.00	0.00	1,872,770.00
5/1/2004	0.00	0.00	0.00	0.00
6/1/2004	756,802.00	2,382,028.00	0.00	3,138,830.00
7/1/2004	321,662.00	816,760.00	1,060,528.00	2,198,950.00
8/1/2004	357,926.00	1,445,254.00	1,092,034.00	2,895,214.00
9/1/2004	532,790.00	2,538,946.00	0.00	3,071,736.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	23,274,558.00

5/1/2004 added

	CT	MA	NH	Totals
1/1/2004	0.00	1,786,742.00	0.00	1,786,742.00
2/1/2004	278,291.00	4,072,647.00	0.00	4,350,938.00
3/1/2004	2,596,826.00	1,362,552.00	0.00	3,959,378.00
4/1/2004	1,412,304.00	460,466.00	0.00	1,872,770.00
5/1/2004	0.00	0.00	0.00	0.00
6/1/2004	756,802.00	2,382,028.00	0.00	3,138,830.00
7/1/2004	321,662.00	816,760.00	1,060,528.00	2,198,950.00
8/1/2004	357,926.00	1,445,254.00	1,092,034.00	2,895,214.00
9/1/2004	532,790.00	2,538,946.00	0.00	3,071,736.00
10/1/2004	0.00	0.00	0.00	0.00
11/1/2004	0.00	0.00	0.00	0.00
12/1/2004	0.00	0.00	0.00	0.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	23,274,558.00

5/1/2004 added

10/1/2004, 11/1/2004, and 12/1/2004 added

Figure 10-37 Specifying range of data when adding missing groups

To specify the range of data to use, you use round and span settings.

Round settings determine where to begin and end the data range using standard start and end points. For example, the standard start and end points for a year are January and December. The standard start and end points for integer units of ten are zero and nine, ten and nineteen, and so on.

Span settings determine a minimum length for the data range without using standard start and end points. When you set a group span, the range begins with

the first group for which there is data. The end point is either with the last group for which there is data, or the last group the span determines. For example, if you set a span of 1000 for a range of customer IDs that begin with 450 and end with 1600, the range ends at 1600. If the customer IDs begin with 450 and end with 1300, the range ends at 1450.

To specify round or span settings, you set the following options:

- The round or span unit determines the length of the range. For date data, the unit can be, for example, a day, a week, a month, a quarter, and so on.
- The round or span interval determines the number of units to use to round out or extend the range.

Depending on what data that you want to display, you can use either round or span, or both concurrently.

For example, the cross tab in Figure 10-38 shows the order amounts of NH and NJ by month. It uses a round unit of quarter and an interval of one. The group range begins with the first month, 1/1/2004, of the first quarter, Q1 2004, for which there is data and ends with the last month, 9/1/2004 of the last quarter, Q3 2004, for which there is data. The Sorting and Grouping window in Figure 10-38 shows the round settings specified on the Advanced Grouping page that resulted in the cross tab.

	NH	NJ	Totals
1/1/2004	0.00	0.00	0.00
2/1/2004	0.00	3,988,333.00	3,988,333.00
3/1/2004	0.00	4,635,358.00	4,635,358.00
4/1/2004	0.00	784,338.00	784,338.00
5/1/2004	0.00	1,824,785.00	1,824,785.00
6/1/2004	0.00	1,055,127.00	1,055,127.00
7/1/2004	1,060,528.00	991,181.00	2,051,709.00
8/1/2004	1,092,034.00	2,316,834.00	3,408,868.00
9/1/2004	0.00	0.00	0.00
Totals	2,152,562.00	15,595,956.00	17,748,518.00

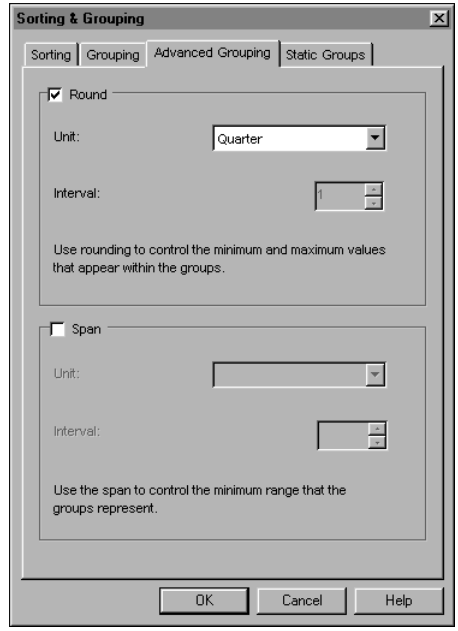


Figure 10-38 Specifying range of data using only round settings

The cross tab in Figure 10-39 uses the same round settings as in Figure 10-38 and adds a span unit of year and a span interval of one. Now, the group range ends with 12/1/2004, which is at least a year from the beginning of the range.

The Sorting and Grouping window shows the round and span settings specified on the Advanced Grouping page that resulted in the cross tab.

	NH	NJ	Totals
1/1/2004	0.00	0.00	0.00
2/1/2004	0.00	3,988,333.00	3,988,333.00
3/1/2004	0.00	4,635,358.00	4,635,358.00
4/1/2004	0.00	784,338.00	784,338.00
5/1/2004	0.00	1,824,785.00	1,824,785.00
6/1/2004	0.00	1,055,127.00	1,055,127.00
7/1/2004	1,060,528.00	991,181.00	2,051,709.00
8/1/2004	1,092,034.00	2,316,834.00	3,408,868.00
9/1/2004	0.00	0.00	0.00
10/1/2004	0.00	0.00	0.00
11/1/2004	0.00	0.00	0.00
12/1/2004	0.00	0.00	0.00
Totals	2,152,562.00	15,595,956.00	17,748,518.00

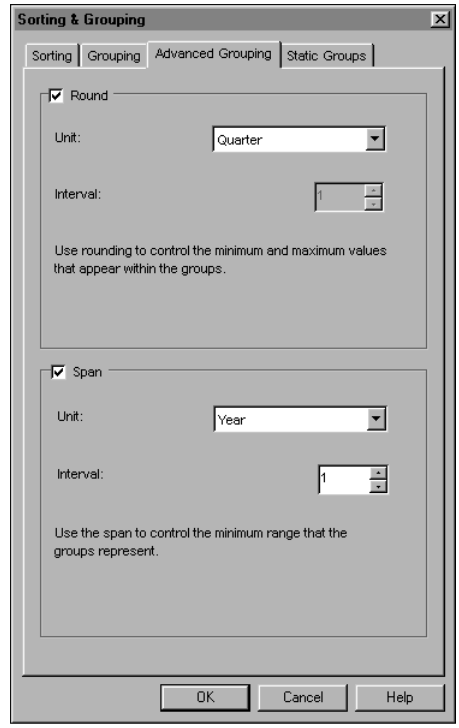


Figure 10-39 Specifying range of data using both round and span settings

How to specify the range of data to use when adding missing groups

1 In Crosstab Builder, select the field to group.

2 Choose Grouping.

3 In Sorting and Grouping—Grouping, if you have not already done so, specify the grouping unit and interval.

4 Select Add missing groups, and optionally, specify the conditions for adding missing groups.

5 Choose Advanced Grouping. Sorting and Grouping—Advanced Grouping appears, as shown in Figure 10-40.

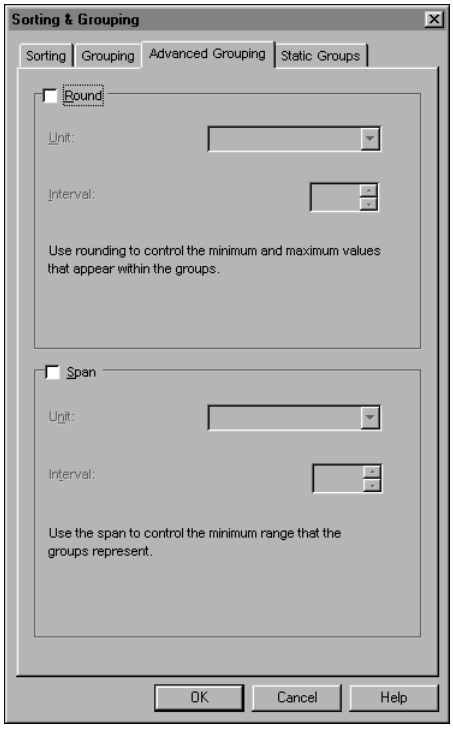


Figure 10-40 Specifying conditions for adding missing groups

6 To set the start and ending values for the group range:

- Select Round.
- To determine where to begin and end adding missing groups, select a rounding unit and a rounding interval. The unit and the interval must be the same or larger than the group unit that you selected on Sorting and Grouping—Grouping. For example, if you chose to group data by month, the rounding units that you can select are month, quarter, half-year, and year.

  If you group data by month and use a rounding unit of Quarter and a rounding unit of 1, the first group is the first month of the first quarter for which there is data. The last group is the last month of the last quarter for which there is data.

7 To set the minimum span of the group range:

- Select Span.
- Select a span unit and a span interval. The span unit must be the same as or larger than the rounding unit that you selected in step 6.

For example, if you group data by month and use a span unit of Year and a span interval of 1, the first group is the first month for which data appears. The last group is at least one year after the first group.

Choose OK.

Customizing the appearance of a cross tab

The default appearance of the generated cross tab is plain. For example, all data is displayed in the same font, size, and color. As with other controls, e.Report Designer Professional provides many options to customize the appearance of the cross tab. You can change the color and thickness of borders, resize columns and rows, and display data in different formats.

If you look at the cross-tab control in the layout window, you notice that it consists of multiple objects. As Figure 10-41 shows, the cross-tab control consists of eight objects. Each cell is an object.

	First([customers	Totals
First([items.cate	Sum([items.qua	Sum([items.qua
Totals	Sum([items.qua	Sum([items.qua

Figure 10-41 Cross-tab control

You can format each object individually. For example, you can set the column headings, row headings, data cells, and Totals cells to different font styles and colors from one another, as shown in Figure 10-42.

Cross tab design

	First([custome	Totals
First([items.cat	Sum([items.qua	Sum([items.qua
Totals	Sum([items.qua	Sum([items.q

Cross tab output

	CT	MA	NH	Totals
Controller	2,042,115.00	2,763,291.00	188,955.00	4,994,361.00
DSP	109,650.00	2,510,560.00	440,980.00	3,061,190.00
Driver	1,000,950.00	1,677,410.00	271,680.00	2,950,040.00
Dynamic Ram	773,907.00	1,693,978.00	426,169.00	2,894,054.00
Processor	206,560.00	1,066,320.00	0.00	1,272,880.00
Static Ram	2,123,419.00	5,153,836.00	824,778.00	8,102,033.00
Totals	6,256,601.00	14,865,395.00	2,152,562.00	23,274,558.00

Figure 10-42 Formatting cross tab objects

You can also apply formatting to entire rows or columns, or to the entire cross tab. Right-click an object in the cross-tab control to view the available formatting options.

The following sections describe the typical ways to customize the appearance of a cross tab.

Resizing cross tab rows and columns

If the generated cross tab displays truncated data or asterisks (*), the row or column is not large enough to display the data. If this occurs, you can resize the rows or columns of the cross-tab control in the layout window. You can use the mouse or specify a value for the row's Height property or the column's Width property.

How to resize a cross tab row

1. In the layout window, place the mouse pointer at the bottom edge of the cross tab row to resize. The pointer changes shape, as shown in Figure 10-43.

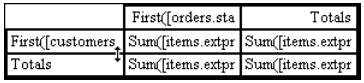


Figure 10-43 Resizing a cross tab row

2. Drag the bottom edge of the row to the desired position.
3. To set an exact size, do the following:
   1. Right-click any of the objects in the row that you want to resize.
   2. Choose Row properties from the context menu.
   3. Set the Height property to the desired value, then choose OK.

How to resize a cross tab column

1. In the layout window, place the mouse pointer at the right edge of the column to resize. The pointer changes shape, as shown in Figure 10-44.

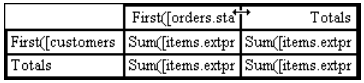


Figure 10-44 Resizing a cross tab column

2. Drag the right edge of the column to the desired position.
3. To set an exact size, do the following:
   1. Right-click any of the objects in the column that you want to resize.
   2. Choose Column properties from the context menu.
   3. Set the Width property to the desired value, then choose OK.

Formatting borders of a cross tab

A cross tab contains several types of borders, and you can change the color and thickness of individual borders. You customize borders by setting the Border properties of the cross tab, a row, or a column.

The following cross tab border properties are shown in Figure 10-45:

- OuterBorder is the border surrounding the entire cross tab.
- ColumnHeadingsBorder is the line between the column headings and cells.
- RowHeadingsBorder is the line between the row headings and cells.

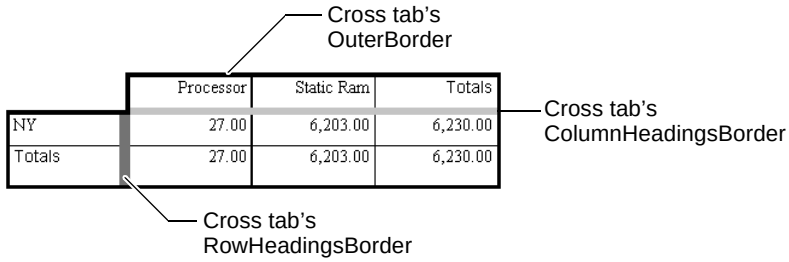


Figure 10-45 Formatting cross tab borders

Figure 10-46 and Figure 10-47 show the following border properties of the cross tab column:

- InnerBorder appears between column levels.

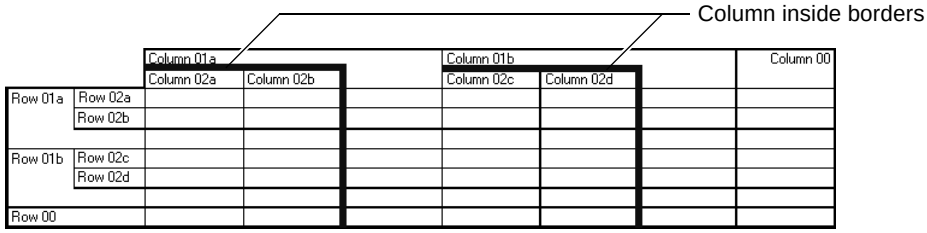


Figure 10-46 Cross tab column property—InnerBorder

- OuterBorder appears between columns at the same level.

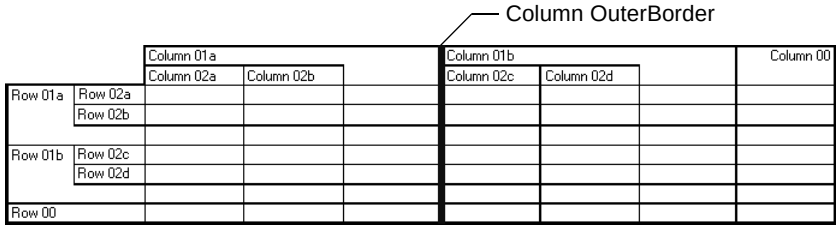


Figure 10-47 Cross tab column property—OuterBorder

Figure 10-48 and Figure 10-49 show the following border properties of the cross tab row:

- InnerBorder appears between row levels.

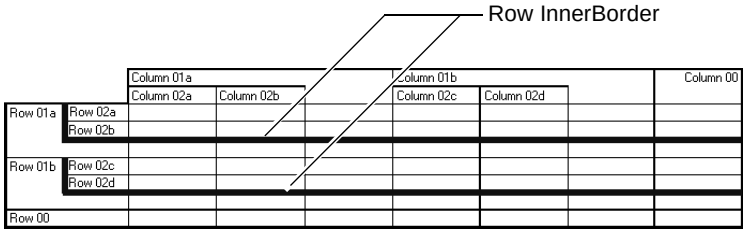


Figure 10-48 Cross tab row property—InnerBorder

- OuterBorder appears between rows at the same level.

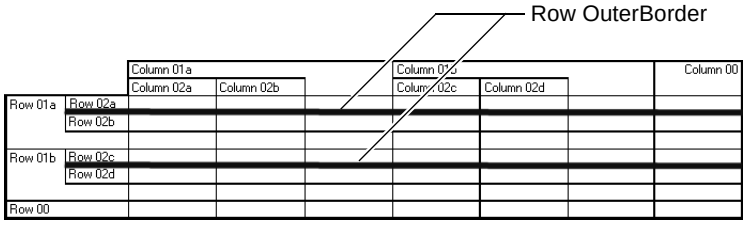


Figure 10-49 Cross tab row property—OuterBorder

How to set the borders of a cross tab

1 In the layout window, right-click an object in the cross-tab control, as shown in Figure 10-50.

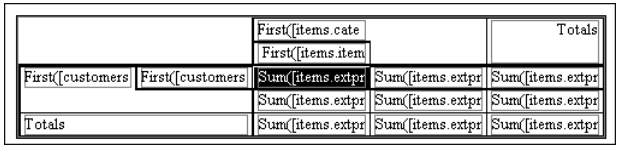


Figure 10-50 Setting cross tab borders

Figure 10-51 shows the context menu that appears.

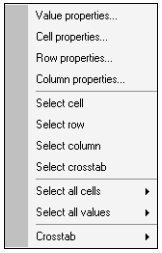


Figure 10-51 Context menu for setting cross tab borders

2 Choose one of the following:

- To customize the main borders of the cross tab, choose Crosstab➛Properties.

- To customize the borders of a row, choose Row properties.
- To customize the borders of a column, choose Column properties.

3 In the Properties page for the cross tab, cross tab row, or cross tab column, modify the border properties to set the color and thickness of the border. Choose OK.

Formatting numbers in a cross tab

By default, numbers in a cross tab's cells are displayed in standard format, as shown in the following example:

```
15,000.00
```

You can display the numbers in different formats. For example, you can display numbers with a currency symbol, a percent symbol, and with or without decimal values.

How to format numbers in a cross tab

1 In the layout window, right-click the data cell whose values you want to format, as shown in Figure 10-52.

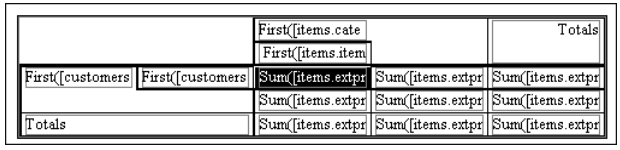


Figure 10-52 Select a cell to format its value

A context menu appears. To format the values of multiple data cells at the same time, press the Ctrl key and click each cell.

2 Choose Value properties from the context menu. Crosstab Value Properties appears, as shown in Figure 10-53.

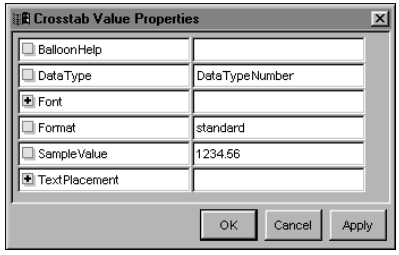


Figure 10-53 Cross tab value properties window

3 Set the Format property to the desired format. Choose OK.

Adding color to a cross tab

Add color to a cross tab for visual interest or to highlight certain information. You can apply a background color to the entire cross tab, to a particular row or column, or to certain cells. In Figure 10-54, the row and column totals are in one color while the grand total cell is in another.

The rest of the cross tab appears in the default color, Transparent.

	Cancelled	Closed	Open	Totals
CT	1,401,826.00	8,009,121.00	1,737,534.00	11,148,481.00
MA	278,291.00	12,460,186.00	6,187,923.00	18,926,400.00
NH	0.00	0.00	1,060,528.00	1,060,528.00
NJ	1,125,593.00	16,790,729.00	3,797,912.00	21,714,234.00
NY	3,193,710.00	12,772,557.00	11,711,727.00	27,677,994.00
Totals	5,999,420.00	50,032,593.00	24,495,624.00	80,527,637.00

Figure 10-54 Adding color to a cross tab

How to add color to a cross tab

1. To add color to the entire cross tab:
   1. In the layout window, right-click any object in the cross-tab control.
   2. Choose Crosstab➛Properties from the context menu.
   3. In the Properties page for the cross tab, set BackgroundColor to the desired color. You can select a color from the drop-down list. The background color of the entire cross tab appears in the selected color.
2. To add color to a particular row or column:
   1. Right-click any of the objects in the row or column of the cross-tab control.
   2. Choose Row properties or Column properties from the context menu.
   3. In the properties page for the cross tab row or column, set BackgroundColor to the desired color, then choose OK. The background color of the entire row or column appears in the selected color. If you also selected a color for the entire cross tab, the row or column appears in the color that you just selected, and the rest of the cross tab appears in the color that you selected for the cross tab.
3. To add color to a particular cell:
   1. Right-click the cell.
   2. In the properties page for the cell, set BackgroundColor to the desired color, then choose OK. The background color of the cell appears in the selected color. If you also selected a color for the column or row in which the cell appears, the cell appears in the color that you just selected, and the rest of the cells in the row or column appear in the color that you selected for the row or column.

Understanding cross tab sizing

When you run the report, the cross tab expands to accommodate its contents. The data it retrieves from the database determines the total number of columns and rows. The frame and page containing the cross tab adjust their size to accommodate the cross tab, even if you specify a fixed page size in the page layout.

If you place other controls in the same frame in which you place a cross tab, the cross tab can change the size and position of other controls as its size changes.

For example, if you place a control below a cross tab and the cross tab expands vertically, the control moves down. If you place a line below the cross tab and the cross tab expands horizontally, the line expands as well.

If you use a cross-tab control in your report, it helps to understand how e.Report Designer Professional resizes and repositions controls and frames during report generation. This information helps you decide how and where to place controls relative to the cross tab, and when to modify the default size and position property values.

Printing a cross-tab report

By default, a cross-tab report expands the size of a page if the contents of the cross tab exceeds a page. If you specify a page layout with a fixed page size, the page's height and width settings are ignored. This default behavior ensures that a cross-tab report always appears in full when viewed online, no matter how large it is.

When you view a large cross-tab report online, you scroll to the right or down to see all the data. If you print the oversized report to a printer or to PDF, e.Report Designer Professional splits the report at logical places and sets reasonable margins when printing it on multiple pages, as shown in Figure 10-55.

e.Report Designer Professional prints an oversized report onto multiple pages only if you maintain the default setting of Split Oversize Pages in Page Setup. The default setting is to select Split Oversize Pages. If you turn off this setting, only one page prints.

You can change how e.Report Designer Professional splits an oversized cross-tab report for printing by changing the values of the page's Printing and PDF properties. These properties are associated with the page component, not the cross tab. For more information, see the following sections.

Cross tab displayed online

		CT	MA	NH	NJ	NY	PA	QLD	RJ	SA	SP	VIC	WA	Totals
Cancelled	2,535,400.00	1,401,826.00	278,291.00	0.00	1,125,593.00	3,193,710.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8,534,820.00
Closed	18,195,424.0	8,009,121.00	12,460,186.0	0.00	16,790,729.0	12,772,557.0	16,038,747.0	0.00	0.00	57,450.00	2,126,000.00	330,000.00	393,900.00	87,194,114.0
In Evaluation	20,556,986.0	1,390,570.00	3,097,874.00	953,742.00	5,156,562.00	7,941,086.00	1,136,876.00	0.00	53,310.00	0.00	3,787,040.00	0.00	444,650.00	44,518,696.0
Open	18,536,643.0	1,737,534.00	6,187,923.00	1,060,528.00	3,797,912.00	11,711,727.0	1,872,240.00	60,668.00	4,340.00	950,000.00	622,500.00	0.00	0.00	46,542,015.0
Selected	0.00	0.00	0.00	0.00	0.00	1,010,287.00	364,917.00	0.00	0.00	0.00	0.00	0.00	0.00	1,375,204.00
Totals	59,824,453.0	12,539,051.0	22,024,274.0	2,014,270.00	26,870,796.0	36,629,367.0	19,432,780.0	60,668.00	57,650.00	1,007,450.00	6,535,540.00	330,000.00	838,550.00	188,164,849

First page of printed output

Second page of printed output

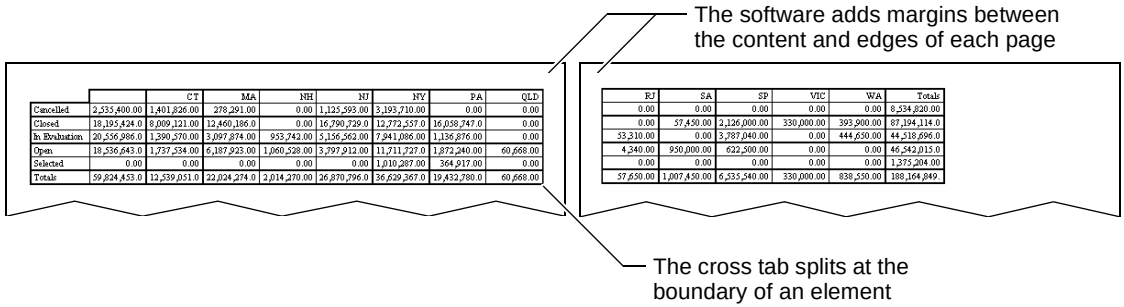


Figure 10-55 Comparing cross tab online and printed output

Figure 10-56 shows the Printing and PDF properties and their default values.

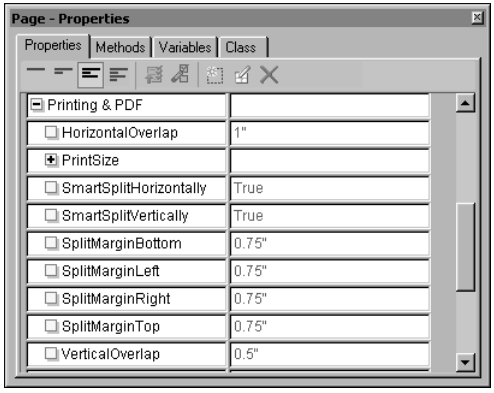


Figure 10-56 Default values of printing and PDF properties

Changing the margins for the printed pages

When e.Report Designer Professional splits an oversized report to print on multiple pages, it adds margins of 0.75 inches to all sides of the printed pages. You can change the size of these margins by setting the following Printing and PDF properties of the page: SplitMarginBottom, SplitMarginTop, SplitMarginLeft, and SplitMarginRight.

Setting print options to join multiple pages

There is no limit to how much an online page can expand vertically or horizontally to display a cross-tab report in full. When you print an oversized cross-tab report, e.Report Designer Professional, by default, splits a cross tab at the boundary of an element in the report. For an example of how a cross-tab report splits by default, see the illustration in "Printing a cross-tab report," earlier in this chapter.

If you want to print an oversized cross-tab report and tape all the pages together to create a single printout of the entire report, set the following Printing and PDF properties of the page:

- Set SmartSplitHorizontally to False. When set to False, e.Report Designer Professional splits the cross tab when its contents fill the printable area of the page, even if it results in truncated text. The default value, True, specifies that e.Report Designer Professional split the cross tab horizontally at the boundary of a cross tab element.
- Set SmartSplitVertically to False. When set to False, e.Report Designer Professional splits the cross tab when its contents fill the printable area of the page, even if it results in truncated text. The default value, True, specifies that e.Report Designer Professional split the cross tab vertically at the top edge of a row.
- Set HorizontalOverlap to specify the amount of horizontal overlap between adjacent pages. The amount of overlap determines the amount of paper available for folding and taping each page. The default amount is 1 inch.
- Set VerticalOverlap to specify the amount of vertical overlap between adjacent pages. The default amount is 0.5 inches.

Figure 10-57 shows how an oversized cross-tab report is printed onto two pages when SmartSplitHorizontally and SmartSplitVertically are set to False and HorizontalOverlap and VerticalOverlap are set to the default values.

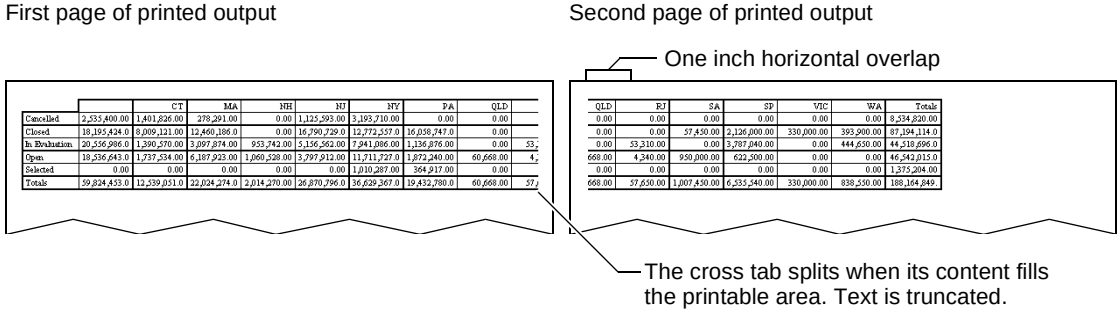


Figure 10-57 Printing an oversized cross-tab report

Figure 10-58 shows an enlarged view of the right edge of the first page and the left edge of the second page. It also shows where to fold the second page and where to place it on the first page to join both pages to form a continuous report.

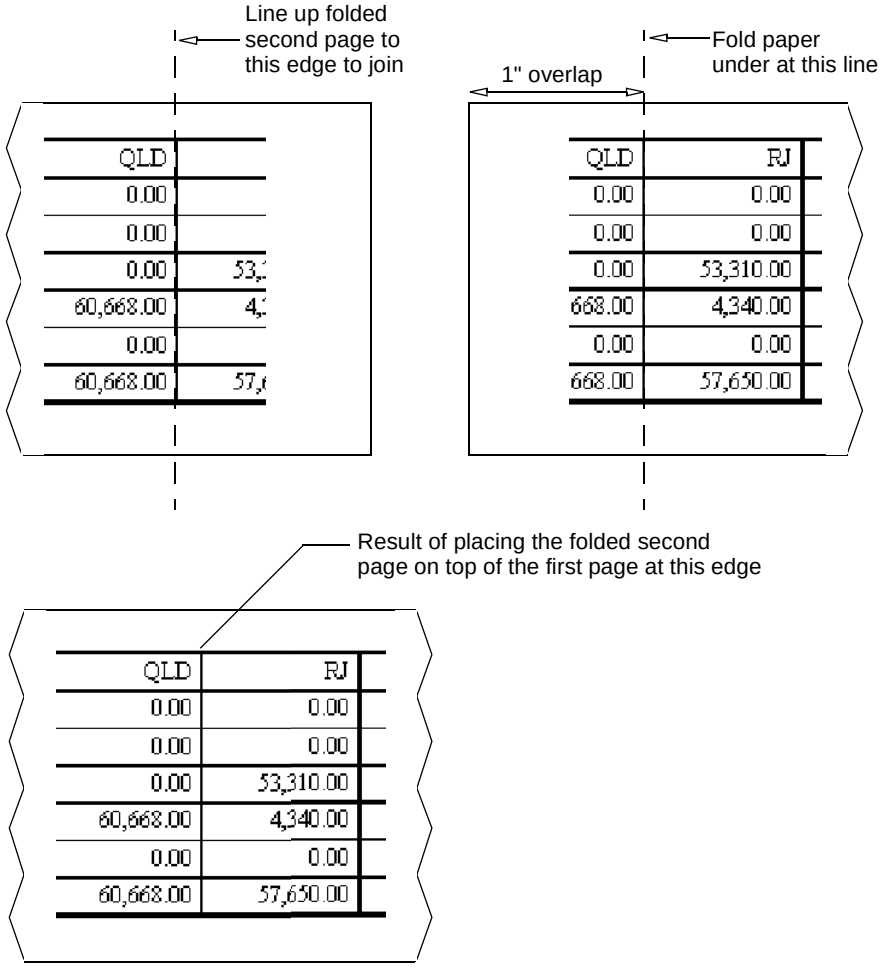


Figure 10-58 Oversized cross-tab report, enlarged views

Eliminating blank space on the right of the cross tab

A cross-tab report often displays blank space on the right of the report. This blank space is apparent if the report design displays a page element, such as a page number or date, in the right corner of the page, as shown in Figure 10-59.

00.00	238,700.00	231,000.00	492,200.00
50.00	1,550,000.00	158,400.00	1,716,650.00
79.00	11,111,320.00	13,702,933.00	46,542,015.00
32.00	0.00	689,095.00	1,010,287.00
57.00	20,770.00	173,531.00	364,917.00
39.00	20,770.00	862,626.00	1,375,204.00
53.00	28,871,390.00	60,871,124.00	188,164,849.00

Wednesday, August 25, 2004

Figure 10-59 Blank space in a cross-tab report

This blank space is equal to the distance between the right edge of the cross tab and the right edge of the frame containing the cross tab. By default, when you insert a frame, its size fits in the available flow space. Figure 10-60 shows the layout of the frame and the cross-tab control within it.

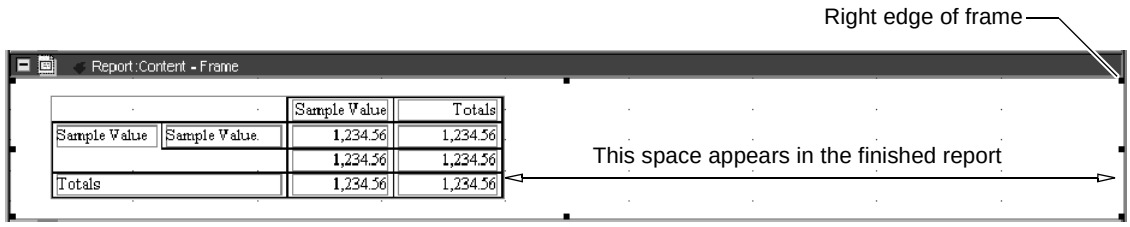


Figure 10-60 Frame layout showing cross-tab control and blank space

To eliminate this blank space, reduce the width of the frame containing the cross tab. Figure 10-61 shows the corrected layout and Figure 10-62 shows the corrected report output.

Design

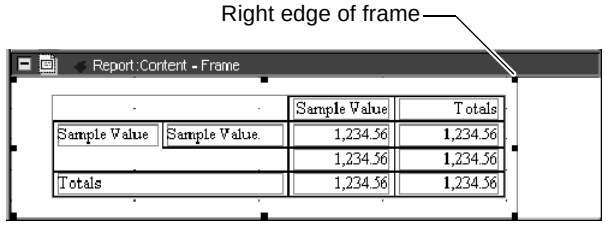
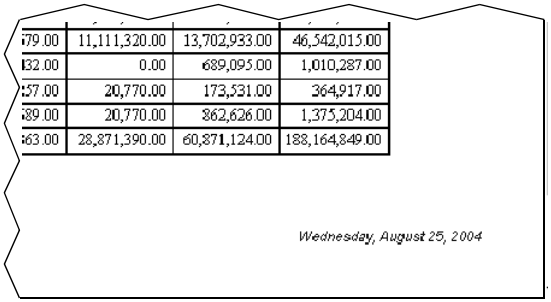


Figure 10-61 Corrected layout

Output



79.00	11,111,320.00	13,702,933.00	46,542,015.00
32.00	0.00	689,095.00	1,010,287.00
57.00	20,770.00	173,531.00	364,917.00
89.00	20,770.00	862,626.00	1,375,204.00
63.00	28,871,390.00	60,871,124.00	188,164,849.00

Wednesday, August 25, 2004

Figure 10-62 Corrected report output

Chapter

11

Displaying variable-length text data

This chapter contains the following topics:

- Displaying memo field data
- Using dynamic text controls
- Modifying the default resizing behavior
- Using dynamic text controls in nested frames
- Adjusting text formats for different rendering environments
- Controlling how a frame and its contents fit in multiple pages
- Understanding how dynamic text controls affect report layout
- Building a report with dynamic text controls
- Solving common output problems

Displaying memo field data

Databases often store large and variable amounts of text data in memo fields. A customer service database, for example, stores customer issues in a memo field. To display memo field data, you can use a text control with its TextPlacement➤MultiLine property set to True but the result can be an unprofessional-looking report.

When you place a text control in a report design, you can set its size to a height and width that are large enough to display the data. In the finished report, if the control is not wide enough to accommodate the data it contains, the data is truncated or overflow characters appear, depending on how you set the control's TextPlacement➤Clip property. If the control is too large, the report shows too much blank space.

e.Report Designer Professional solves this problem with the dynamic text control. This control resizes dynamically to accommodate the amount of data it needs to display.

Using dynamic text controls

The dynamic text control is a data control that displays:

- A large and variable amount of text.
- Text data that contains multiple style formats. For example, a dynamic text control can display text with paragraph styles such as bulleted lists and numbered lists, and character formats such as bold and italic.

A dynamic text control adjusts its size to accommodate varying amounts of data. It can span multiple pages for very large amounts of text. The frame containing the dynamic text control also adjusts its size and contents to accommodate the control. Similarly, if you select a dynamic page size, the page and flow resize to accommodate the dynamic content.

Use dynamic text controls for the following types of reports:

- Investment reports that contain formatted narratives which look more like the contents of a presentation than a standard report
- Reports that include memos ranging from zero lines to any number of lines
- Address labels with fields that can contain a variable number of lines
- Letters that are assembled from a number of formatted boilerplate paragraphs, where each paragraph is stored in a database row

Comparing a dynamic text control with a text control

Figure 11-1 shows the output of a report that uses text controls for the contact name, address 1, address 2 and customer notes. There are blank lines where a contact name or address 2 information is missing, and data in the notes for the second customer is truncated. Data is truncated because the text control's size is not large enough to contain all the data.

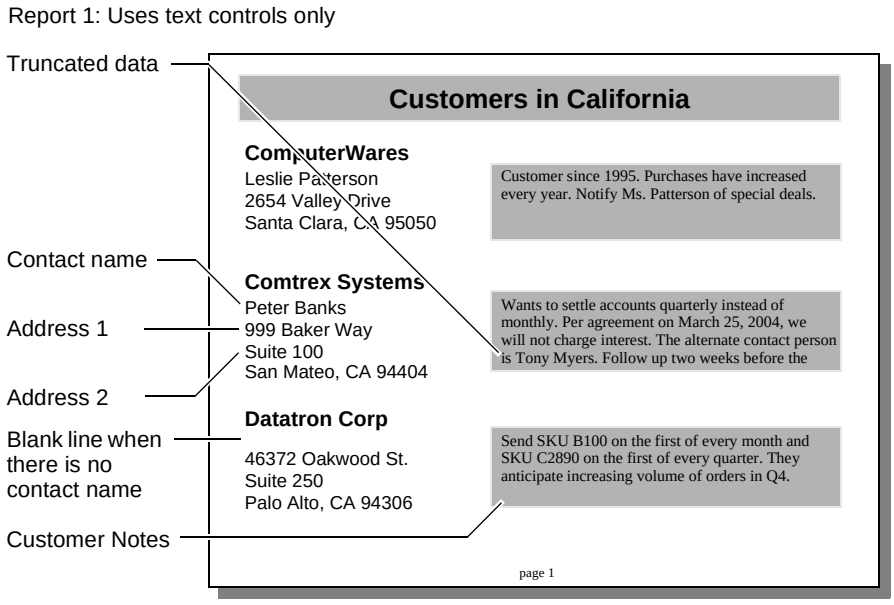


Figure 11-1 Output of a report that uses text controls only

Figure 11-2 shows the output of a report that uses dynamic text controls for the contact name, address 1, address 2, and customer notes. Notice that there are no blank lines for missing contact names or address 2 information.

Notice, too, that the colored background behind the customer notes adjusts to the size of the text block.

About the supported text formats

The dynamic text control can display text in the following formats:

- Plain text

  Plain text has no format tags, but it can contain ASCII control codes for specifying carriage returns, line feeds, tabs, and so on. The dynamic text control supports the CR, LF, and TAB control codes. All others are ignored.

- RTF

  The dynamic text control supports a subset of the RTF 1.6 standard. For a list of supported RTF tags, see Appendix A, “Working with RTF and HTML tags in a dynamic text control.”e.Report Designer Professional ignores tags it does not recognize.

- HTML

  The dynamic text control supports a subset of the HTML 4 standard. To use HTML formatting, you do not need to create a complete HTML document. The HTML tags are not enclosed in <HTML></HTML> or <BODY></BODY> tags to form a complete HTML document. For a list of supported tags, see Appendix A, “Working with RTF and HTML tags in a dynamic text control.” e.Report Designer Professional ignores tags it does not recognize.

Report 2: Contains dynamic text controls

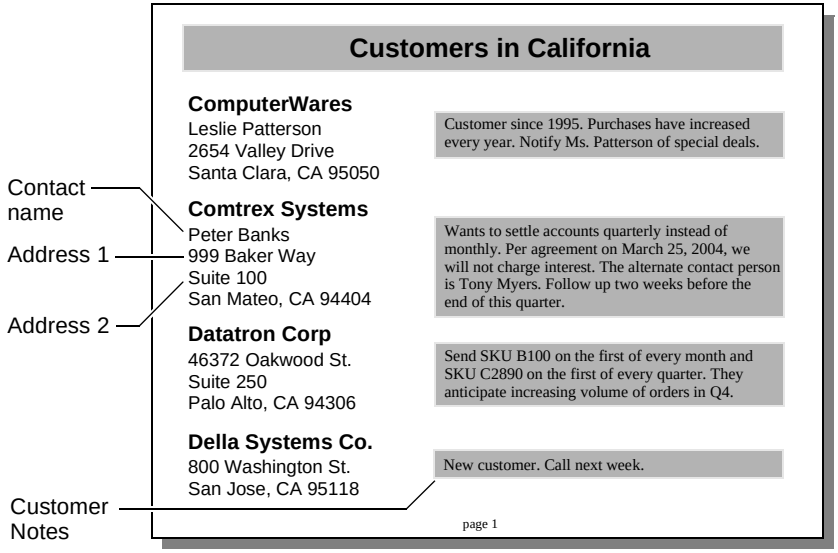


Figure 11-2 Output of a report that uses dynamic text controls

A single dynamic text control supports only one text format. A dynamic text control cannot, for example, support both HTML and RTF formats concurrently.

When you use RTF or HTML tagged text in a dynamic text control, the text can use more than one font. If the text does not specify a font internally, the Font properties of the control define the font for the text in the same way as other textual controls.

The tagged text can be modified programmatically. The methods SetTaggedText( ) and GetTaggedText( ) respectively set and get the tagged text in a dynamic text control. To access row data, you can call these methods from the OnRow( ) or

BuildFromRow( ) methods. To modify the content of a dynamic text control after processing all row data, override ProcessText( ). Make your changes to the content of the dynamic text control before the call to Super::ProcessText( ).

Using full text justification in a dynamic text control

For languages that display text in right-to-left orientation, e.Report Designer Professional does not support full text justification for a dynamic text control.

Considering performance issues

When a report contains dynamic text controls to display multiple style formats, e.Report Designer Professional parses every tag or control code in the formatted text. If the dynamic text control displays data of varying lengths, e.Report Designer Professional computes dynamic frame and page layouts at run time. These processing tasks increase the time it takes to generate the report.

Use the dynamic text control only if the data that you want to display contains multiple style formats or a variable amount of text. Otherwise, use the text control.

Placing a dynamic text control in a report design

You place a dynamic text control in a report design the same way that you place other controls. Because a dynamic text control is a data control that typically uses data from a data row, you almost always place it in a frame.

You can place a dynamic text control directly onto a page or subpage, and assign a literal value with embedded HTML tags, ASCII codes, or RTF tags to format a string. When you use this strategy, consider the effect on report generation time. Dynamic text controls take longer to generate than label controls. Unless the report requires static text that contains formatted words, you should use a label control instead to display a literal value.

How to place a dynamic text control in a report design

1 Choose Insert➛Dynamic Text Control and place the control in the desired frame. Component Properties appears, as shown in Figure 11-3.

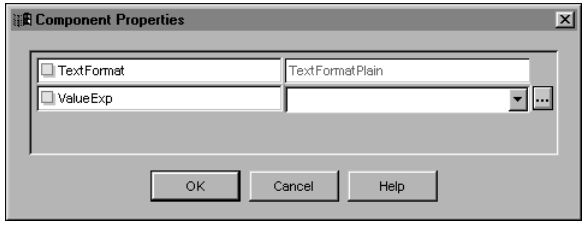


Figure 11-3 Placing a dynamic text control in a report design

2 Select a value for TextFormat. This value must match the text format in which the data is stored in the data source.

- Select TextFormatPlain, the default value, if the text to display is in ASCII format.
- Select TextFormatHTML if the text to display is in HTML format.
- Select TextFormatRTF if the text to display is in RTF format.

If the TextFormat setting does not correspond to the format of the text, the finished report displays unexpected results. For example, if the TextFormat setting is TextFormatPlain but the text is in HTML or RTF format, the text in the finished report displays all the HTML tags or RTF tags as literal text.

3 Specify a value for ValueExp. The value that you specify is typically a database field. You can use Expression Builder to select a field or create a value expression. Choose OK.

You can now customize the properties and behavior of the dynamic text control, if desired. These tasks are described later in this chapter.

Developing content for a dynamic text control

Typically, you display text from a data field in a dynamic text control. You set the control's TextFormat property according to the type of text in the data field. In other designs, you need to create formatted text or modify the text from the data field. When the data field contains unformatted text, these changes are straightforward. Make the changes by typing text in quotation marks and by using the Actuate Basic Language string functions.

Making changes to formatted text or typing static formatted text requires more complex development. In these cases, you typically create and test the static RTF or HTML in an external tool. For example, you can use Microsoft WordPad to develop text in RTF format and a web page design tool to develop text in HTML format. After developing the static text and saving it in a file, you can open the file in a simple text editor. You can then copy the content that you need from the file and paste it into the property editor or method editor in e.Report Designer Professional.

How to develop text in RTF format for a dynamic text control

1 To open Microsoft WordPad, choose Start➤All Programs➤Accessories➤WordPad.

2 In WordPad, type the text with the format that you need, for example:

```
Hello world
Hello stars
Hello moon
```

This text contains paragraph markers and uses bold and italic attributes.

3 Save the file. Close WordPad.

4 Open this file in Microsoft Notepad or another plain-text editor. You see text similar to the following lines:

```
{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0
   \fswiss\fcharset0 Arial;}}
{\*\generator Msftedit 5.41.15.1507;}\viewkind4\uc1\pard\f0
   \fs20 Hello world\par
\b\i Hello\i0  stars\b0\par
Hello moon\par
}
```

For a dynamic text control, you need only the text that contains the content and formatting attributes. You do not need the header and footer material. In this example, you need only the following text:

```
Hello world\par
\b\i Hello\i0  stars\b0\par
Hello moon\par
```

5 Create a dynamic text control in a report design.

6 In the ValueExp property of the dynamic text control, perform the following steps:

   1 Type a quotation mark(").

   2 Paste each line of text that you need. If a line ends with an RTF tag and is not the last line of text, type a space before you paste the next line.

   3 Type a final quotation mark. In this example, the ValueExp property contains the following value:

   ```
   "Hello world\par\b\i Hello\i0  stars\b0\par Hello moon"
   ```

   Note that there are two spaces between \i0 and stars. The first space terminates the RTF tag, \i0. The second space separates Hello and stars.

7 Run the report. In this example, the content of the dynamic text control looks similar to Figure 11-4.

Figure 11-4 Static RTF text in a report

Building complex HTML-tagged text

When you build HTML-tagged text in a method or the ValueExp property, you must understand how to set up the attributes of the text. HTML tags within a dynamic text control must be well formed and complete.

Changing text size

To change the size of text within a dynamic text control, use the SIZE attribute in a FONT tag. For example, the following code causes the text to appear as size 5:

```
<FONT SIZE="5">your text</FONT>
```

For the list of actual text sizes given a specific SIZE attribute, see Table A-8, "HTML parameters" in Appendix A, "Working with RTF and HTML tags in a dynamic text control."

Creating HTML lists

If you need to create an HTML list that uses several rows of data, the tags and associated text must all be aggregated into one control. To aggregate the text into one control, set the ValueType property of the control to SummaryControl. This type of control consumes all the row data for the enclosing section. Tags also must be aggregated into one control because HTML tag formatting does not persist across controls. For example, the tags for the ordered list, such as <OL> and <LI> must be placed within one long dynamic text control. The first part of the control must contain the <OL> tag, and the last entry in the list must contain the closing </OL> tag. In between, each individual <LI></LI> must be nested within the same dynamic text control. The aggregated text within a dynamic text control for an ordered list containing three items appears as follows:

```
<OL><LI>Item 1</LI><LI>Item 2</LI><LI>Item 3</LI></OL>
```

Building complex RTF-tagged text

When you build RTF-tagged text in a method or the ValueExp property, you must understand how to set up the attributes of the text. RTF tags begin with \. You can follow one RTF tag directly with another. For example, \b\i sets bold and italic attributes. A space or new line character terminates a tag or consecutive set of tags. You must terminate most RTF tags before the start of the text to format. If you put text immediately after the tag, the factory interprets the text as part of an unknown tag. For example, the following RTF code:

```
\b\iHello world
```

produces the following output, because \iHello is not a valid RTF tag:

world

If you need an RTF tag to affect only some of the text in a dynamic text control, enclose the tag and the text in braces, {}. For example, the following RTF code:

```
\b {\iHello} world
```

produces the following output:

***Hello* world**

Displaying Unicode characters in RTF-tagged text

You can include Unicode characters in RTF-tagged text by using the \uN tag, where N is the decimal value of the character's Unicode code point. RTF supports displaying one or more ANSI characters when the Unicode character is not available. Put the ANSI string immediately after the code point value. By default, the dynamic text control interprets one character after the code point value as the ANSI alternative to the Unicode value. To change this behavior, use the \ucN tag, where N is the number of characters to use in place of the Unicode character. Place the \ucN tag before the \uN tag. For example, the RTF code, \u9642*, produces a square bullet character with an asterisk (*) as an alternative character. The RTF code, \uc0\u9642, produces a square bullet character with no alternative character.

Indenting RTF-tagged text

The dynamic text control supports indenting RTF-tagged text. To indent a paragraph, use the \liN tag, where N is the number of twips to indent the text. To indent the first line of a paragraph by a different amount, use the \fiN tag, where N is the additional number of twips to indent the first line. To indent the first line less than other lines in the paragraph, use a negative number for N in \fiN.

Using fonts and colors in RTF-tagged text

If you use a single font or color in the dynamic text control, set the values of the appropriate properties on the Properties page. To use multiple fonts, you must set up a font table at the beginning of the text. Similarly, to use multiple colors, you must set up a color table. For example, the following RTF code sets up a font table and a color table:

```
{\fonttbl {\f0 Arial;}{\f1 Courier;}{\f2 Times New Roman;}}
{\colortbl \red0\green0\blue0;\red128\green128\blue128;}
```

The first line sets up a font table containing three font families, Arial, Courier, and Times New Roman. The second line sets up a color table containing two colors, black and gray. To set text to use these attributes, you use the \cfN and \fN tags respectively, where N is the zero-based entry of the color or font in the table.

To set the font size, use the \fsN tag, where N is the font size in half points.

The following RTF code example shows the use of color tables, font tables, text indenting, Unicode characters, and other RTF functionality:

```
{\fs48\fi-720\li720 {\cf1\bullet}\tab
\f1 Now is the winter of our discontent \f2 made glorious summer
   by this son of York}\par
{\fs40\fi-720\li1440 {\cf1\uc0\u9642}\tab
\f0 And all the clouds that lour'd upon our house in the deep
   bosom of the ocean buried}\par
```

This code produces the following output:

- Now is the winter of our discontent made glorious summer by this son of York
  - And all the clouds that lour'd upon our house in the deep bosom of the ocean buried

Table 11-1 describes the RTF tags that the RTF code uses to produce these effects.

Table 11-1 RTF tags in code example

RTF tags	Description
\fs48	Set font size to 24 points.
\fi-720	Indent the first line by minus 720 twips, which is minus half an inch. This value is added to the line indent value, shown in the next entry in this table, to make a total indent of 0 for the first line.
\li720	Indent lines by 720 twips, which is half an inch. This command ensures that text in wrapped lines aligns with the text after the tab tag, which is described later in this table.
\cf1	Use color 1 for the text inside the {} characters. Colors in the code table start at 0, so this color is the gray color.
\bullet	Write a bullet.
\tab	Write a tab. The text following this tag starts at the tab stop, at half an inch.
\f1	Use font 1 for the following text. Fonts in the font table start at 0, so the text uses Courier font.
\f2	Use font 2 for the following text.
\par	Start a new paragraph.
\fs40	Set font size to 20 points.
\uc0\u9642	Print a square bullet, with no alternative character if this Unicode code point is not available.

Modifying the default resizing behavior

When e.Report Designer Professional generates content for a dynamic text control, the application uses the following default algorithms to determine the size of the control:

- A dynamic text control can increase its height, but not reduce it. To override this default behavior, see "Shrinking a dynamic text control," later in this chapter.
- There is no limit to a dynamic text control's height increase. To override this default behavior, see "Setting the maximum height of a dynamic text control," later in this chapter.
- A dynamic text control can increase in height only. To override this default behavior, see "Increasing the width of a dynamic text control," later in this chapter.

Shrinking a dynamic text control

The default resizing behavior for a dynamic text control is to stay the same size or expand vertically. Extra space appears if the size of the control in the report design is larger than the actual amount of text. You can shrink a dynamic text control to completely surround the text if you do not want to display extra space.

Shrinking a dynamic text control can also make it disappear if there is no data. This feature is useful for eliminating blank lines in the following situations:

- An Address2 field can contain suite numbers but this field is sometimes empty.
- An Issues field contains notes about an order but this field is sometimes empty.

To shrink a dynamic text control:

- Set its Dynamic Size and Position➤CanReduceHeight property to True.
- If you want to set a limit to how small the control gets, specify a value for Dynamic Size and Position➤MinimumHeight as well. If you specify a value that is larger than the control's initial height—specified by the Size➤Height property—the control does not get taller. In this case, the control's Height value takes precedence over MinimumHeight, and the initial height sets the control's minimum height.
- To hide the control when there is no text, leave MinimumHeight at its default value of zero.

You can shrink a dynamic text control vertically only. You cannot reduce a dynamic text control's width when you run or print the report.

Moving a control up when a dynamic text control above it shrinks

When a dynamic text control shrinks from its initial size, the gap between the dynamic text control and another control directly below it increases. To maintain a constant space between the two controls, set the lower control's Dynamic Size and Position➤CanMoveUp property to True.

To decrease the frame's height so there is no extra space below the lower control, set the frame's Dynamic Size and Position➤CanReduceHeight property to True, as shown in Figure 11-5.

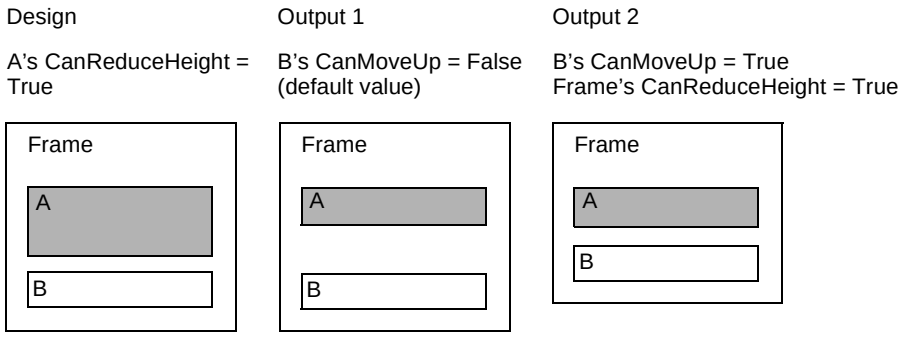


Figure 11-5 Maintaining constant space between controls

Setting the maximum height of a dynamic text control

The default behavior for a dynamic text control is to expand vertically to contain all the data. You can set a limit to how large it gets by setting a value for Dynamic Size and Position➤MaximumHeight. This setting is useful if you do not want a dynamic text control to expand to multiple pages or if you want to limit its height to align its bottom with an adjacent control, or group of controls, of a fixed size.

If you specify a MaximumHeight value that is smaller than the control's initial height—specified by the Size➤Height property—the control does not decrease its height. In this case, the control's Height value takes precedence over MaximumHeight, and the initial height sets the control's maximum height.

Increasing the width of a dynamic text control

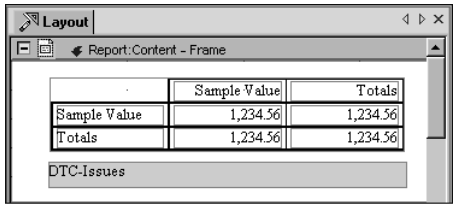
The default behavior for a dynamic text control is to expand only vertically to accommodate its content. If you place a dynamic text control below another dynamically sized component, such as a cross tabulation or nested frame, you can specify that the dynamic text control expand its width if the dynamically sized component above or below it expands horizontally. To do so:

- Set both components to the same initial size. Align the left and right sides of both components.

- Set the dynamic text control's Dynamic Size and Position➤HorizontalSize property to HorizontalSizeRelative.
- Set the dynamic text control's Dynamic Size and Position➤CanIncreaseWidth to True (the default).

Figure 11-6 shows an example report layout with a dynamic text control and a cross tabulation, and the output.

Design: Dynamic text control is below a cross tabulation. Both components are the same size and the dynamic text control's HorizontalSize property = HorizontalSizeRelative.



Output: Dynamic text control's width expands when the cross tabulation width expands. The dynamic text control's content, however, does not expand beyond the control's original width.

		CT	MA	NH	NJ	NY
Cancelled	2,535,400.00	1,401,826.00	278,291.00	0.00	1,125,593.00	3,193,710.00
Closed	18,195,424.00	8,009,121.00	12,460,186.00	0.00	16,790,729.00	12,772,557.00
In Evaluation	20,556,986.00	1,390,570.00	3,097,874.00	953,742.00	5,156,562.00	7,941,086.00
Open	18,536,643.00	1,737,534.00	6,187,923.00	1,060,528.00	3,797,912.00	11,711,727.00
Selected	0.00	0.00	0.00	0.00	0.00	1,010,287.00
Totals	59,824,453.00	12,539,051.00	22,024,274.00	2,014,270.00	26,870,796.00	36,629,367.00

The order may be downgraded if there is a downturn in the industry.

Figure 11-6 Dynamic text control, cross tabulation, and output

Notice that the content in the dynamic text control does not expand beyond the control's original width. The width of content in a dynamic text control is fixed at design time. You cannot modify this behavior using the control's properties.

Setting the maximum width of a dynamic text control

You can set a limit to how wide a dynamic text control gets, relative to an adjacent dynamically sized component, by specifying a value for Dynamic Size and Position➤MaximumWidth. This setting is useful if you want to limit its width to align its right side with another control, or group of controls, that have a fixed width.

If you specify a MaximumWidth value that is smaller than dynamic text control's initial width, specified by the Size➤Width property, the control does not decrease its width. In this case, the control's Width value takes precedence over MaximumWidth, and the initial width sets the control's maximum width.

If you do not want the dynamic text control's width to expand at all, set its Dynamic Size and Position➤CanIncreaseWidth property to False.

Using dynamic text controls in nested frames

A nested frame is a frame within a frame. You can use a nested frame to group some, but not all, controls in a frame. This technique is useful if you want to keep certain controls together and manipulate them as a single object.

e.Report Designer Professional uses the same resizing principles for nested frames as it does for individual controls in a frame. If a nested frame contains a dynamic text control and it expands to accommodate the control, the nested frame's parent frame also expands. In this case, the resizing behavior of the parent frame is the same as if the nested frame is a dynamic text control. You can, therefore, resize the nested frame the same way that you can resize a dynamic text control. For example, you can shrink a nested frame, set its minimum and maximum height, and so on.

Figure 11-7 and Figure 11-8 shows an example of how e.Report Designer Professional resizes controls, nested frames, and parent frames. A and B are dynamic text controls, C is a static control.

Design

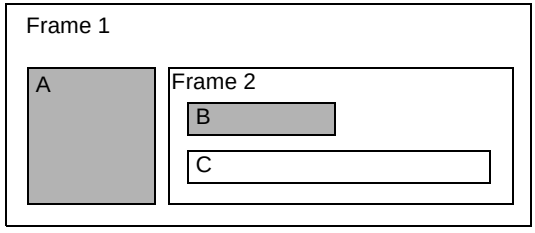


Figure 11-7 Design for resizing controls and frames

Output

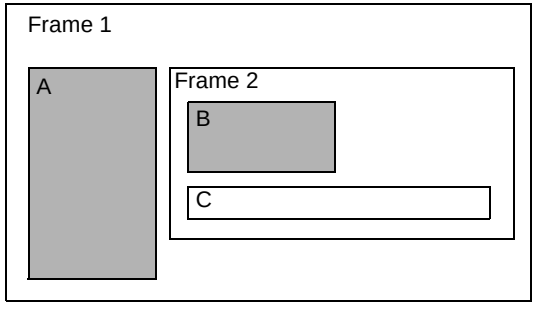


Figure 11-8 Output for resized controls and frames

Using dynamic text controls in different types of frames

The way e.Report Designer Professional resizes a frame to accommodate a dynamic text control depends on where you place the frame. You can place a frame in the following slots of a section component:

- Content
- Before
- After
- PageHeader
- PageFooter

Using dynamic text controls in Content, Before, and After frames

Frames in the Content, Before, and After slots expand to accommodate dynamic text controls. If their contents exceed a page, the frames and contents divide across multiple pages.

Using dynamic text controls in PageHeader frames

A frame in the PageHeader slot expands to accommodate dynamic text controls in the same way as a Content, Before, and After frame. Unlike these frames, however, a run-time exception occurs if a PageHeader frame is too large to fit in a flow.

If you use a dynamic text control in a PageHeader frame, set the frame's MaximumHeight property to prevent the frame from getting too large.

Using dynamic text controls in PageFooter frames

Unlike frames in the other slots, a PageFooter frame does not change size. At run time, e.Report Designer Professional places the PageFooter frame in the page before processing its contents. The frame's position and size are set before a dynamic text control gets its data.

If you use a dynamic text control in a PageFooter frame, set the control's MaximumHeight property appropriately so that the control fits in the frame. Alternatively, you can specify a static height for the dynamic text control by setting its initial height, then setting its CanIncreaseHeight property to False.

Using the first technique, the dynamic text control expands dynamically up to the specified maximum height. Using the second technique, the dynamic text control's height is fixed at design time.

Adjusting text formats for different rendering environments

When e.Report Designer Professional generates a report, it attempts to produce consistent output for all the rendering environments it supports, which include view perspective, DHTML Viewer, PDF Viewer, printing on a Windows platform, and server printing on Windows and UNIX platforms. A report, however, inevitably appears or prints differently in different rendering environments.

If the report you design appears or prints only on a specific device, you can optimize the dynamic text output for that particular rendering environment. You can modify the dynamic text control's default Text Layout property values to adjust line and paragraph spacing, and tab positions as needed. For example, by modifying the LineWidthPadding property value, you can increase or decrease the amount of space on the right of the text, if the rendering environment displays too little or too much blank space there.

To get the best output results for a rendering environment or environments, you probably have to adjust property values through trial and error. Remember to always check the resulting output, especially if the report appears or prints on multiple rendering environments.

The following sections describe how you can fine-tune the text layout for a dynamic text control.

Adjusting the space on the right of the text

When e.Report Designer Professional generates the text contents of a dynamic text control, it uses fixed line breaks in a fixed content width, so that all rendering environments display the same number of lines of text. e.Report Designer Professional determines where to break a line by calculating the available line width for text. To ensure that text does not overflow the width of the control when rendered slightly larger in some environments, it builds in a padded amount of white space when calculating the available line width.

The default padding value is 7.5% of the width of the control. For example, if the control is 200 pt wide, the available line width for text is approximately 186 pt (200/1.075).

You can modify the padding value to optimize the available line width for a particular rendering environment, as shown in Figure 11-9.

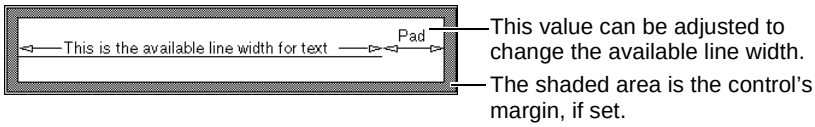


Figure 11-9 Padding value

Figure 11-10 shows how the amount of blank space at the end of lines can differ depending on the rendering environment.

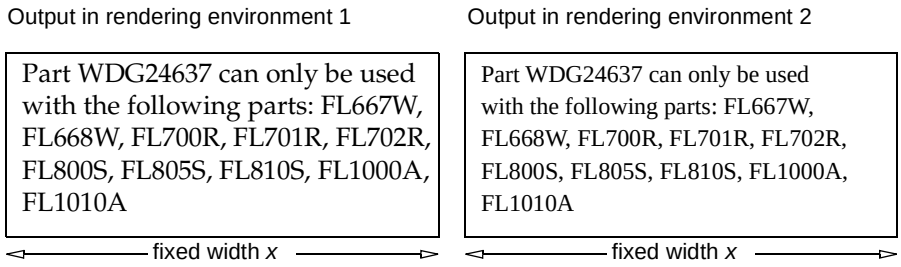


Figure 11-10 Rendering environments

If the report appears or prints to the second rendering environment in the example above, you can choose to reduce the space on the right of the text. To do so, decrease the default padding value by setting the dynamic text control's Text Layout➤LineWidthPadding property to a lower value. Note, however, that by reducing the LineWidthPadding value to optimize for the second rendering environment, text appearing on the first rendering environment might overflow or truncate.

Figure 11-11 shows the effect of reducing LineWidthPadding for the previous examples. By decreasing the padding, more words fit on a line. The text layout improves for the second rendering environment but text is now truncated in the first rendering environment.

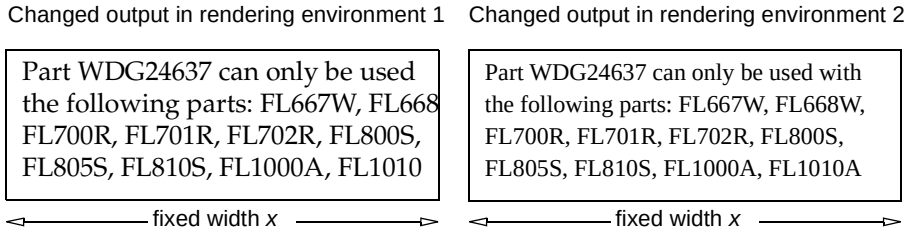


Figure 11-11 Reducing LineWidthPadding

Always check the resulting output, especially if the report appears or prints on multiple rendering environments.

Adjusting the space between lines of text

The following factors control vertical space between lines:

- The line height. e.Report Designer Professional computes the line height from the font at run time. The line height accommodates the largest font in the line, plus a small amount of extra space. The amount of space varies depending on the font.

- The line spacing (single line, 1.5 line, double line). This value is multiplied by the line height value to calculate the vertical space between lines.
- The space between lines in a paragraph. This is a fixed amount of space that is added between the lines of a paragraph.

For example, if the first line's vertical position is 8.8", its line height is 0.175", the line spacing is 1.5, and the fixed space between lines is 0.05", the second line's vertical position is calculated as follows:

```
8.8" + (0.175" x 1.5) + 0.05" = 9.112"
```

Figure 11-12 and Figure 11-13 show how e.Report Designer Professional calculates the second line's vertical position given the values above. Example 1, Figure 11-12 shows the results with a line spacing of 1.5. Example 2, Figure 11-13 shows the results with a line spacing of 1.

Example 1: 1.5 line spacing

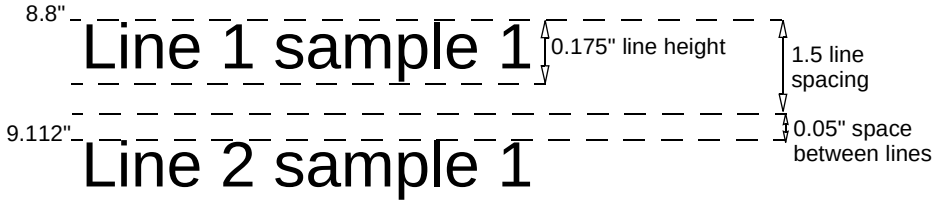


Figure 11-12 Line spacing set to 1.5

Example 2: 1.0 line spacing

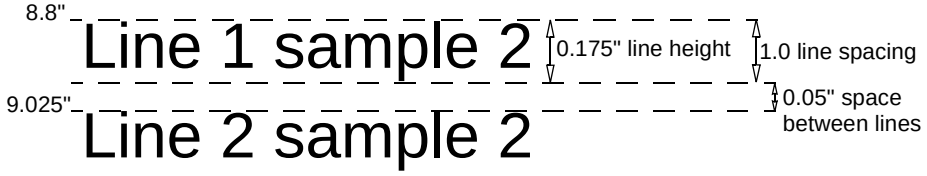


Figure 11-13 Line spacing set to 1

To increase or decrease the vertical space between lines, you can set the following Text Layout properties:

- LineSpacing

  This property is similar to line spacing (single line, 1.5 line, double line) properties in word processors, such as Microsoft Word. The value that you specify is multiplied by the line height value to calculate the actual line spacing.

- SpaceBetweenLines

  Specify the space between lines in a paragraph. This value specifies a fixed amount of space to add between lines. The LineSpacing value does not affect SpaceBetweenLines.

The LineSpacing and SpaceBetweenLines properties control text layout for plain text and HTML output. RTF text typically contains tags that specify line spacing and line leading. RTF tag values have precedence over the control's LineSpacing and SpaceBetweenLines property values.

Adjusting the space between paragraphs

The space between paragraphs for plain text and HTML output is controlled by the following factors:

- The line spacing, specified by the Text Layout➤LineSpacing property.
- The Text Layout➤SpaceBetweenParagraphs property value. This value specifies a fixed amount of space to add between paragraphs. The default value is 0.08".

RTF text typically contains tags that specify line and paragraph spacing. These tag values have precedence over the control's LineSpacing and SpaceBetweenParagraphs property values.

To increase the space between paragraphs, specify a value higher than 0.08" for the SpaceBetweenParagraphs property. To decrease the space between paragraphs, specify a value lower than 0.08".

Figure 11-14 and Figure 11-15 show the effects of changing the SpaceBetweenParagraphs property value.

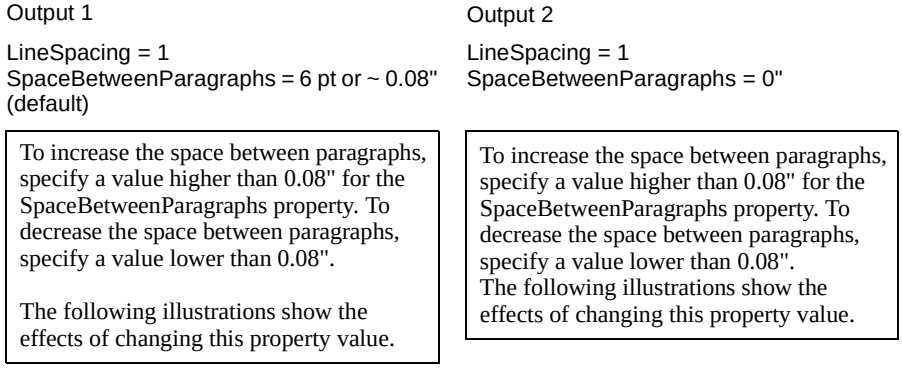


Figure 11-14 Examples of changing SpaceBetweenParagraphs

Output 3

LineSpacing = 1
SpaceBetweenParagraphs = 0.15"

To increase the space between paragraphs, specify a value higher than 0.08" for the SpaceBetweenParagraphs property. To decrease the space between paragraphs, specify a value lower than 0.08".

The following illustrations show the effects of changing this property value.

Output 4

LineSpacing = 1.2
SpaceBetweenParagraphs = 0.08" (default)

To increase the space between paragraphs, specify a value higher than 0.08" for the SpaceBetweenParagraphs property. To decrease the space between paragraphs, specify a value lower than 0.08".

The following illustrations show the effects of changing this property value.

Figure 11-15 Examples of changing SpaceBetweenParagraphs (continued)

Adjusting the space between tabbed text

The space between tabbed chunks of text is determined by the following factors:

- The width between tab stops. The default value is 0.5". The first tab stop is at 0.5", the second is at 1", the third is at 1.5", and so on.
- The calculated width of each text chunk. This width includes the width of the actual text plus a padded amount of space. When generating the content of a dynamic text control, e.Report Designer Professional adds extra space when it calculates the text chunk's end position. The default padding value is 7.5% of the width of the actual text. This technique prevents tabbed text from overlapping when rendered in environments whose font metrics differ from those used to compute tabs at generation time.

  For example, if the text chunk's start position is 2.5", its width is 1.45", and the padding value is 7.5%, the end position of the text chunk is calculated as follows:

  ```
  2.5" + (1.45" x 1.075) = 4.059"
  ```

  Figure 11-16 shows the calculation. For legibility, the actual widths are not to scale.

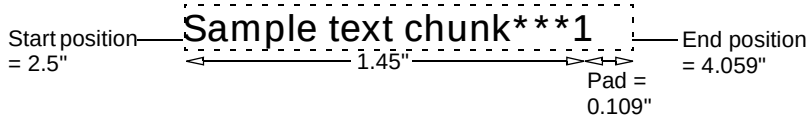


Figure 11-16 Calculating positions of text chunks

If there are tab stops at 4" and 4.5", the 4.5" tab stop is used as the start position of the next text chunk. Without the padding, the end of the current text chunk would be 3.95" (2.5" + 1.45"), and the 4.0" tab stop would be used. With only 0.05" separating the two text chunks, there might be text overlap in some rendering environments.

Figure 11-17 shows the results when no padding is used, and when it is. For legibility, the actual widths are not to scale.

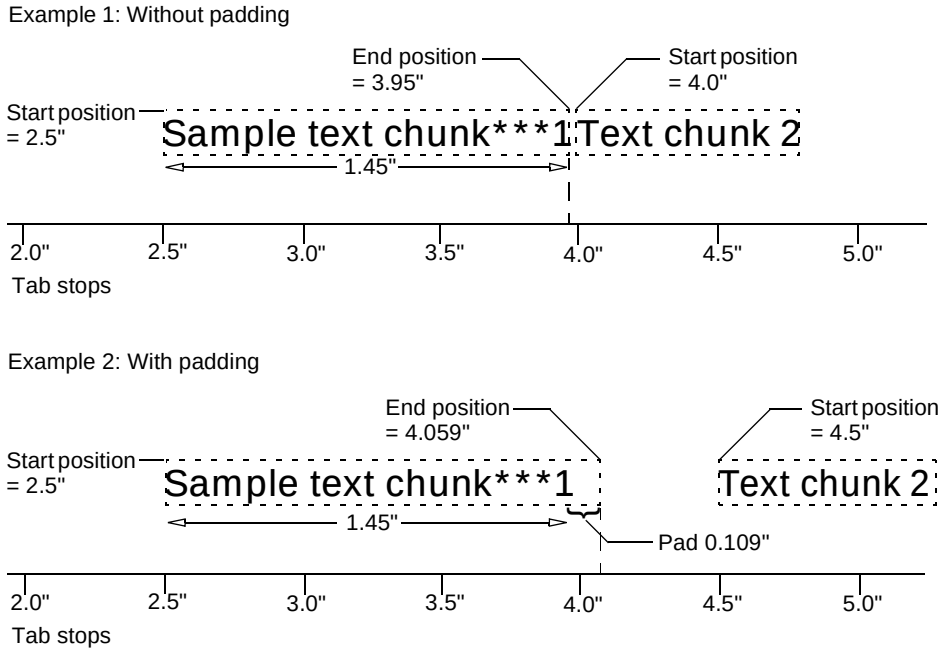


Figure 11-17 Formatting with and without padding

To increase or decrease the amount of space in and between tabbed text, set the following Text Layout properties:

- TabPadding
  This property, which appears in the Text Layout property group, specifies the percentage by which the width of the text chunk increase when e.Report Designer Professional calculates the text chunk's end position. To reduce the amount of space between the text and the end of the text chunk, set TabPadding to a value less than the default value of 7.5%. If the text overlaps, to increase the amount of space, set TabPadding to a value higher than 7.5%.

- TabSpacing
  This property specifies the width between tab stops. This value also determines the automatic indentation of bulleted and numbered lists. To decrease the amount of space between text chunks, set TabSpacing to a value less than the default value of 0.5". To increase the amount of space, set TabSpacing to a value higher than 0.5".

Controlling how a frame and its contents fit in multiple pages

When a frame increases in height to accommodate one or more dynamic text controls that increase in height, it is sometimes necessary to split the contents of the frame over multiple pages. If you specify a fixed page size in Page Setup, rather than dynamic page size, e.Report Designer Professional computes logical places to split a frame and its contents.

You can fine-tune how and where splits occur by modifying the default values of the frame or dynamic text control's Pagination properties.

The following sections describe the different ways that you can split the contents of a frame.

Specifying how a frame and its contents should split

e.Report Designer Professional's default behavior is to split a frame and its contents only if the frame contains at least one dynamic text control. It splits the frame contents and places the segments in the most space-efficient manner, while presenting a reasonable layout. For example, e.Report Designer Professional ensures that no segments are too small at the top and bottom of pages when it splits a frame over multiple pages.

You can modify how e.Report Designer Professional splits a frame or a dynamic text control by setting the Pagination➤AutoSplitVertical property.

You can specify that e.Report Designer Professional takes one of the following split actions:

- Do not split the frame or control. To specify this action, set AutoSplitVertical to DoNotSplit. Only data that fits in the flow appears. The remaining data truncates. Use this setting to limit data to one page.
- Split the frame, excluding header and footers, only if it is the first one in the flow. Subsequent frames are placed on the next page, where again only the first frame is split, if necessary. To specify this action, set AutoSplitVertical to SplitIfNecessary. Use this setting to minimize the number of split frames. The trade-off is an increase in the amount of empty space on pages and, therefore, the number of pages.
- Split the frame and its contents to maximize the use of space in the flow. To specify this action, set AutoSplitVertical to SplitIfPossible. For frames that contain dynamic text controls, the default setting—DefaultSplitting—yields the same results as SplitIfPossible. Use this setting to minimize the number of pages. The trade-off is more data is split up across pages.

For more information about the AutoSplitVertical property and its different values, see *Programming with Actuate Foundation Classes.*

Splitting frames that contain dynamic text controls

Figure 11-18, Figure 11-19, and Figure 11-20 show the different results when:

- You set the frame's Pagination➛AutoSplitVertical property to DefaultSplitting (the default), SplitIfNecessary, and DoNotSplit, respectively.
- The frames contain at least one dynamic text control.

In Figure 11-18 and Figure 11-19, Frame 1 and Frame 2 can be in the following configurations:

- Both frames are in the same Content slot.
- Frame 1 is in a Before slot, and Frame 2, in a Content slot.
- Frame 1 is in a Content slot, and Frame 2, in an After slot.

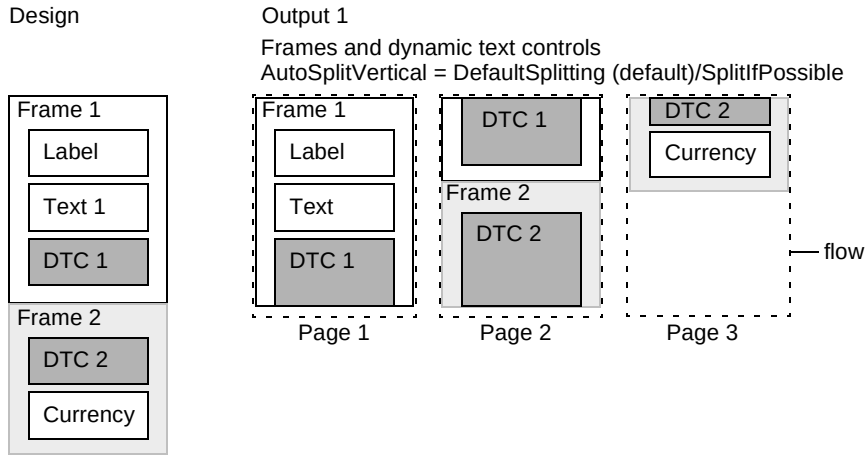


Figure 11-18 Output 1 dynamic text control

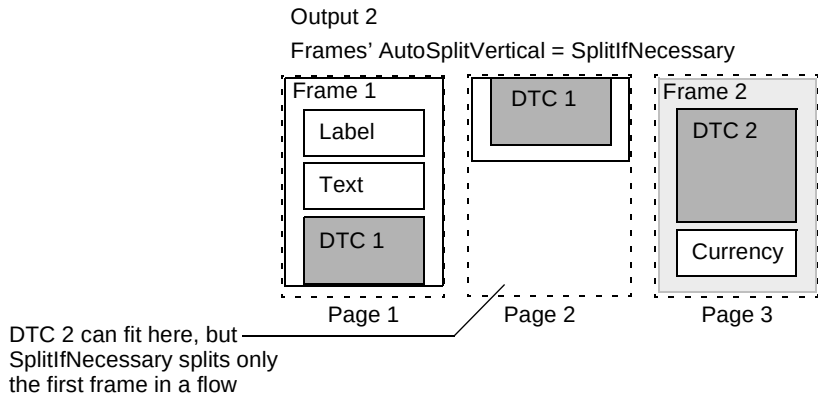


Figure 11-19 Output 2 dynamic text control

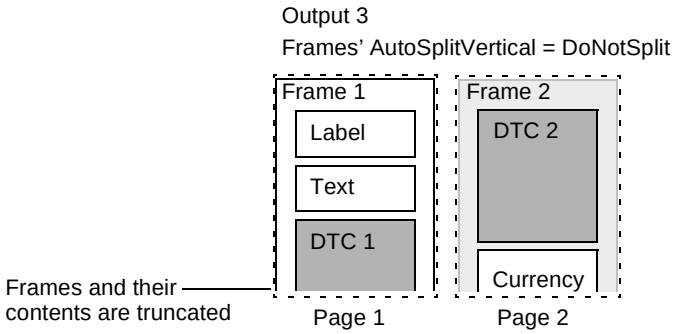


Figure 11-20 Output 3 dynamic text control

Splitting frames that do not contain dynamic text controls

Figure 11-21, Figure 11-22, and Figure 11-23 show examples of results when frames do not contain dynamic text controls, and when the Pagination➛AutoSplitVertical property is set to SplitIfPossible, SplitIfNecessary, and DoNotSplit, respectively. The default value of DefaultSplitting has no effect on frames that do not contain dynamic text controls.

In the illustrations, Frame 1 and Frame 2 can be in the following configurations:

- Both frames are in the same Content slot.
- Frame 1 is in a Before slot, and Frame 2, in a Content slot.
- Frame 1 is in a Content slot, and Frame 2, in an After slot.

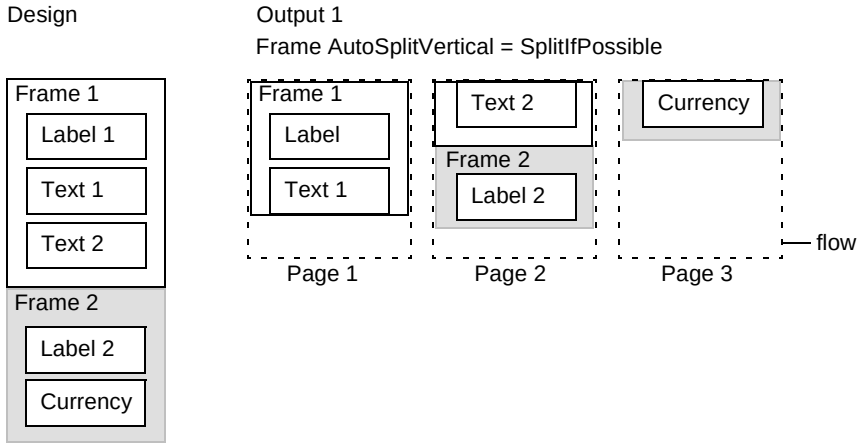


Figure 11-21 Frame AutoSplitVertical = SplitIfPossible

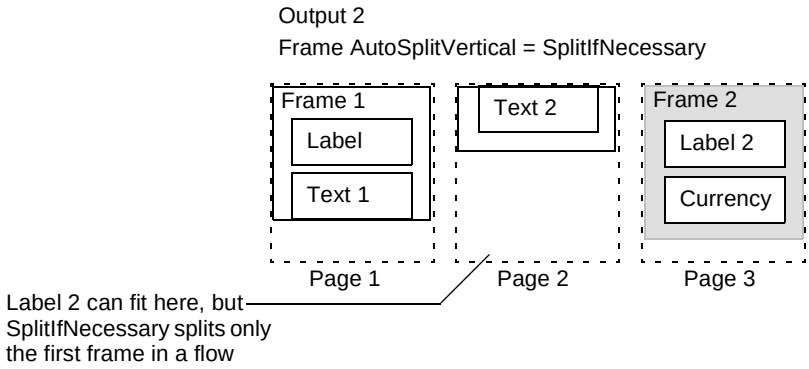


Figure 11-22 Frame AutoSplitVertical = SplitIfNecessary

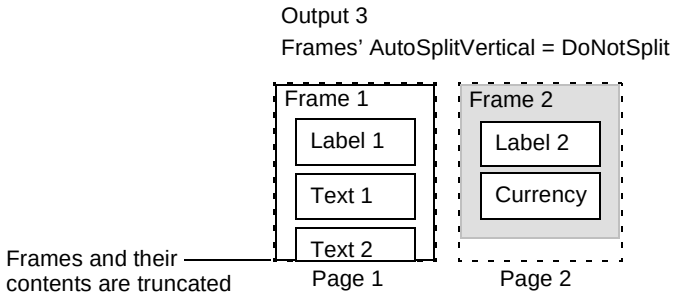


Figure 11-23 Frames' AutoSplitVertical = DoNotSplit

Specifying an area of the frame that should not be split

To present a reasonable layout when splitting frames and their contents, e.Report Designer Professional ensures that the first and last segments of a split frame are at least one inch in height. This prevents segments from being too small.

To modify this default behavior, consider the following behavior when you change the values of the frame's Pagination➛NoSplitTop and NoSplitBottom properties:

- NoSplitTop specifies the height of the area that must not be split at the top of the frame, or the minimum height of the first segment.
- NoSplitBottom specifies the height of the area that must not be split at the bottom of the frame, or the minimum height of the last segment.

Compare the layouts in the Figure 11-24 and Figure 11-25. The first layout shows the results of using the default NoSplitTop and NoSplitBottom setting. The second layout shows the results of using a smaller NoSplitTop and NoSplitBottom value. In output 2, the smaller NoSplitTop value enables the first frame segment of the second frame to fit in the small space at the bottom of page

one, and the smaller NoSplitBottom value results in a smaller last segment on page three. However, output 1 looks better than output 2.

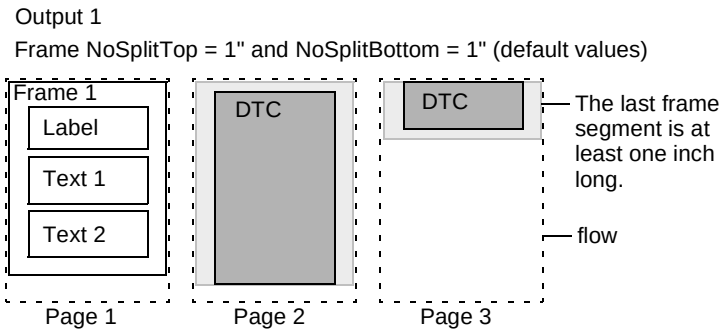


Figure 11-24 Result of default NoSplitTop and NoSplitBottom

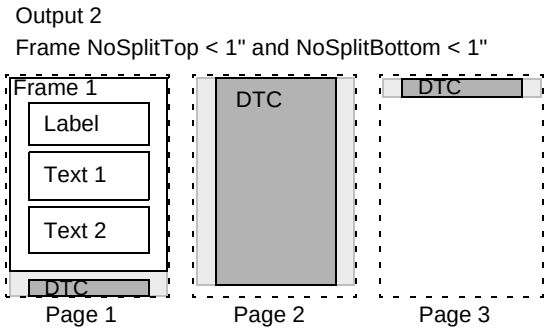


Figure 11-25 Result of NoSplitTop and NoSplitBottom set to <1

You can also use the NoSplitTop and NoSplitBottom properties to keep a group of related controls together at the top or bottom of the frame, to prevent some of them from being divided across two pages.

If the frame's height is less than the sum of NoSplitTop and NoSplitBottom, the frame is not split.

Setting margins for the segments of a split dynamic text control

There are two types of margins that you can set for a dynamic text control:

- Margins➛Bottom, Left, Right, and Top
  Margin properties specify a blank area between the content and the border of the dynamic text control. The default setting for the margin values is zero. Margins are useful if you set a background color or border for the dynamic text control and want some space between the text and the edges of the control.

Compare Output 1 and Output 2 in Figure 11-26. The first shows the default setting of no margins, the second shows equal margins set on four sides of the control. The dotted line does not appear in the output. The illustration uses this line to show the margins.

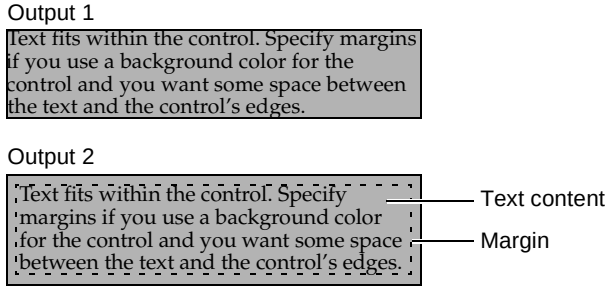


Figure 11-26 Comparison of no margin and margins set

- Pagination➤SplitMarginBottom and SplitMarginTop

  These properties are relevant only if the dynamic text control is split to fit on multiple pages. SplitMarginBottom specifies a blank area between the bottom edge of each segment, except the last, and its contents. SplitMarginTop specifies a blank area between the top edge of each segment, except the first, and its contents.

  The default setting for SplitMarginBottom and SplitMarginTop is zero.

The difference between the Margins properties and the SplitMarginBottom and SplitMarginTop properties is evident when a dynamic text control splits into multiple segments. The top and bottom margins that you set using Margins➤Top and Margins➤Bottom appear at the top and the bottom of the control. When the control splits, the top of the control is in the first segment, and the bottom of the control is in the last segment.

Figure 11-27 shows the effects of margins on a split dynamic text control.

Notice there is no space between the content and the bottom edge of the first segment. In the second segment, there is no blank space on either the top or bottom of the segment. In the third segment, there is no blank space at the top.

If you want space on the tops and bottoms of the segments, modify the values of SplitMarginTop and SplitMarginBottom. Typically, you set SplitMarginTop to the same value as Margins➤Top, and SplitMarginBottom to the same value as Margins➤Bottom. This gives the segments of the dynamic text control consistent top and bottom margins.

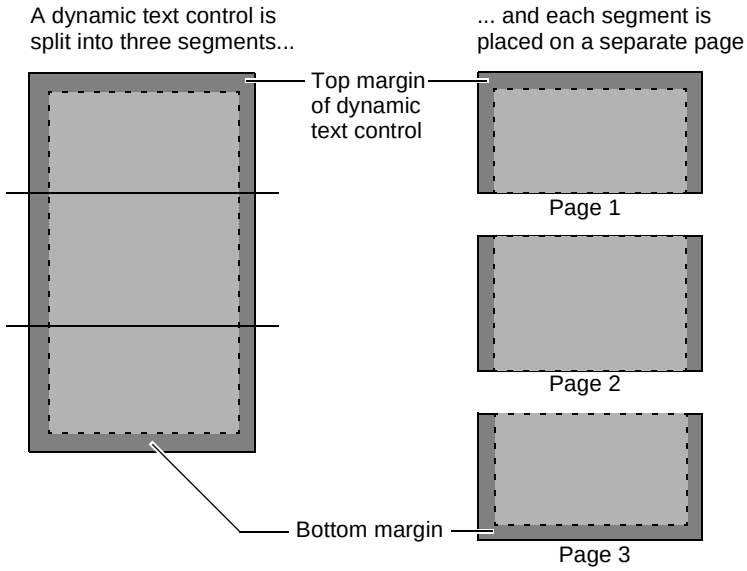


Figure 11-27 Effects on margins using split dynamic text controls

Figure 11-27 shows the effects of the Margin and SplitMarginTop and SplitMarginBottom properties on a dynamic text control that is split.

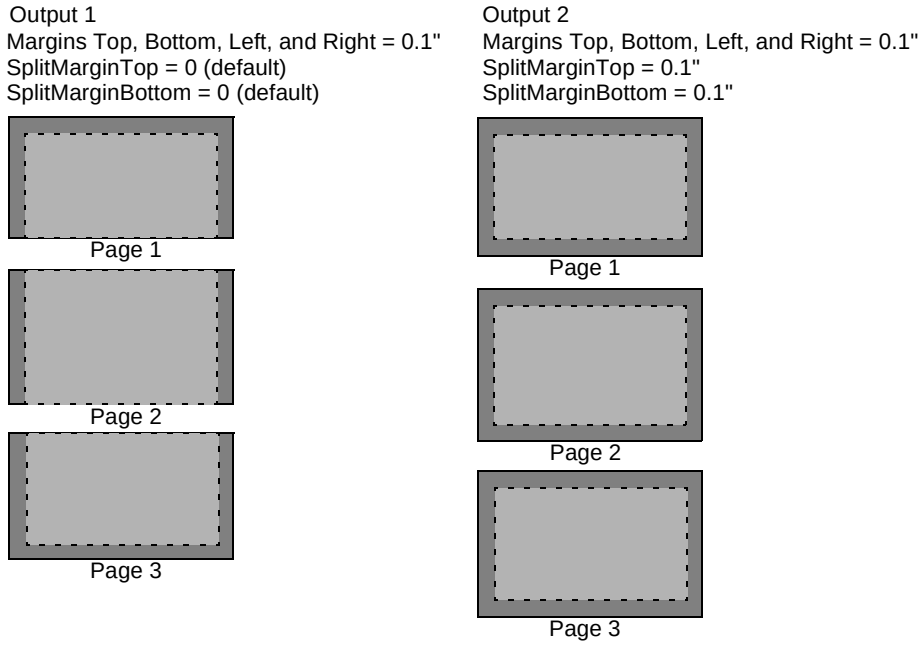


Figure 11-28 Formatting without and with dynamic text control

Setting margins for the segments of a split frame

When a frame and its contents are split over multiple pages, the default behavior is for each content segment to occupy as much of the flow as possible. The frame resizes to fit each content segment. You can, however, add space between the top of each frame segment and the top of its contents, and between the bottom of each frame segment and the bottom if its contents.

To do so, modify the values of the frame's Pagination➤SplitMarginTop and SplitMarginBottom properties. The default setting is zero.

SplitMarginTop specifies the margin between the top edges of all frame segments, except the first, and their contents. The space between the top edges of the first frame segment and the top-most content is equal to the space specified in the report design.

SplitMarginBottom specifies a margin between the bottom edges of all frame segments, except the last, and their contents. The space between the bottom edges of the last frame segment and the bottom-most content is equal to the space specified in the report design.

Figure 11-29 shows the difference between the output using the default values and the output when the frame's SplitMarginBottom and SplitMarginTop properties are set to 0.1".

Controlling how lines in a paragraph break

The default behavior, when e.Report Designer Professional splits a dynamic text control over multiple pages, is to prevent a widow, which is the last line of a paragraph that appears by itself at the top of a page, and an orphan, the first line of a paragraph that appears by itself at the bottom of a page. e.Report Designer Professional prevents widows and orphans by moving lines as needed between pages.

If you do not mind widow and orphan lines in your report, or want to use as much of the available space as possible, change the Pagination➤WidowAndOrphanControl value to False.

Understanding how dynamic text controls affect report layout

Report designs typically contain multiple controls in a frame. Without dynamic text controls, the sizes of frames and controls and their positions relative to one another are set at design time. The layout of content in the generated report reflects the layout in the report design. If you place two controls a quarter inch apart in the report design, the contents of the controls appear a quarter inch apart in the generated report.

Because of their dynamic sizing behavior, dynamic text controls can change the size and positions of other controls in the same frame. If a dynamic text control expands, other controls in the frame move relative to it.

If you plan to use dynamic text controls in your report, it helps to understand how e.Report Designer Professional resizes and repositions controls during report generation. This information helps you decide how and where to place controls, and when to modify the default size and position property values.

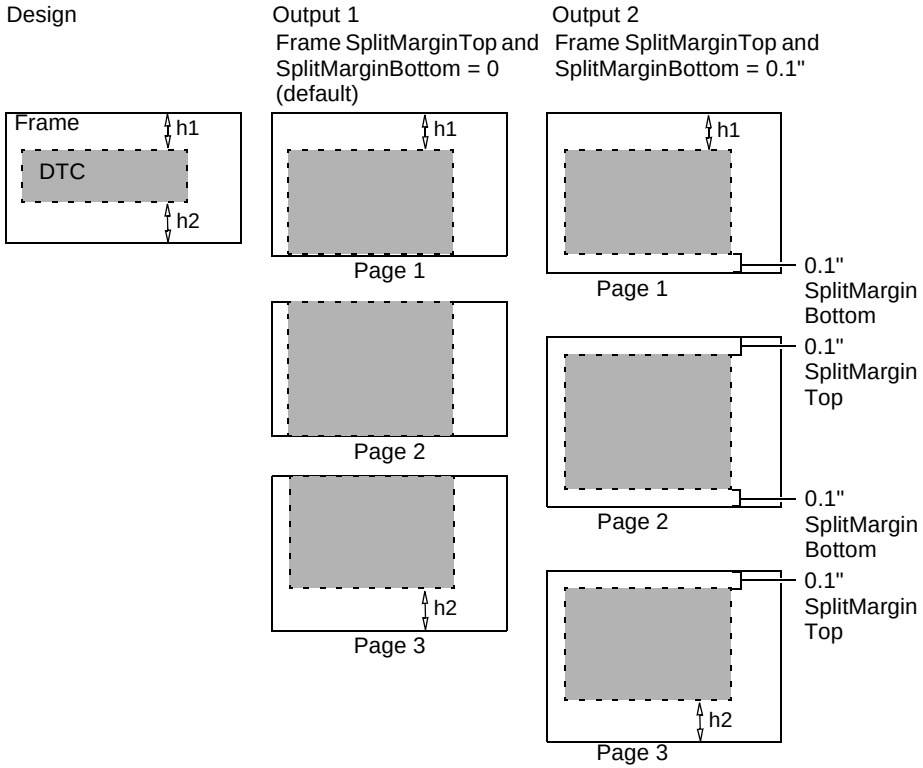


Figure 11-29 Variance between default and custom margin setting

Building a report with dynamic text controls

This section provides step-by-step instructions for building a report that uses regular controls and a dynamic text control. The report organizes orders by customer, and displays only orders that have data in an Issues field. The Issues field stores text data of varying lengths, and this report uses a dynamic text control to display the data.

The report uses data from the Customers and Orders tables in the Sfdata database. Figure 11-30 shows the finished report.

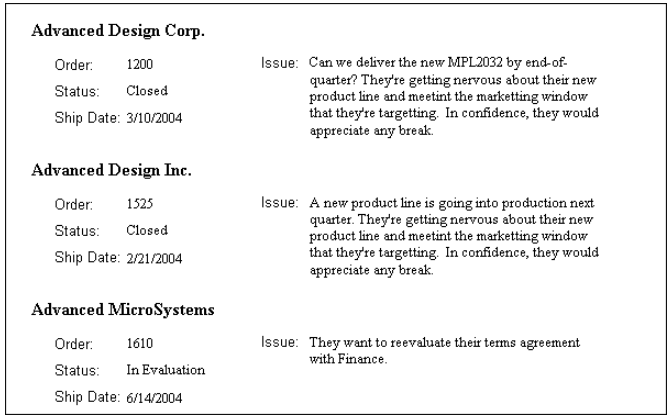


Figure 11-30 Finished report

In this tutorial, you perform the following tasks:

- Create a new report.
- Build the query.
- Group order data by customer.
- Lay out data.
- Run and view the report.
- Change a text control to a dynamic text control.
- Change the relative positions of controls.

How to create a new report

1. Choose File➛New. Create New Report appears.
2. Select Blank Report. Choose OK. A simple report design appears. This default report has a connection to the Sfdata database.
3. Save the report design.

How to build the query

1. Choose View➛Data. Database Login appears.
2. Choose OK. Datasource—Query Editor appears as shown in Figure 11-31.
3. Drag the customers and orders tables from Database Browser to Query Editor.
4. Select Columns and add the following fields to Column Name:
   - customers.customName

- orders.orderID
- orders.status
- orders.shipByDate
- orders.issue

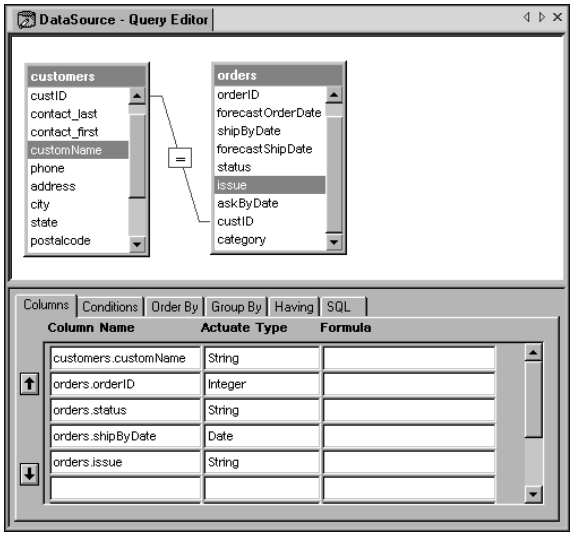


Figure 11-31 Building the query

5 Select Conditions.

6 To specify that the query retrieve only order records that contain data in the Issues field, do the following:

1 Drag the issues column from the orders table and drop it in Column Name.

2 In Query Expression, type the following:

```
!null
```

How to group order data by customer

1 Choose View➛Design. The report design appears.

2 Choose Tools➛Sorting and Grouping. Sorting and Grouping displays the fields available for grouping, as determined by the query.

3 To group data by customer, in Available fields, select customers.customName. Choose the right arrow (>), then choose OK. A group section appears in the Content slot of the report. Within the group section, frames appear in the Before and Content slots.

How to lay out data

1 To display the customer name at the beginning of each group, insert the database field, customers.customName, in the Before—CustomersCustomNameBefore frame.

   1 Choose View➛Fields.

   2 From Fields, drag customers.customName and drop it into the Before—CustomersCustomNameBefore frame.

2 To display order information for every customer, insert the following fields in Content—Frame:

   - orders.orderID
   - orders.status
   - orders.shipByDate
   - orders.issue

3 Choose Insert➛Label to place labels next to each of the fields. For each label, change the text to the following:

   - Order:
   - Status:
   - Ship Date:
   - Issue:

4 Arrange and format the controls in the layout so that they appear as in Figure 11-32.

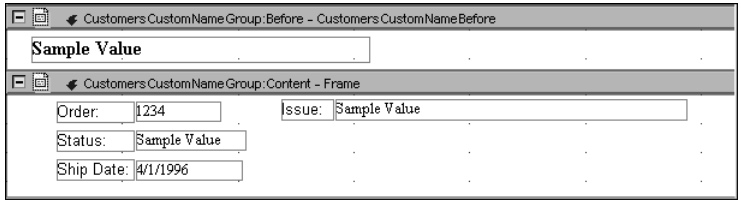


Figure 11-32 Laying out data

5 Save the report design.

How to run and view the report

Choose Report➛Build and Run. The report appears as in Figure 11-33.

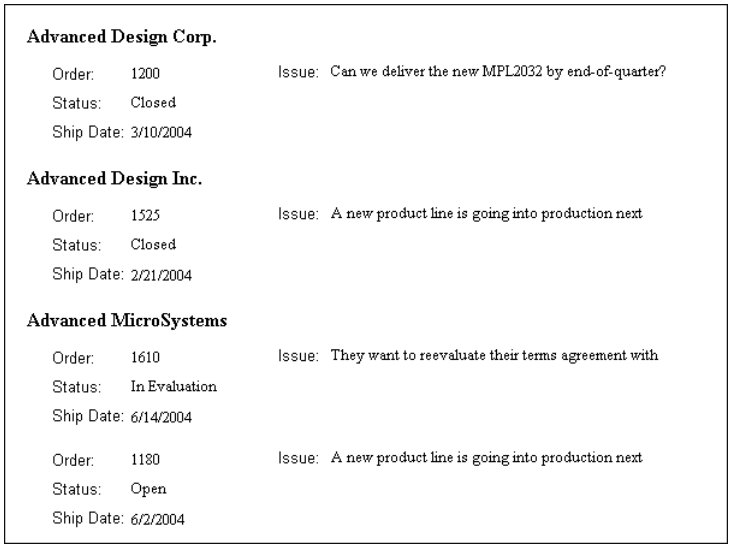


Figure 11-33 Finished report

The Issues data appears on a single line and is truncated because the control that displays the data is a text control. When you use Fields to insert a field, e.Report Designer Professional creates, by default, a text control for string data. To change the text control to a dynamic text control, see the next section.

How to change a text control to a dynamic text control

1. Choose View➛Design to return to the design perspective.
2. In Report Structure, right-click Content—OrdersIssueControl. Choose Properties.
3. In the Properties page for the control, choose Class.
4. In the Class page in Super class, change the control's super class name from AcTextControl to AcDynamicTextControl.
5. Choose Report➛Build and Run. The report appears as shown in Figure 11-34.
6. Observe the output:
   - The Issues data appears in full.
   - The placement of data on the left of the report has changed. There is a gap between the order ID and order status, and the amount of blank space corresponds to the height of the dynamic text control.

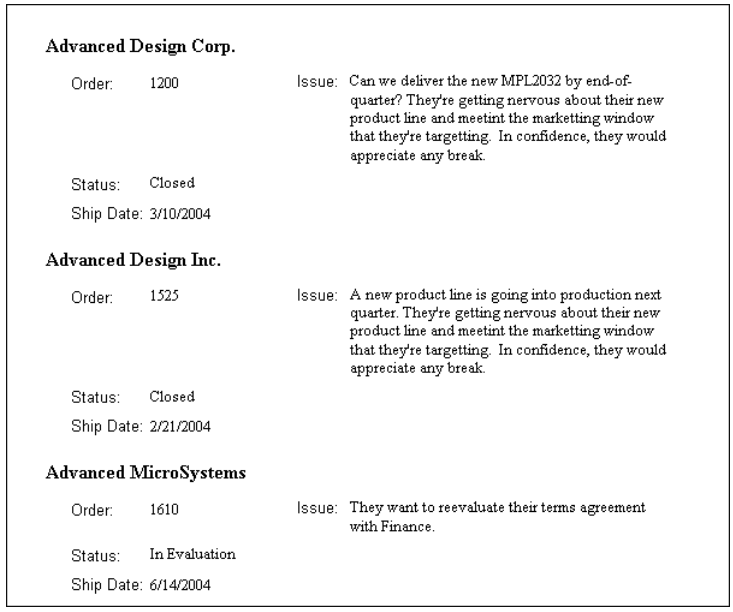


Figure 11-34 Report output modification of text control

How to change the relative positions of controls

The size and position of a dynamic text control affect the positions of controls adjacent to it. By default, any control placed below a dynamic text control moves down when the dynamic text control expands vertically.

In this section, you change the relative positions of controls on the left to eliminate the blank space between the controls.

1. Choose View➛Design.
2. In Report Layout, select all the controls below the dynamic text control, as shown in Figure 11-35.

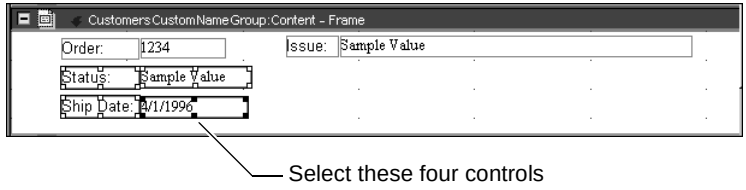


Figure 11-35 Changing the position of controls

3. Choose View➛Properties.
4. In the Properties page for the selected controls, set Dynamic Size and Position➛VerticalPosition to VerticalPositionFrameTop.

This property value specifies that the position of the selected controls at run time are relative to the top of the frame instead of the dynamic text control. Effectively, the controls' vertical positions are fixed. The controls do not move.

5 Choose Report➤Build and Run. The report looks like the one in Figure 11-36.

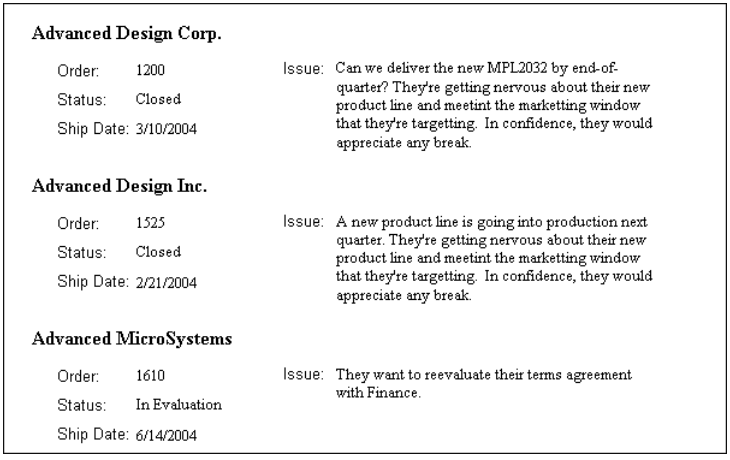


Figure 11-36 Report output modification control position

The layout of data on the left matches the layout in the report design. The controls maintain their positions even as the dynamic text control expand vertically.

Solving common output problems

This section describes a few common output problems when you use dynamic text controls in a report, their probable causes, and ways to solve them.

Problem 1: Only one page is generated

Controls in a frame are truncated at the bottom and only one page is generated, as shown in Figure 11-37.

Solution 1a

Check the Pagination➤AutoSplitVertical properties of the frame and the dynamic text control. If either one of them is set to DoNotSplit, the frame and its contents are truncated if they exceed a page.

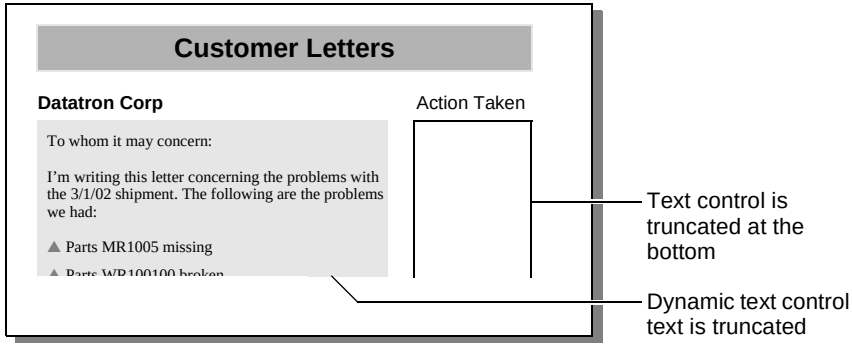


Figure 11-37 One page generated

Solution 1b

If you have multiple controls—dynamic text controls and other controls—in the frame, check the Dynamic Size and Position➤VerticalSize setting of the other controls. If you use VerticalSizeRelative, you must ensure that Pagination➤IsFrameDecoration is set to True. This setting is required to properly display the contents of the controls if they exceed one page. In the example shown in the previous illustration, the text control's VerticalSize is set to VerticalSizeRelative, and its IsFrameDecoration property should be set to True.

Problem 2: Text is truncated

Text in a dynamic text control overflows or is truncated at the end of lines in some rendering environments, as shown in Figure 11-38.

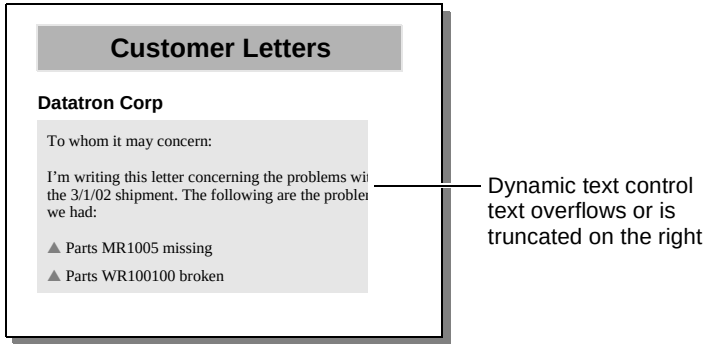


Figure 11-38 Truncated text

Solution 2

Increase the dynamic text control's Text Layout➤LineWidthPadding property value.

Problem 3: Text contains extraneous characters

Text in a dynamic text control contains extraneous characters, as shown in Figure 11-39.

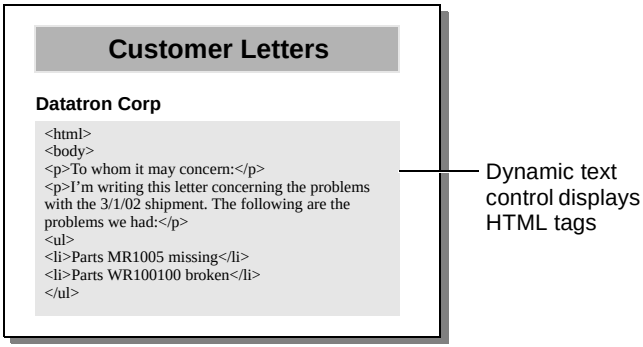


Figure 11-39 Extraneous characters

Solution 3

Check the dynamic text control's TextFormat property value. If set to TextFormatPlain, e.Report Designer Professional interprets HTML or RTF tags as literal text. Set TextFormat to the correct text format.

Chapter

12

Formatting a report with dynamically sized components

This chapter contains the following topics:

- About dynamically sized components
- Formatting controls in a frame that contains a dynamically sized component
- Modifying how pages resize
- Solving common output problems

About dynamically sized components

A dynamically sized component is a component whose size changes at report-generation time, in response to its content. The following components are examples of dynamically sized components:

- A dynamic text control, whose height expands to fit text data.
- A cross-tabulation control, or cross tab. The cross tab's height and width expands to fit rows and columns of data.
- A nested frame, the height and width of which expand to fit dynamically When the size of a dynamically sized component changes, it can cause its container to resize. It can also cause other components within that container to resize or move from their original positions. For example, if a dynamic text control expands, its containing frame expands as well. If you place a line below a cross tab of the same width and the cross tab expands horizontally, the line expands as well.

Use dynamically sized components judiciously. Reports that contain dynamically sized components take longer to generate because e.Report Designer Professional must compute dynamic frame and page layouts.

Formatting controls in a frame that contains a dynamically sized component

You can place any number of controls, in different configurations, in a frame. When a report includes dynamically sized components, e.Report Designer Professional resizes and repositions the controls in an intuitive manner when it generates the report.

e.Report Designer Professional also provides report developers with the flexibility of fine-tuning the resizing and repositioning behavior to achieve different layouts. Controls, frames, flows, and pages contain properties that give you a high level of control over their dynamic sizing and positioning behavior.

If these properties are not visible on the Properties page, choose Expert Properties, as shown in Figure 12-1.

Typically, you can achieve a professional-looking layout by using the default settings or by setting just one or two properties on one or two controls. The following factors determine the appearance of a report that contains dynamically sized components:

- The positions of controls, relative to one another
- The size of controls, relative to one another

- The values of each component's Dynamic Size and Position properties

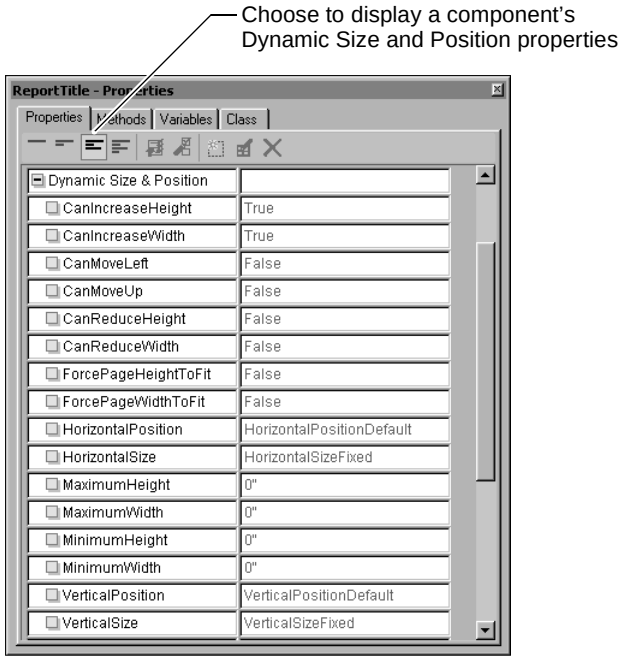


Figure 12-1 Dynamic size and position properties

When you lay out and format controls and frames, you need to decide how and where to place controls relative to cross tabs or dynamic text controls, and what properties to set to achieve the report layout that you want. The rest of this chapter helps you make these decisions by describing how e.Report Designer Professional resizes and repositions controls and frames, and by describing some of the typical changes to the default behavior.

Understanding how Actuate software resizes and repositions components

e.Report Designer Professional uses the following key principles to resize and reposition components:

- A dynamic text control remains the same height or expands vertically. It does not shrink. To modify this behavior, set its CanReduceHeight property to False.
- A cross tab can increase and decrease its size both horizontally and vertically. Its contents determine its size. Changes in the size of a cross tab's container or other content within the container do not affect the cross tab's size. You cannot modify this behavior using the cross tab's properties.

- The container frame remains the same size or expands with dynamically sized components within it. You can set the CanReduceHeight and CanReduceWidth properties to shrink a frame if its contents shrink.
- Pages and flows expand to accommodate a cross tab report, even if you specify a fixed page size. To prevent a page from expanding, set the cross tab's ForcePageHeightToFit and ForcePageWidthToFit properties to False. Using these property values, the cross tab is truncated if it does not fit on the page.
- Depending on the original size and placement of lines and rectangles, these controls expand when an adjacent dynamically sized component expands. To stop a line or rectangle from expanding, set the CanIncreaseHeight and CanIncreaseWidth properties to False.
- Other controls stay the same size unless you specify that they expand relative to the dynamically sized component or frame. To expand a control, modify the following properties:
  - Set VerticalSize to VerticalSizeRelative or VerticalSizeFrameRelative
  - Set HorizontalSize to HorizontalSizeRelative or HorizontalSizeFrameRelative
- A control moves down if a dynamically sized component above it expands vertically.
- A control does not move up if a dynamically sized component above it reduces its height. You can set the control's CanMoveUp property to modify this behavior.
- A control moves to the right if a dynamically sized component on the left of it expands horizontally.
- A control does not move to the left if a dynamically sized component to the left of it reduces its width. You can set the control's CanMoveLeft property to modify this behavior.
- A component's vertical expansion is independent of its horizontal expansion. You can modify the vertical resizing and repositioning behavior separately from the horizontal resizing and repositioning behavior. For example, you can specify a maximum width for a component without specifying a maximum height.

The following sections describe in more detail how e.Report Designer Professional resizes and repositions controls.

Understanding vertical resizing and repositioning

Figure 12-2 shows the layout of controls in the design and the layout in the generated report when a dynamically sized component expands its height. The grey item is a dynamically sized component. B and F are lines. A, C, D, E, G, and H are non-dynamic controls.

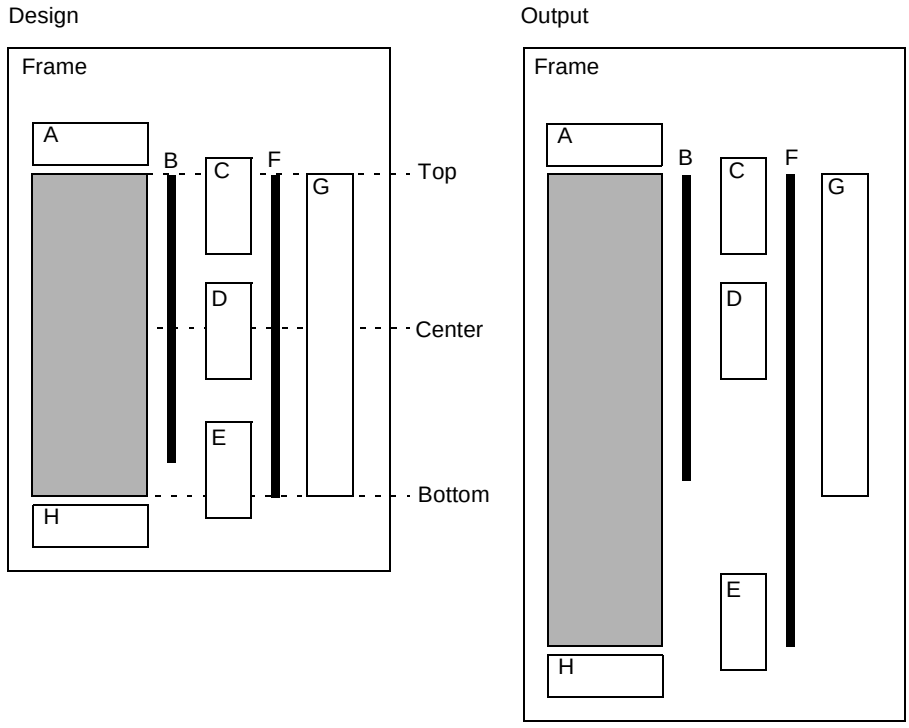


Figure 12-2 Design and generated report layout height comparison

The components in Figure 12-2 exhibit the following behavior:

- The frame expands its height to contain its contents.
- Control A does not expand or move because it is above the dynamically sized component.
- Line B does not expand because its bottom is above the bottom of the dynamically sized component.
- Control C does not move because its top is above the center of the dynamically sized component.
- Control D does not move because its top is above the center of the dynamically sized component.
- Control E moves down because its top is below the center of the dynamically sized component.
- Line F expands because its top and bottom are aligned with the top and bottom of the dynamically sized component, and because its VerticalSize property's default value is VerticalSizeRelative.

- Control G does not expand because its VerticalSize property's default value is VerticalSizeFixed.
- Control H moves down to accommodate changes in the height of the dynamically sized component.

Summary of vertical resizing principles

A control expands vertically if all the following conditions are met:

- Its VerticalSize property is set to VerticalSizeRelative or VerticalSizeFrameRelative.
- Its top is at or above the top of the dynamically sized component.
- Its bottom is at or below the bottom of the dynamically sized component.
- The dynamically sized component expands vertically.

Summary of vertical repositioning principles

A control's vertical movement depends on where its top and bottom align relative to the dynamically sized component.

If the top of the control is above the vertical center of the dynamically sized component, it does not move when the dynamically sized component expands vertically.

If the top of the control is at or below the vertical center of the dynamically sized component, it moves relative to the bottom of the dynamically sized component when the latter expands vertically.

Understanding horizontal resizing and repositioning

Figure 12-3 shows the layout of controls in the design and Figure 12-4 shows the layout in the generated report when the dynamically sized component expands its width. The grey item is a dynamically sized component.

C and G are lines. A, B, D, E, F, and H are non-dynamic controls.

In Figure 12-3 and Figure 12-4, observe the following effects:

- The frame expands its width to contain its contents.
- Control A does not expand or move because it is on the left of the dynamically sized component.
- Control B moves to the right to accommodate changes in the width of the dynamically sized component.
- Line C does not expand because its right side is not aligned with the right side of the dynamically sized component.

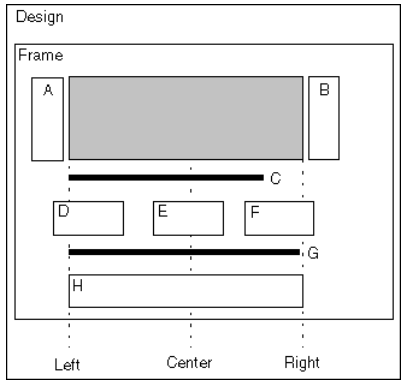


Figure 12-3 Design layout for width comparison

Figure 12-4 Report output for width comparison

- Control D does not move because its left side is to the left of the dynamically sized component's center.
- Control E does not move because its left side is to the left of the dynamically sized component's center.
- Control F moves to the right because its left side is to the right of the dynamically sized component's center.
- Line G expands because its left and right sides are aligned with the left and right sides of the dynamically sized component, and because its HorizontalSize property's default value is HorizontalSizeRelative.
- Control H does not expand because its HorizontalSize property's default value is HorizontalSizeFixed.

As you have probably deduced, horizontal geometry behaves the same as vertical geometry, rotated 90 degrees counterclockwise. The descriptions of vertical resizing and repositioning apply to horizontal resizing and repositioning, except that horizontal replaces vertical, width replaces height, left replaces bottom, and right replaces top.

Summary of horizontal resizing principles

A control expands horizontally if all the following conditions are met:

- Its HorizontalSize property is set to HorizontalSizeRelative or HorizontalSizeFrameRelative.
- Its left side is at or to the left of the dynamically sized component's left side.
- Its right side is at or to the right of the dynamically sized component's right side.
- The dynamically sized component expands horizontally.

Summary of horizontal repositioning principles

A control's horizontal movement depends on where its left and right sides align relative to the dynamically sized component.

If the left side of the control is to the left of the dynamically sized component's horizontal center, it does not move when the dynamically sized component expands horizontally.

If the left side of the control is at, or to the right, of the dynamically sized component's horizontal center, it moves relative to the right of the dynamically sized component when the latter expands horizontally.

Modifying the default resizing and repositioning of controls

You can modify e.Report Designer Professional's default sizing and positioning actions for individual controls. Set values for the Dynamic Size and Position properties for each control that you want to manipulate. Table 12-1 describes the properties that you typically modify.

Table 12-1 Modifying default settings of properties

Property	Description
HorizontalPosition	Specifies how to reposition a control horizontally when a dynamically sized component in the same frame changes in width
HorizontalSize	Specifies how to resize a control's width when a dynamically sized component in the same frame changes in width
VerticalPosition	Specifies how to reposition a control vertically when a dynamically sized component in the same frame changes in height

Table 12-1 Modifying default settings of properties

Property	Description
VerticalSize	Specifies how to resize a control's height when a dynamically sized component in the same frame changes in height

The following sections describe typical tasks when formatting controls in a frame that contains a dynamically sized component.

Resizing a control to match the height of a dynamically sized component

When a dynamically sized component resizes vertically, the default behavior for lines and rectangles is to resize automatically, if their tops and bottoms align with the dynamically sized component's top and bottom. Other controls remain the same size by default. To resize a control relative to the dynamically sized component, set the control's Dynamic Size and Position➤VerticalSize property.

Figure 12-5 shows how a control increases in height to match the height of the dynamically sized component. A is the dynamically sized component.

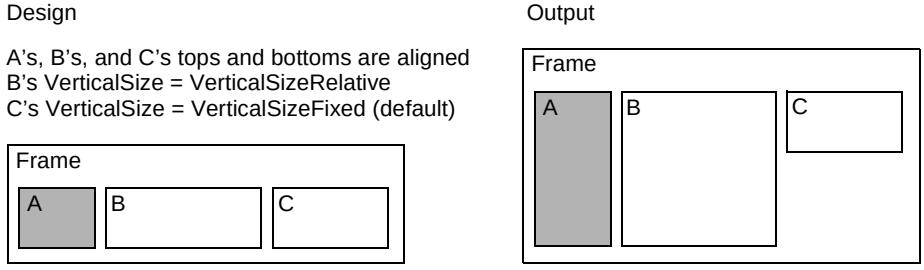


Figure 12-5 Components dynamically sized for height

For a comparison of how e.Report Designer Professional resizes a control when you set its VerticalSize property to each of the different values, see the illustration in "Resizing a control relative to the height of the frame," later in this chapter.

How to resize a control to match the height of a dynamically sized component

1. Align the control as follows:
   - Its top is at or above the top of the dynamically sized component.
   - Its bottom is at or below the bottom of the dynamically sized component.
2. Set the control's Dynamic Size and Position➤VerticalSize property to VerticalSizeRelative.
3. Set the control's Pagination➤IsFrameDecoration property to True. This setting is required to allow the control to split across multiple pages.

Resizing a control to match the width of a dynamically sized component

When a dynamically sized component resizes horizontally, the default behavior for lines and rectangles is to resize automatically, if their left and right sides align with the dynamically sized component's left and right. Other controls remain the same size by default. To modify this behavior, set the control's Dynamic Size and Position➤HorizontalSize property.

Figure 12-6 shows how a control increases in width to match the width of the dynamically sized component. A is the dynamically sized component.

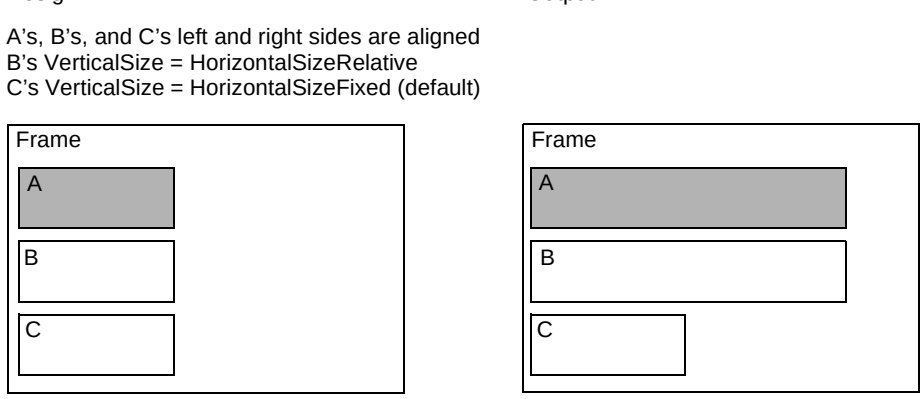


Figure 12-6 Components dynamically sized for width

For a comparison of how e.Report Designer Professional resizes a control when you set its HorizontalSize property to each of the different values, see Figure 12-10.

How to resize a control to match the width of a dynamically sized component

1. Align the control as follows:
   - Its left side aligns with the left side of the dynamically sized component.
   - Its right side aligns with the right side of the dynamically sized component.
2. Set the control's Dynamic Size and Position➤HorizontalSize property to HorizontalSizeRelative.
3. Set the control's Pagination➤IsFrameDecoration property to True. This setting is required to allow the control to split across multiple pages.

Resizing a control relative to the height of the frame

You can specify that a control increase in height relative to the frame's height increase, or relative to a dynamically sized component's height increase. The property to set is Dynamic Size and Position➤VerticalSize.

Figure 12-7 and Figure 12-8 compare the different results when the VerticalSize property is set to each of the three values. A, B, and C are dynamically sized components. D is a non-dynamic control.

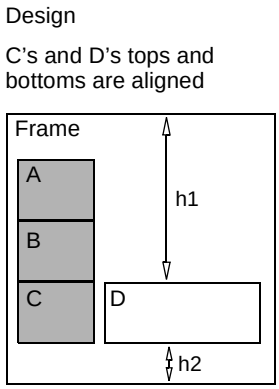


Figure 12-7 VerticalSize with three values and non-dynamic control in a design

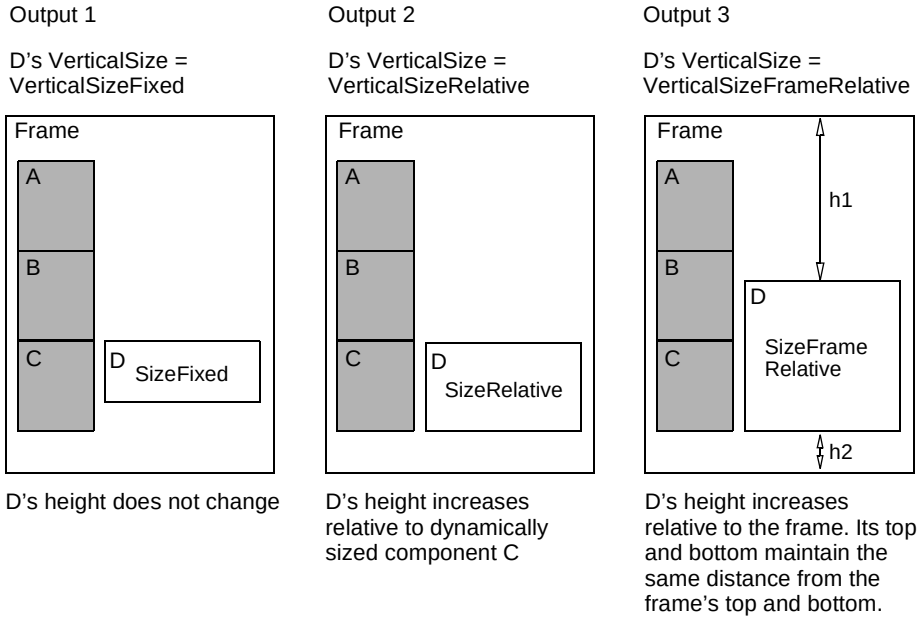


Figure 12-8 VerticalSize with three values and non-dynamic control in the output

How to resize a control relative to the height of the frame

1 Set the control's Dynamic Size and Position➤VerticalSize property to VerticalSizeFrameRelative.

If the frame's height increases, the control's height increases by the same amount. The distance between the top of the control and the top of the frame is constant, as is the distance between the bottom of the control and the bottom of the frame.

2 Set the control's Pagination➛IsFrameDecoration property to True. This setting is required to allow the control to split across multiple pages.

Resizing a control relative to the width of the frame

You can specify that a control's width increases relative to the frame's width increase, or relative to a dynamically sized component's width increase. The property to set is Dynamic Size and Position➛HorizontalSize.

Figure 12-9 and Figure 12-10 compare the different results when the HorizontalSize property is set to each of the three values. B and C are dynamically sized components. A and D are controls.

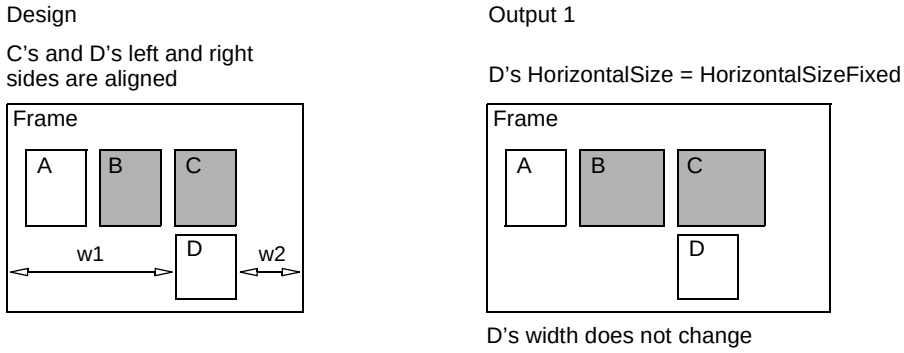


Figure 12-9 HorizontalSize with three values and two controls

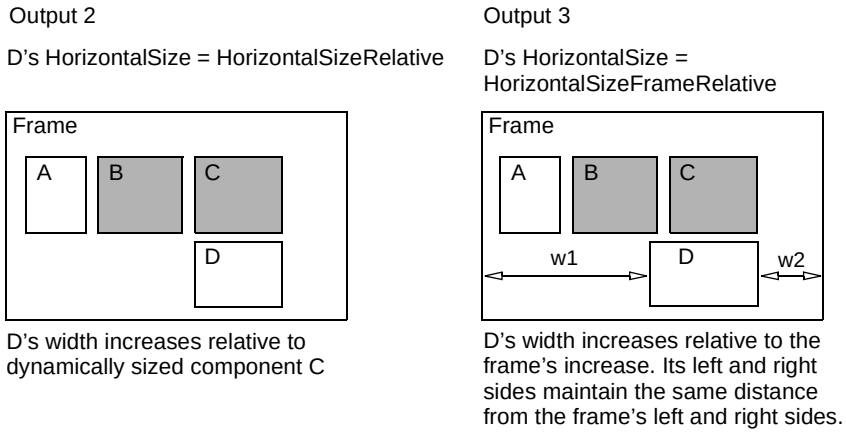


Figure 12-10 HorizontalSize with three values and two controls (continued)

How to resize a control relative to the width of the frame

1 Set the control's Dynamic Size and Position➛HorizontalSize property to HorizontalSizeFrameRelative.

   If the frame's width increases, the control's width increases by the same amount. The distance between the left of the control and the left of the frame is constant, as is the distance between the right of the control and the right of the frame.

2 Set the control's Pagination➛IsFrameDecoration property to True. This setting is required to allow the control to split across multiple pages.

Repositioning a control vertically relative to the frame

The default setting for a control's vertical position is to adjust it relative to the position of a dynamically sized component. You can, however, reposition a control vertically relative to the frame instead. The property to set is Dynamic Size and Position➛VerticalPosition.

Figure 12-11, Figure 12-12, and Figure 12-13 compare the different results when the VerticalPosition property is set to each of the six values.

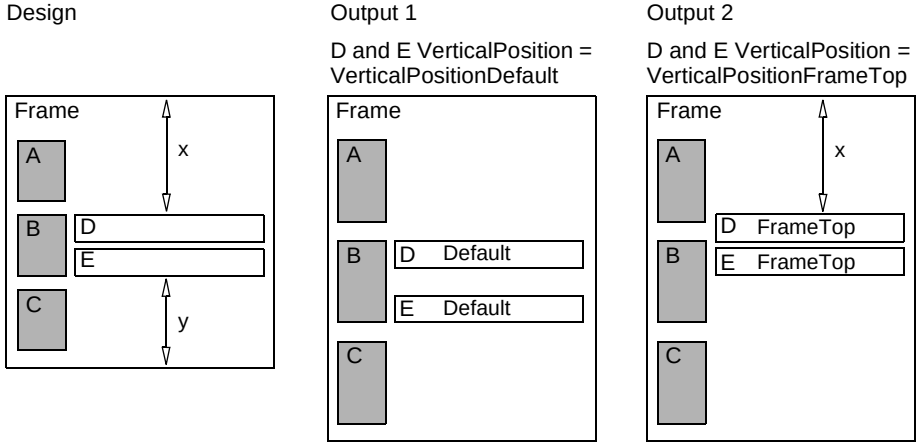


Figure 12-11 Vertical Position property set to six possible values

How to reposition a control vertically relative to the frame

Set the control's Dynamic Size and Position➛VerticalPosition property to one of the following values:

- VerticalPositionFrameTop
  The control's vertical position is fixed. The control does not move.

- VerticalPositionFrameBottom

  The control moves vertically to keep the distance between its bottom and the bottom of the frame constant.

- VerticalPositionFrameMiddle

  The control moves vertically to keep the distance between its middle and the middle of the frame constant.

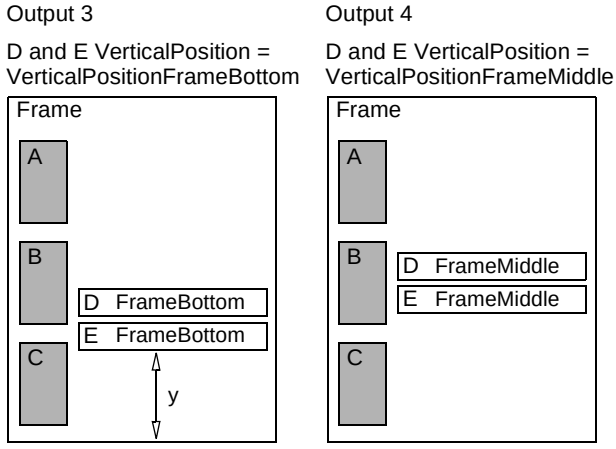


Figure 12-12 Vertical Position property set to six possible values (continued)

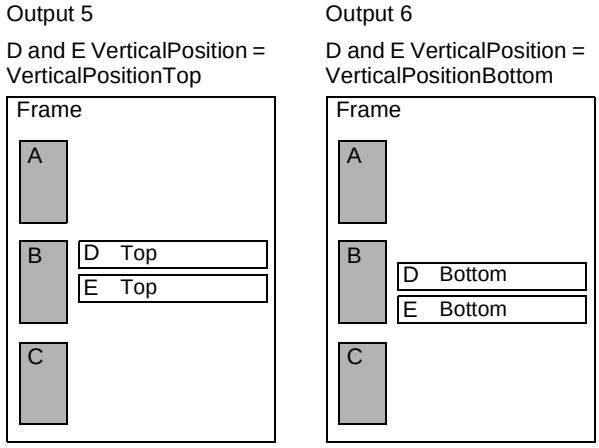


Figure 12-13 Vertical Position property set to six possible values (continued)

Repositioning a control horizontally relative to the frame

The default setting for a control's horizontal position is to adjust it relative to the position of a dynamically sized component. You can, however, reposition a

control horizontally relative to the frame instead. The property to set is Dynamic Size and Position➛HorizontalPosition.

Figure 12-14, Figure 12-15, Figure 12-16, and Figure 12-17 compare the design layout and the different results when the HorizontalPosition property is set to each of the six values.

How to reposition a control horizontally relative to the frame

Set the control's Dynamic Size and Position➛HorizontalPosition property to one of the following values:

- HorizontalPositionFrameLeft
  The control's horizontal position is fixed. The control does not move.
- HorizontalPositionFrameRight
  The control moves horizontally to keep the distance between its right side and the right side of the frame constant.
- HorizontalPositionFrameCenter
  The control moves horizontally to keep the distance between its middle and the middle of the frame constant.

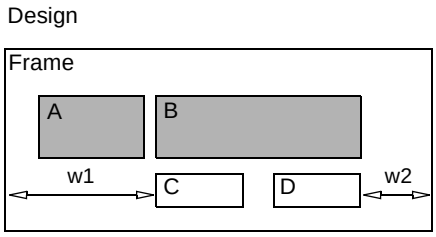


Figure 12-14 HorizontalPosition property six possible settings

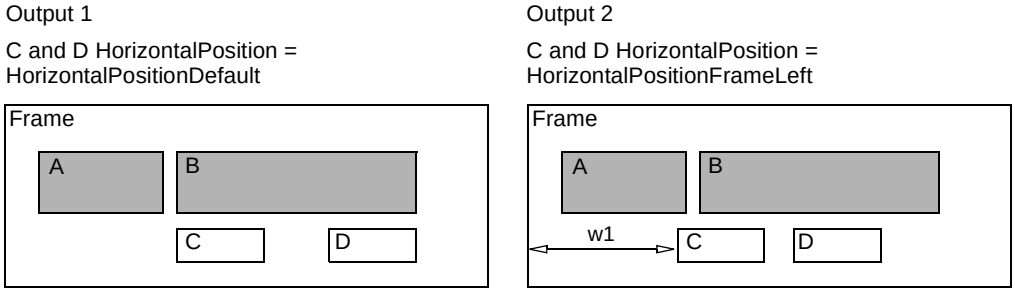


Figure 12-15 HorizontalPosition property six possible settings (continued)

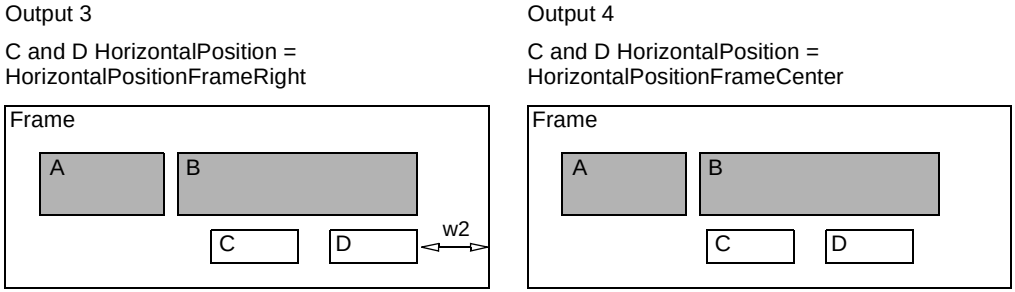


Figure 12-16 HorizontalPosition property six possible settings (continued)

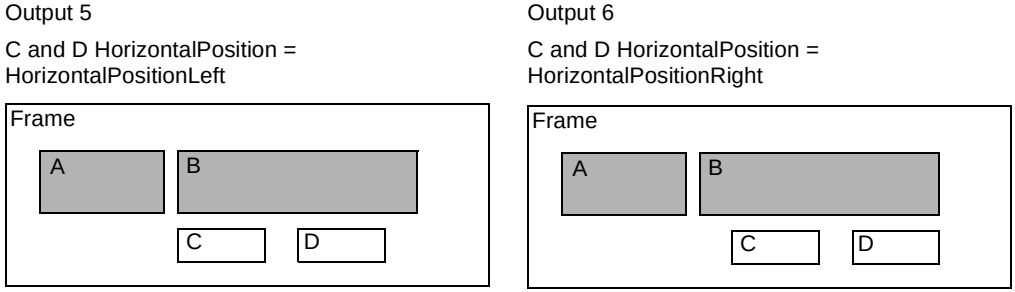


Figure 12-17 HorizontalPosition property six possible settings (continued)

Modifying how pages resize

The following factors control the sizes of pages in the generated report:

- Whether you specify a fixed or dynamic page size. If you specify a dynamic page size, the page expands to accommodate the report contents, and the entire report appears on one page. In this case, dynamically sized components obviously have no effect on individual pages.
- The type of dynamically sized component that you place in the report.
  - A cross tab causes a page to expand to fit it, even if you specify a fixed page size. To modify this behavior, see "Enforcing a fixed page size," later in this chapter.
  - If you specify a fixed page size, a dynamic text control's contents splits over multiple pages, if necessary. Unlike the cross tab, it does not cause the page to expand. To modify this behavior, see "Resizing a page to fit the contents of a dynamic text control," later in this chapter.

If a report design uses a fixed page size and it contains a cross tab and dynamic text control, the generated report can contain pages of different sizes.

Enforcing a fixed page size

The default behavior for a page is to expand to fit a cross tab, even if you specify a fixed page size. This behavior ensures that a cross tab appears in full. This behavior also can cause the page displaying the cross tab to be larger than other pages. If you want all your report pages to be a uniform size, you can enforce the specified page size by setting the following properties:

- Set the cross tab's Dynamic Size and Position➤ForcePageHeightToFit and ForcePageWidthToFit properties to False.
- Set the page's Dynamic Size and Position➤CanIncreaseHeight and CanIncreaseWidth properties to False, the default setting.

These settings can result in the cross tab report being truncated if it expands beyond the page size. Figure 12-18 and Figure 12-19 show how report output differs when the cross tab's ForcePageWidthToFit property is set to True the default setting, and False.

Cross tab ForcePageWidthToFit = True
The page that displays the cross tab is larger in width than the second page because the cross tab forced it to expand

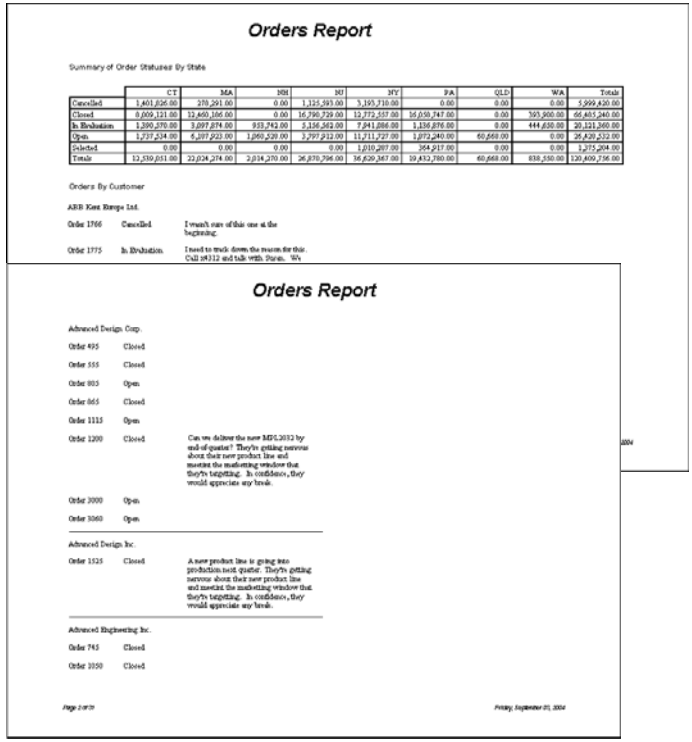


Figure 12-18 True (default) option of ForcePageWidthtoFit property

Cross tab ForcePageWidthToFit = False

The page does not expand to accommodate the cross tab

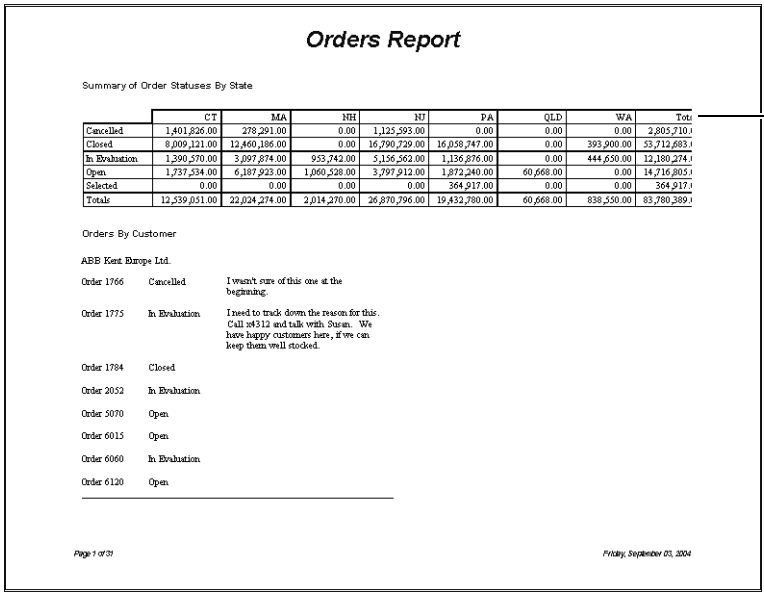


Figure 12-19 False option of ForcePageWidthtoFit property

Resizing a page to fit the contents of a dynamic text control

By default, if you specify a fixed page size, the content of a dynamic text control appears on multiple pages if it is too large to fit on one page. To specify that the page expand to fit all the contents of the dynamic text control, set the dynamic text control's Dynamic Size and Position➛ForcePageHeightToFit property to True. You do not need to set the ForcePageWidthToFit property because the content of a dynamic text control expands vertically only.

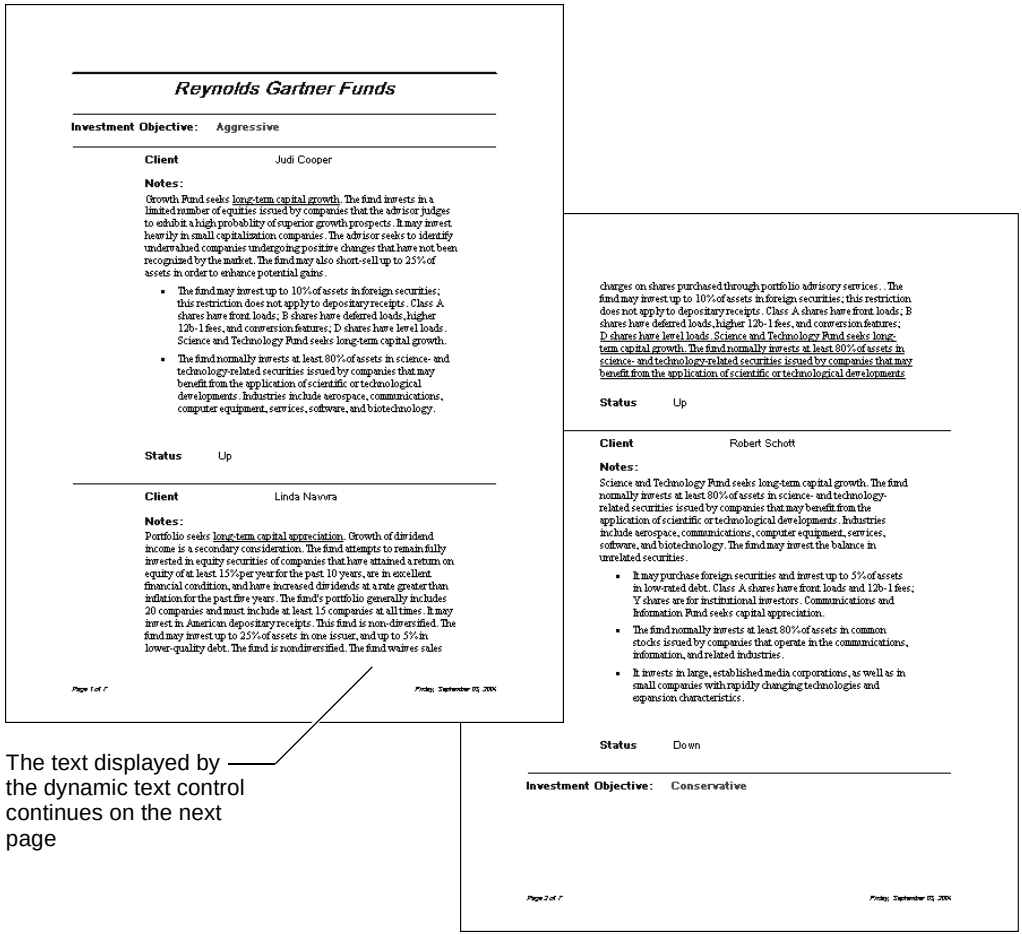
By resizing the page to fit all the content, you improve the report's readability online. The user can view all pertinent information on a single page. Naturally, this action also results in pages of different sizes.

Figure 12-20 and Figure 12-21 show how report output differs when the dynamic text control's ForcePageHeightToFit property is set to False the default value, and True.

If you set a component's ForcePageHeightToFit or ForcePageHeightToFit properties to True, these settings override the page's CanIncreaseHeight and CanIncreaseWidth property values of False.

Dynamic text control ForcePageWidthToFit = False (default)

The page does not expand to accommodate the contents of the dynamic text control



The text displayed by the dynamic text control continues on the next page

Figure 12-20 False (default) option of ForcePageHeighttoFit property

Dynamic text control ForcePageWidthToFit = True

The page expands to accommodate the contents of the dynamic text control

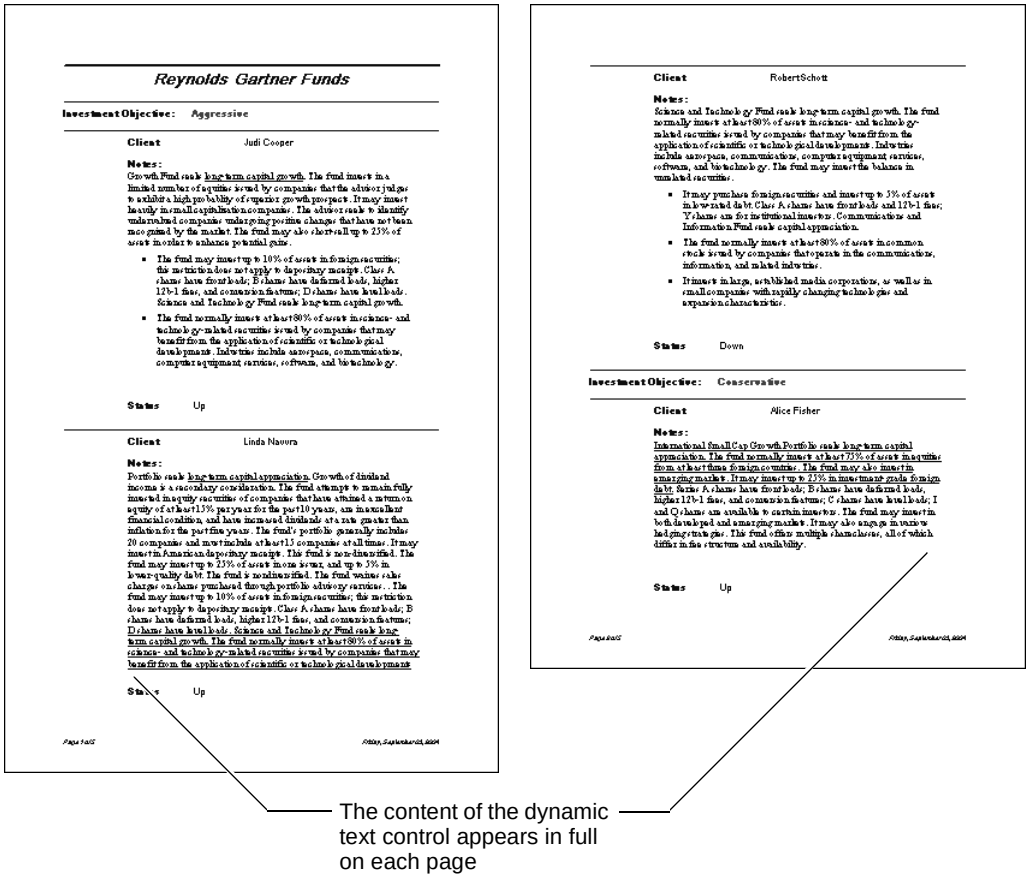


Figure 12-21 True option of ForcePageHeighttoFit property

Solving common output problems

This section describes a few common output problems when you use dynamically sized components in a report, their probable causes, and ways to solve them.

Problem 1: Overlapping controls

A frame displays overlapping controls, as shown in Figure 12-22. A is a dynamically sized component, B is another control.

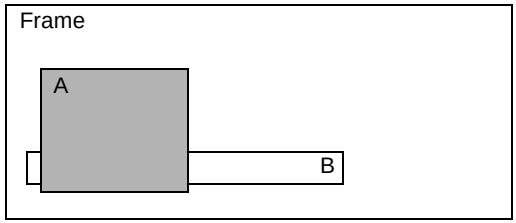


Figure 12-22 Overlapping controls

Solution 1

When a dynamically sized component increases in height and overlaps another control, it usually means that the other control's vertical position is set relative to the frame instead of the dynamically sized component. In this example, C's VerticalPosition is set to VerticalPositionFrameBottom, which means its bottom maintains the same distance from the bottom of the frame. If you do not know if, or how much, a dynamic text control above a control will increase in height, set the control's VerticalPosition property to VerticalPositionDefault, the default value. When the dynamic text control increases in height, the control moves down accordingly.

Problem 2: Truncated controls in a nested frame

A frame displays truncated controls in a nested frame, as shown in Figure 12-23. A and B are dynamically sized components, C is another control, Frame 2 is a nested frame.

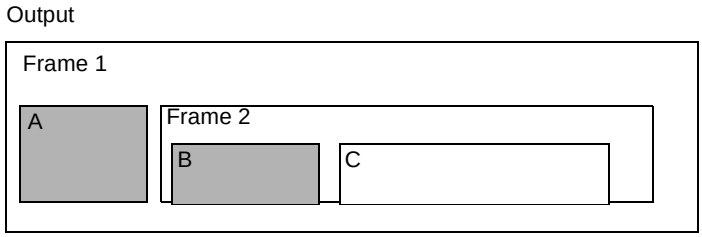


Figure 12-23 Truncated controls in a nested frame

Figure 12-24 shows the report design and property values that resulted in the output above.

Design

Frame 2 VerticalSize = VerticalSizeFrameRelative
Control C VerticalSize = VerticalSizeRelative

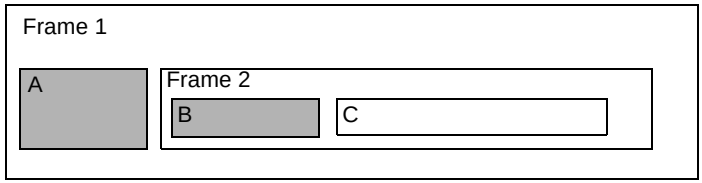


Figure 12-24 Property values for truncated controls

Solution 2

When controls in a nested frame are truncated, this typically indicates that a dynamically sized component increased in height but its containing frame did not increase enough. Check the frame's VerticalSize property value. In this example, Frame 2's VerticalSize property is set to VerticalSizeFrameRelative. This setting causes Frame 2 to increase by the same amount as Frame 1, rather than increasing to accommodate its contents. Frame 1, in turn, increases to accommodate control A. As a result, Frame 2 might not expand sufficiently to accommodate control B's data. When control B is truncated, control C is also truncated because its size expands to match B's size.

To fix the problem in this example, set Frame 2's VerticalSize property to VerticalSizeFixed. This setting enables Frame 2 to expand to accommodate its contents.

Problem 3: Truncated cross-tab report

A report displays a cross tab with data truncated on the right, as shown in Figure 12-25.

	CT	MA	NH	NJ	PA	QLD	WA	Tot
Cancelled	1,401,826.00	278,291.00	0.00	1,125,593.00	0.00	0.00	0.00	2,805,710.0
Closed	8,009,121.00	12,460,186.00	0.00	16,790,729.00	16,058,747.00	0.00	393,900.00	53,712,683.0
In Evaluation	1,390,570.00	3,097,874.00	953,742.00	5,156,562.00	1,136,876.00	0.00	444,650.00	12,180,274.0
Open	1,737,534.00	6,187,923.00	1,060,528.00	3,797,912.00	1,872,240.00	60,668.00	0.00	14,716,805.0
Selected	0.00	0.00	0.00	0.00	364,917.00	0.00	0.00	364,917.0
Totals	12,539,051.00	22,024,274.00	2,014,270.00	26,870,796.00	19,432,780.00	60,668.00	838,550.00	83,780,389.0

Figure 12-25 Truncated cross-tab report

Solution 3

- Check the cross tab's ForcePageWidthToFit property value. It must be set to True to enable a page to expand horizontally to fit the cross tab.
- Check the CanIncreaseWidth property of the frame that contains the cross tab. It must be set to True to enable the frame to expand horizontally to fit the cross tab.

If the cross tab is truncated at the bottom, set the cross tab's ForcePageHeightToFit property to True. At the same time, set to True the CanIncreaseHeight property of the frame that contains the cross tab.

Chapter

13

Building subreports and conditional sections

This chapter contains the following topics:

- About building a report that contains subreports
- Setting up the structure of the main report
- Creating a page layout for a subreport
- Building a report with side-by-side subreports
- Building a report with sequential subreports
- Building a report with linked subreports
- Building a report with conditional sections
- Examining a report containing multiple sections and subreports

About building a report that contains subreports

You can build a report that contains multiple reports, or subreports, and you can lay out the reports in a variety of configurations. Within a single report, you can, for example, do the following:

- Display multiple reports, one after another. For example, display top ten customers, top ten sales representatives, and top ten products.
- Display multiple reports, side by side. For example, display employee information and salary history.
- Display one report within another. For example, display detailed customer notes within general customer information.
- Display different reports, or parts of a report, based on a condition. For example, display different sections for permanent or contract employees.
- Combine any of the report configurations listed above. For example, build a single-page sales summary report that displays a chart of planned, forecast, and actual sales in the upper left, top deals in the upper right, sales with pending issues in the lower left, cancelled sales in the lower right, and grand totals above each summary report.

A subreport is simply a report that appears inside another report. You already know how to create a report. Basically, you have one additional step to create a report that contains subreports—set up the structure of the main report to include and organize subreports.

After you set up the structure, you create, lay out, and test each subreport, one at a time, as you would any other report. Each subreport can access its own set of tables and fields, have its own data selection criteria, and its own page layout. Subreports can be linked to one another, or be independent of each other. Two reports are linked when the data of one report controls the data that appears in another.

Figure 13-1 shows the first page of an example report, Forecast, that ships with the product. This report contains multiple subreports.

For information about how the Forecast report is designed, see "Examining a report containing multiple sections and subreports," later in this chapter.

Setting up the structure of the main report

To build a report with subreports, you begin by creating a report structure that specifies the order in which subreports appear and their placement relative to other subreports. e.Report Designer Professional provides several structure components for organizing subreports, described in the next section.

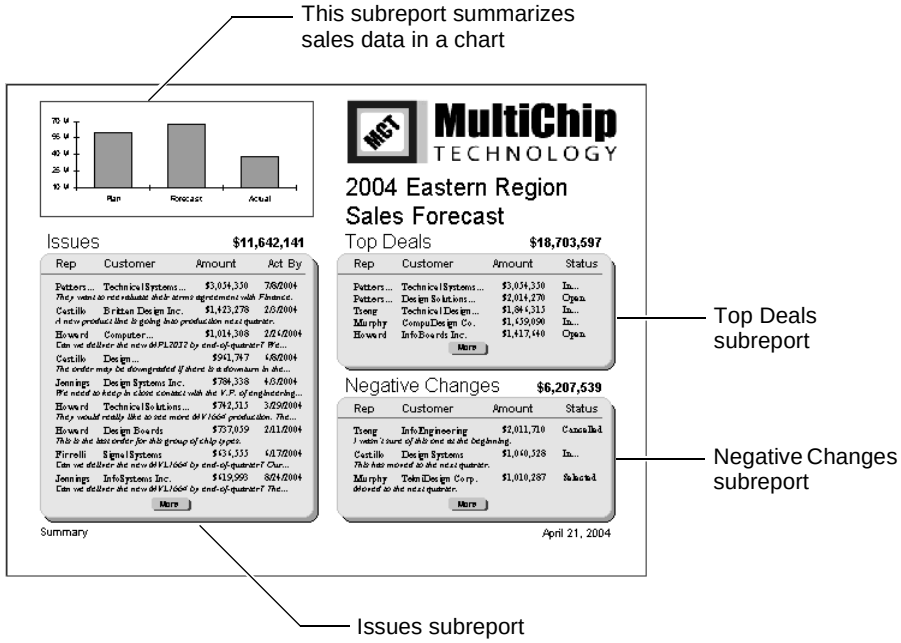


Figure 13-1 Report containing multiple subreports

There is no limit to the number of subreports or levels of nested subreports that you can create. Be aware, however, that each subreport or nesting level that you create adds complexity to the report. Report generation takes longer, debugging the report can be more difficult, and viewing and working with the different reports and parts of each report can become unmanageable.

Plan your report and examine your options carefully. If your report contains too many subreports, maybe you are presenting too much information in the report. Consider creating several separate reports instead.

About the components that define a report's structure

Table 13-1 describes the components that you use to create the logical structure of a report. These components, called section components, act as containers for other components. They appear as icons in the report structure, but do not appear in the layout because they are not visible in the report. The only visible sign that a section component exists is the effect it has on the organization of visual components, such as frames and controls.

Table 13-1 Report structure components

	Section	Description	Example
	Report section	Use to start a new report, or a subreport within a main report. This section component is the only one that can contain a data stream to retrieve information from a data source.	Design a report or subreport.
	Sequential section	Use to contain two or more reports. The reports appear one after another in a specified order.	Display two reports, such as a sales history and a staffing history, one after the other.
	Parallel section	Use to contain two or more reports. The reports appear side by side on the same page.	Present two related reports, such as employee addresses and salary histories, displayed side by side for easy comparison.
	Conditional section	Use to contain two reports or frames. The report or frame that appears depends on a condition that you specify.	Display different frames for salaried or commission employees.
	Group section	Use to sort data on a common field, such as customers grouped by state.	Design a grouped report to display orders for customers in various states.

Placing section components

You can place the different types of section components in many different slots in a report design. For example, you can place a sequential section in any of the following slots:

- The Content slot of a top-level report component or another sequential section
- The Then or Else slots of a conditional section
- The Before, Content, or After slots of a report section or a group section

This flexibility enables you to create reports with simple to complex layouts, and with one to multiple levels of nested subreports.

For most reports, you place a section component in the Content slot of the top-level report component or the Content slot of another section. You do not have to memorize the rules that govern where you place components. e.Report Designer Professional helps you create a logical report structure by preventing you from placing components in the wrong slots. For example, e.Report Designer

Professional stops you from placing a section component in a PageHeader or PageFooter slot.

The following examples show some simple report structures to illustrate the basic concepts of using each type of section component.

Examining a report containing three sequential subreports

The report structure shown in Figure 13-2 contains three subreports that appear one after another in the generated report. For clarity, empty slots do not appear.

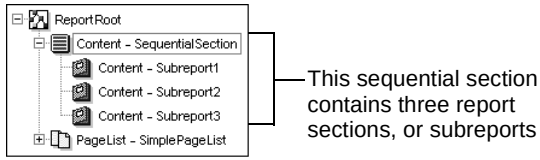


Figure 13-2 Report containing three sequential subreports

The sequential section indicates the start of a series of subreports. The three report sections inside the sequential section represent the three subreports. The order in which the report sections appear in the report structure is the order in which the subreports appear in the finished report.

Examining a report containing two side-by-side subreports

The report structure shown in Figure 13-3 contains two subreports that appear side by side in the generated report. For clarity, empty slots do not appear.

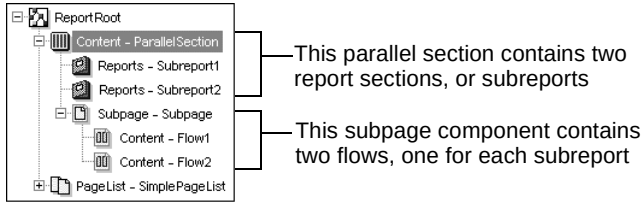


Figure 13-3 Report containing two side-by-side subreports

The parallel section indicates the start of a series of subreports that appear side by side. The two report sections inside the parallel section represent the two subreports. In the finished report, Subreport1 appears on the left side of the page and Subreport2, on the right side. To display these two subreports side by side, you must create a page or subpage that has two flows. A flow defines the printable area on a page or subpage.

Examining a report containing two conditional subreports

The report structure shown in Figure 13-4 contains two conditional subreports.

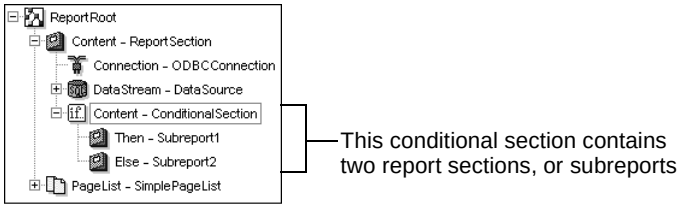


Figure 13-4 Report containing two conditional subreports

The two report sections inside the conditional section represent the two subreports. Based on a condition that you specify, one of the subreports appears in the finished report.

Placing components in a report section

After you place a report section in a report design, you can place the following components in that report section:

- A connection in the Connection slot. A simple report that contains no subreports requires a connection component. If the report contains multiple report sections, you can use one connection that other report sections share.
- A data source in the DataStream slot. If you do not place and define a data source, the report has no access to any data so the report is empty.
- A frame in the Before, After, PageHeader, or PageFooter slot.
- A section or frame in the Content slot.
- An optional subpage in the Subpage slot.

Placing components in a sequential section

After you place a sequential section in a report design, you can place the following components in that sequential section:

- An optional connection in the Connection slot.
- Multiple sections or frames in the Content slot. The order in which these components appear in the report structure is the order in which they appear in the generated report.
- An optional subpage in the Subpage slot.

Placing components in a parallel section

After you place a parallel section in a report design, you can place the following components in that parallel section:

- An optional connection in the Connection slot.

- Multiple report sections in the Reports slot. You can place only report sections within a parallel section.
- An optional subpage in the Subpage slot.

Placing components in a conditional section

After you place a conditional section in a report design, you can place the following components in that conditional section:

- An optional connection in the Connection slot.
- A section or frame in the Then slot. If the condition that you specify evaluates to True, the section or frame in this slot appears.
- A section or frame in the Else slot. If the condition that you specify evaluates to False, the section or frame in this slot appears.
- An optional subpage in the Subpage slot.

Specifying the condition for a conditional section

When you use a conditional section in a report design, you must specify the conditional expression that determines which section or frame to display when a user runs the report. To specify the conditional expression, set the conditional section's IfExp property to an expression that evaluates to True or False.

Depending on where you place the conditional section, the expression can refer to data row columns, parameters, or global variables, as shown in the following examples:

- The following expression uses a column name:

  ```
  [offices.officeID] = "1"
  ```

  Basically, the expression says, "If office ID is equal to 1, then display the section or frame in the Then slot, otherwise display the section or frame in the Else slot."

- The following expression uses an ad hoc parameter, which you define in Query Editor:

  ```
  DataSource::offices_officeID = "1"
  ```

  Basically, the expression says, "At run time, if the user enters 1 for the offices_officeID parameter, display the section or frame in the Then slot, otherwise display the section or frame in the Else slot." DataSource::offices_officeID is the fully-scoped parameter name, where DataSource is the name of the data stream component and offices_officeID is the parameter name.

How to specify the condition for a conditional section

1. Right-click the conditional section for which you want to specify a condition, then choose Properties.
2. In the Properties page for the conditional section, set the IfExp property to an expression that evaluates to True or False, as shown in Figure 13-5.

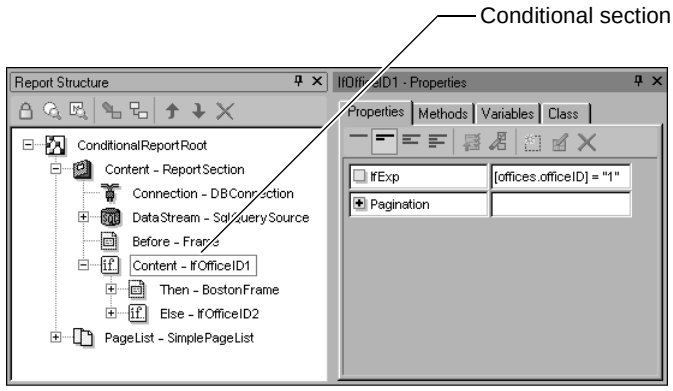


Figure 13-5 Conditional section of the Properties page

Using an aggregate function in a conditional section or sequential section

Actuate e.Report Designer Professional does not support using an aggregate function in a conditional section or a sequential section. If you use an aggregate function in a conditional section or a sequential section, the aggregate function retrieves only the data value from the first data row. You can use a lookahead aggregate in a control that is in a sequential section or in a conditional section. For more information about adding aggregate information to a report design, see "Adding aggregate information," in Chapter 4, "Laying out a report."

Creating a page layout for a subreport

When you create a report that contains multiple subreports or sections, you can create a different page layout for each section. You can, for example, use a different display size, background color, or border for different sections of a report. If you do not create a special page layout for a section, e.Report Designer Professional uses the main report's page layout by default.

The only case where you need to customize a page layout is if your report design contains a parallel section to display multiple subreports side by side. Typically, the page layout for a report design displays report data in a single-column flow. Parallel subreports, however, require multicolumn flows. To display the contents

of parallel subreports successfully, you must create a page layout that has enough flows to display all the subreports.

You create a page layout for subreports in one of the following ways:

- Create a page layout specific to the section's Subpage slot.
- Modify the main report's default page layout.

The option that you choose depends on whether or not your report design needs to support multiple page layouts. You can create, for example, one layout for a parallel section and one for the rest of the report. If your report design contains only one section, modify the main report's default page layout.

Figure 13-6 shows a report structure that contains two page layouts, one for the main report and the other for a parallel section. The page layout for the parallel section uses two flows. The page layout for the main report uses a single flow.

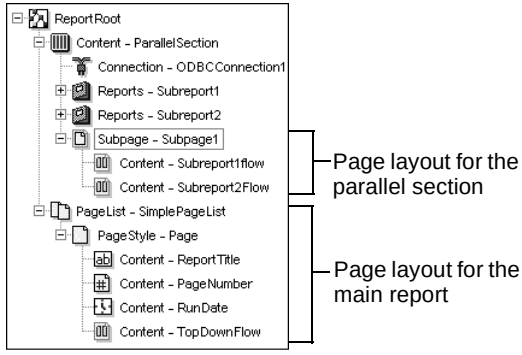


Figure 13-6 Multiple page layouts

Using a subpage

Every section component has a Subpage slot in which you can optionally place a subpage that specifies a page layout for that particular section. A subpage is just like a page, except that it fits into a flow on the main report's page. This fact is important to remember when you design a subpage because the subpage must fit inside the flow of the main report's page.

For example, if the size of the main report's page flow is 7 inches by 10 inches, the size of the subpage must be 7 inches by 10 inches, or smaller. If the subpage is larger, you get an error when you run the report.

How to create a page layout using a subpage

1. Place a subpage component in the Subpage slot of the section.
2. Make sure the size of the subpage is smaller than, or equal to, the size of the flow in the main report's page.

3 Place a flow component in the Content slot of the Subpage component. For a parallel section, place as many flow components as there are subreports.

4 Customize the page layout by setting properties of the subpage and flow components. You can also add decorative elements, such as lines, text labels, or images onto a subpage.

Creating a multiflow page layout for parallel subreports

To display the contents of parallel subreports, you must create a page layout that has enough flows to display all the subreports, then you must specify the flow in which each subreport appears.

How to create a multiflow page layout for parallel subreports

1 Place a subpage component in the Subpage slot of the parallel section, as shown in Figure 13-7.

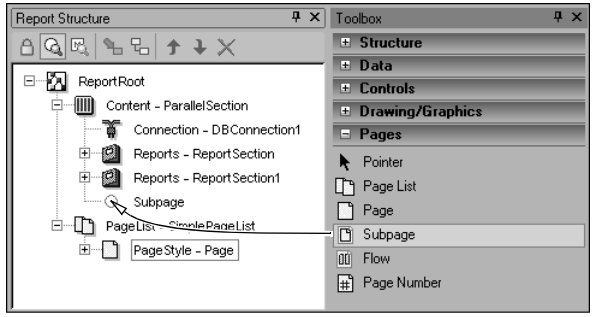


Figure 13-7 Adding a subpage component to the page layout

The subpage appears in the report structure.

2 Place a flow component in the subpage, as shown in Figure 13-8.

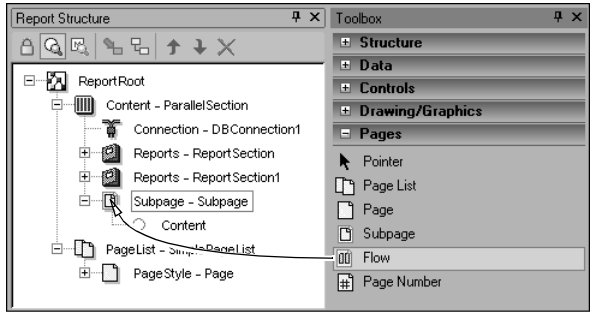


Figure 13-8 Adding a flow component to the page layout

3 Repeat step 2 to create flows for all subreports in the parallel section. If the parallel section contains two subreports, create two flows.

4 Check the size of the subpage. It must be equal to or smaller than the size of the main report's page flow.

5 Adjust the size and position of each flow to fit within the subpage, as shown in Figure 13-9.

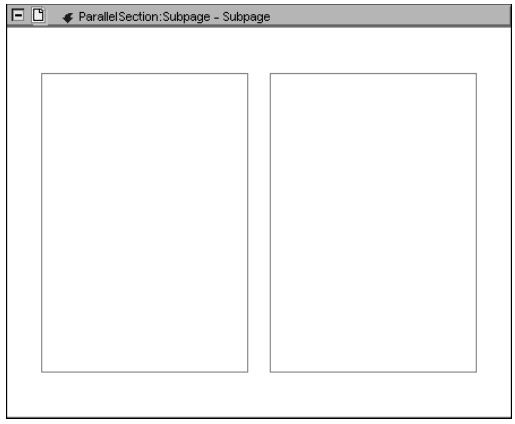


Figure 13-9 Making adjustments to page flows

How to assign flows to parallel subreports

1 Right-click the first report section in the parallel section, then choose Properties.

2 In the Properties page for the report section, choose Advanced Properties.

3 Set the FlowName property to the name of the flow in which the subreport is to appear. The flow name that you type must match exactly the name that you assigned to the flow when you created it. Class names are case-sensitive.

4 Repeat steps 1 and 3 to assign flows to all the report sections in the parallel section, as shown in Figure 13-10.

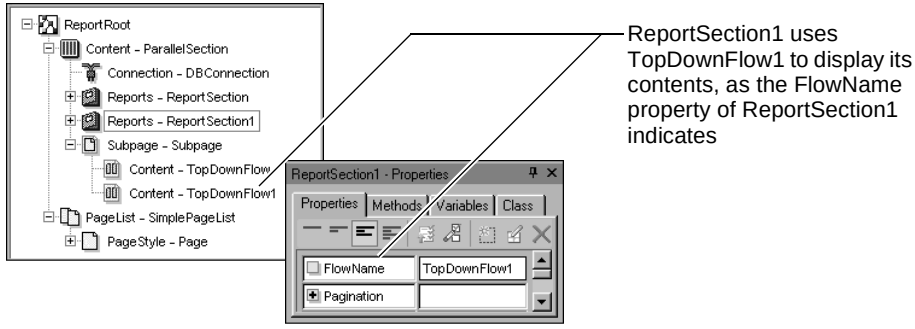


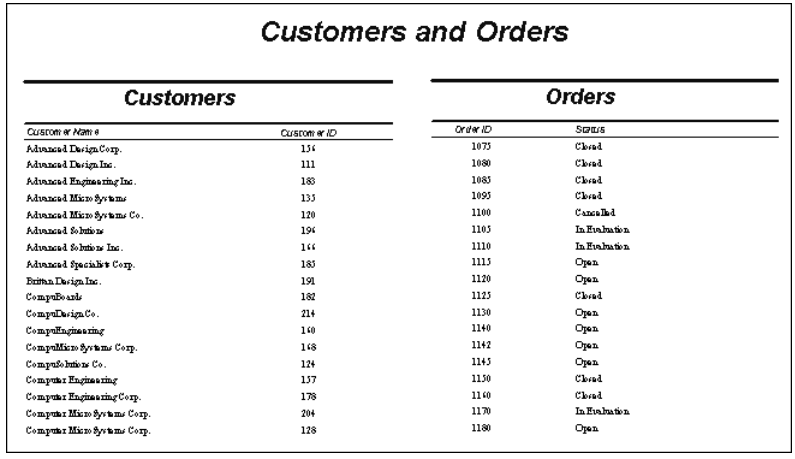
Figure 13-10 Assigning flows to parallel subreports

Building a report with side-by-side subreports

This section provides step-by-step instructions for building a report that contains a parallel section consisting of two subreports. The first subreport displays a list of customer names and customer IDs. It uses data from the Customers table in the Sfdata database. The second subreport displays a list of order IDs and statuses. It uses data from the Orders table in the Sfdata database. The subreports are independent of one another. The data is not linked in any way.

In this tutorial, the data for each subreport comes from the same database. This condition, however, is not a requirement. With a parallel section, you can create subreports that access different databases, even different data sources. For example, one subreport can access data from a database and the other can access data from a text file.

Figure 13-11 shows the finished report.



Customers and Orders

Customers

Customer Name	Customer ID
Advanced Design Corp.	134
Advanced Design Inc.	111
Advanced Engineering Inc.	188
Advanced Micro Systems	133
Advanced Micro Systems Co.	120
Advanced Solution	194
Advanced Solution Inc.	144
Advanced Specialist Corp.	185
Britton Design Inc.	191
CompuBoards	182
CompuDesign Co.	214
CompuEngineering	140
CompuMicro Systems Corp.	148
CompuSolutions Co.	124
Computer Engineering	137
Computer Engineering Corp.	178
Computer Micro Systems Corp.	204
Computer Micro Systems Corp.	128

Orders

Order ID	Status
1075	Closed
1080	Closed
1085	Closed
1095	Closed
1100	Cancelled
1105	In Evaluation
1110	In Evaluation
1115	Open
1120	Open
1125	Closed
1130	Open
1140	Open
1142	Open
1145	Open
1150	Closed
1160	Closed
1170	In Evaluation
1180	Open

Figure 13-11 Parallel subreports

In this tutorial, you perform the following tasks:

- Create a new report and add a parallel section.
- Add the first subreport to the parallel section.
- Add the second subreport to the parallel section.
- Share a connection between the subreports.
- Design a page layout for the subreports.
- Assign a flow to each subreport.

- Run and view the report.
- Enhance the report.

How to create a new report and add a parallel section

1. Choose File➤New.
2. In Create New Report, select Blank Report. Choose OK. A simple report design appears.
3. Delete Content—Report from the report design.
4. Drag a parallel section from Toolbox—Structure and drop it on the Content slot, as shown in Figure 13-12.

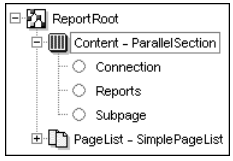


Figure 13-12 Adding a parallel section to a new report

5. Save the report design.

How to add the first subreport to the parallel section

You can add a subreport in either of the following ways:

- Use Insert➤Report. This option enables you to choose a report creation wizard that walks you through all the steps for creating a complete report, from selecting data fields to specifying a basic layout.
- Drag a report section from Toolbox—Structure and drop it in the parallel section's Reports slot. Using this option, you manually add data, structure, and visual components to the report to build the query, group data, and lay out and format the report contents. This option gives you more control over the layout and format of the report.

This tutorial uses the first way to create the subreport.

1. Complete the following steps to create a new report and connect to the Sfdata database:
   1. In the report structure, select Content—ParallelSection.
   2. Choose Insert➤Report.
   3. In Create New Report, select Listing Report Wizard. Choose OK.
   4. In Report Wizard—Data Sources, accept the default selection. Choose Next.

5. In Report Wizard—Choose Database, select SFData Sample Database. Choose Next.
6. In Database Login, choose OK.

2. In Report Wizard—Select Fields, complete the following steps to add data and apply sorting:
   1. In Table or View, select the customers table.
   2. In Available fields, double-click these fields in the following order:
      - customName
      - custID

      Choose Next.
   3. In Report Wizard—Specify Grouping, choose Next.
   4. In Report Wizard—Data Sorting, select customName from the first drop-down list, and accept the default ascending sort order. Choose Next.
3. In Report Wizard—Choose Layout Style, accept the default style, Columnar. Choose Next.
4. In Report Wizard—Finish:
   1. Accept the default report title, Customers.
   2. Select Change the report design. Choose Finish.

   The report design appears, as shown in Figure 13-13.

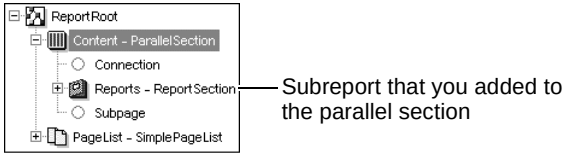


Figure 13-13 Adding a subreport to a parallel section

5. Expand Reports—ReportSection. Based on your choices, the wizard created a report section that contains these components:
   - Connection
   - Data stream
   - Before frame containing a label control for the subreport title, Customers
   - PageHeader frame containing label controls for column headings, Custom Name and Cust ID
   - Content frame containing data controls to display customer names and customer IDs

How to add the second subreport to the parallel section

1 Create the second subreport using the procedure that you followed to create the first subreport, but substitute the following information:

- In Report Wizard—Select Fields, in Table or View, select the orders table, and from Available fields, add these fields in the following order:
  - orderID
  - status
- In Report Wizard—Data Sorting, select orderID and sort in ascending order.

2 When you finish creating the subreport, expand it in Report Structure. Report Structure should look like Figure 13-14.

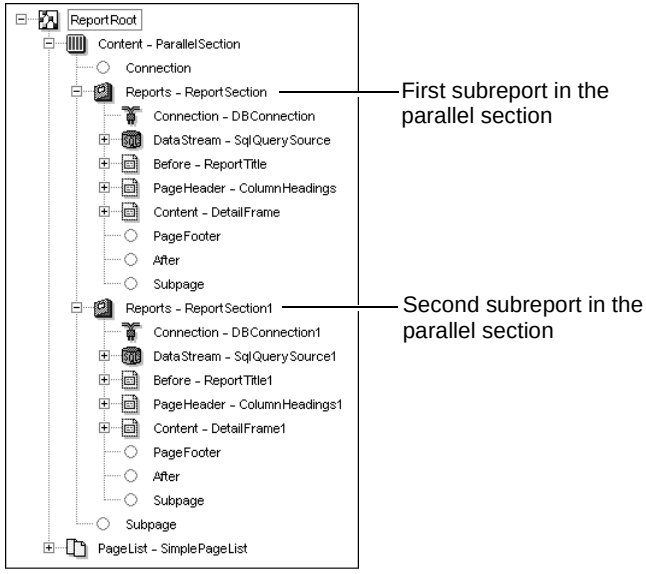


Figure 13-14 Adding multiple subreports to a parallel section

As the illustration shows, each subreport has its own connection and data stream query. Because both subreports connect to the same database, you can use one connection rather than two. For larger, more complex reports, sharing a connection improves performance. For instructions about sharing connections, see the next section.

How to share a connection

1 Move the connection component from the first subreport to the parallel section's Connection slot, as shown in Figure 13-15.

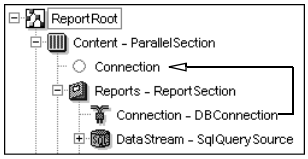


Figure 13-15 Moving connections

2 Delete the connection component from the second subreport.

How to create a page layout for the subreports

You create a page layout for the subreports in one of the following ways:

- Create a new page layout by inserting a subpage component in the parallel section's Subpage slot.
- Modify the main report's default page layout.

This tutorial uses the second way to create the page layout.

1 In the report structure, expand PageList—SimplePageList. This component and the components within it define the page layout for the entire report.

2 Expand PageStyle—Page. Report Structure shows the components within the page.

3 Right-click PageStyle—Page and choose Edit Layout. The layout shows the page layout, which contains a report title, a flow, a page number, and a date, as shown in Figure 13-16.

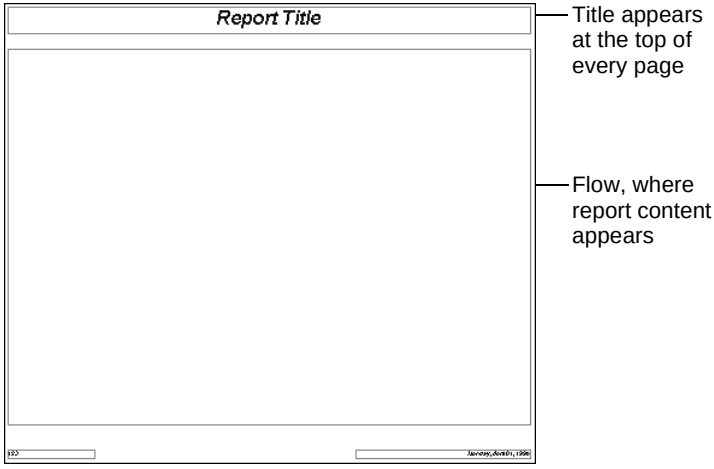


Figure 13-16 Creating a page layout for subreports

4 Change the report title to Customers and Orders.

5 Right-click the flow and choose Properties.

6 In the Properties page for the flow, change the Size➛Width property to 4.25 inches.

7 Rename the flow:

1 Right-click Content—:Flow. Choose Rename.

2 In Rename, type:

```
CustomersFlow
```

Choose OK.

8 Create a second flow:

1 In the page layout, select CustomersFlow.

2 Press Ctrl and drag the flow to the right. This action makes a copy of the flow. Position the copy to the right of the original flow.

3 In the report structure, right-click Content—CustomersFlow1. Choose Rename.

4 In Rename, type:

```
OrdersFlow
```

Choose OK. The layout looks like Figure 13-17.

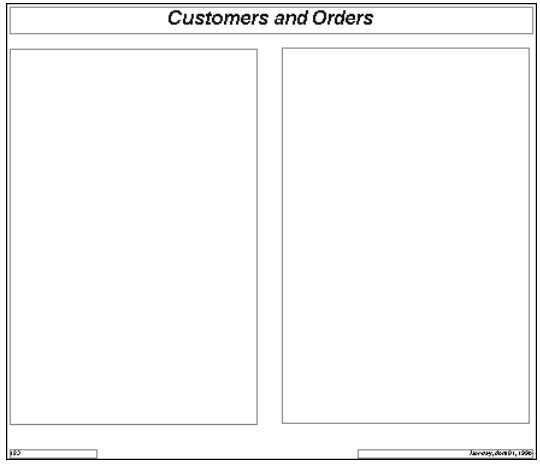


Figure 13-17 Modifying the default page layout

How to assign a flow to each subreport

1 In the report structure, double-click the first subreport, Reports—ReportSection.

2 In the Properties page for the subreport, in FlowName, type:

```
CustomersFlow
```

The flow name that you type must match exactly the name that you assigned to the flow. Class names are case-sensitive.

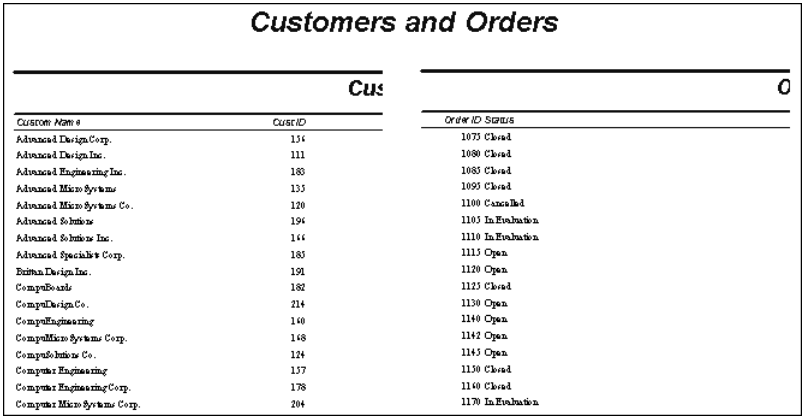
3 Double-click the second subreport, Reports—ReportSection1.

4 In the Properties page for the subreport, in FlowName, type:

```
OrdersFlow
```

How to run and view the report

1 Choose Report➛Build and Run. The first page of the report appears, as shown in Figure 13-18. Immediately, you observe some design problems. The flows are not wide enough to display the title of each subreport and the columns in the second subreport are too close together.



Customers and Orders

Custom Name	Cust ID
Advanced Design Corp.	134
Advanced Design Inc.	111
Advanced Engineering Inc.	180
Advanced Micro Systems	135
Advanced Micro Systems Co.	120
Advanced Solutions	194
Advanced Solutions Inc.	144
Advanced Specialties Corp.	185
Britton Design Inc.	191
CompuBoards	182
CompuDesign Co.	214
CompuEngineering	140
CompuMicro Systems Corp.	148
CompuSolutions Co.	124
Computer Engineering	157
Computer Engineering Corp.	178
Computer Micro Systems Corp.	204

Order ID	Status
1073	Closed
1080	Closed
1085	Closed
1093	Closed
1100	Cancelled
1105	In Evaluation
1110	In Evaluation
1115	Open
1120	Open
1125	Closed
1130	Open
1140	Open
1142	Open
1145	Open
1150	Closed
1160	Closed
1170	In Evaluation

Figure 13-18 Viewing the side-by-side subreports

2 Choose View➛Design to return to the report design. The report design appears.

How to improve the report design

1 In the layout, move the status controls in the Orders subreport to the right to add more space between the Order ID and Status.

2 Center the subreport titles above the data controls.

3 Run and view the report.

4 Choose View➛Next Page to page through the report to check for other design issues.

Building a report with sequential subreports

This section provides instructions for building a report that contains a sequential section consisting of two subreports. The report displays the same data as the report that you created in the previous section, "Building a report with side-by-side subreports." This time, the subreports appear sequentially rather than side by side.

In this tutorial, the data for each subreport comes from the same database. This condition, however, is not a requirement. With a sequential section, you can create subreports that access different databases, even different data sources. For example, one subreport can access data from a database and the other can access data from an Oracle application.

Figure 13-19 shows how the subreports look in the finished report.

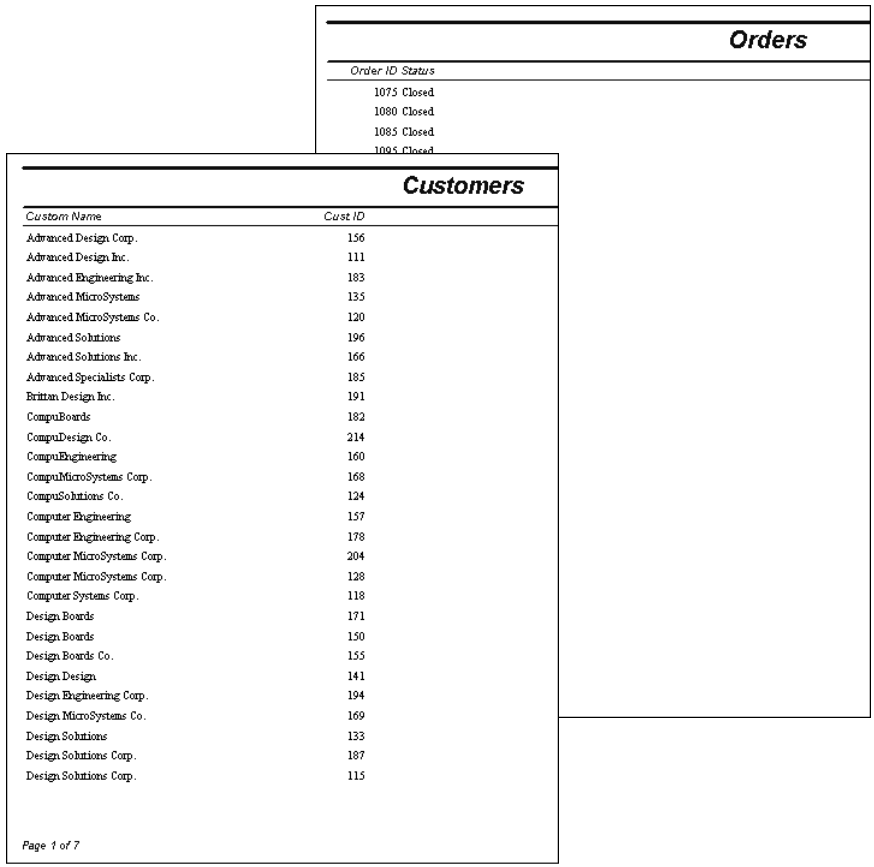


Figure 13-19 Sequential subreports

In this tutorial, you perform the following tasks:

- Create a new report and add a sequential section.
- Add subreports to the sequential section.
- Insert a page break between the subreports.
- Run and view the report.

How to create a new report and add a sequential section

1. Choose File➤New.
2. In Create new Report, select Blank Report. Choose OK. A simple report design appears.
3. Delete Content—Report from the report design.
4. Drag a sequential section from Toolbox—Structure and drop it in the Content slot. A sequential section appears in the report design, as shown in Figure 13-20.

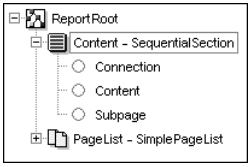


Figure 13-20 Adding a sequential section to the report

5. Save the report design.

How to add subreports to the sequential section

1. In the report structure, select Content—SequentialSection.
2. Choose Insert➤Report.
3. In Create New Report, select the Listing Report Wizard to create the first subreport using the following information:
   1. Connect to the Sfdata sample database.
   2. Select the customers table and these fields in the following order:
      - customName
      - custID
   3. Sort this data by customName.
   4. Accept the default layout style and report title. Choose Finish. Content—ReportSection appears in the report structure.
4. In the report structure, select Content—SequentialSection.

5 Choose Insert➤Report.

6 In Create New Report, select the Listing Report Wizard to create the second subreport using the following information:

   1 Connect to the Sfdata sample database.

   2 Select the orders table and these fields in the following order:

      - orderID
      - status

   3 Sort this data by orderID.

   4 Accept the default layout style and report title. Choose Finish. Content—ReportSection1 appears in the report structure.

7 Expand the report sections in the report structure. The report structure should look like Figure 13-21.

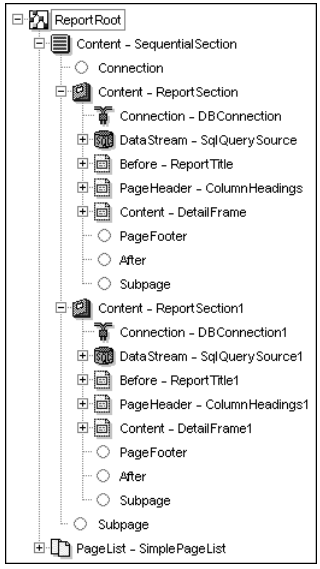


Figure 13-21 Adding sequential subreports

How to run and view the report

1 Choose Report➤Build and Run. The report appears in the view perspective.

2 Choose View➤Next Page to page through the report to check the output. The Customers subreport appears first, followed by the Orders subreport, as shown in Figure 13-22.

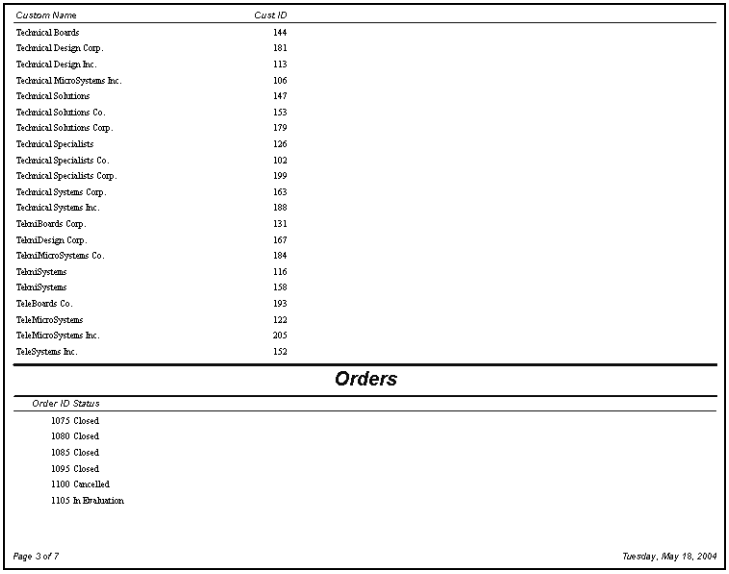


Figure 13-22 Viewing sequential subreports

How to insert a page break between the subreports

1. Choose View➛Design. The report design appears.
2. Right-click the first subreport, Content—ReportSection, then choose Properties.
3. In the Properties page for the subreport, set Pagination➛PageBreakAfter to True, as shown in Figure 13-23.

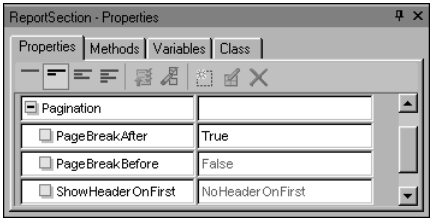


Figure 13-23 Page-break options

4. Run the report and view the results. The Orders subreport starts at the top of a new page.

Building a report with linked subreports

In the preceding tutorials, we create reports containing subreports that are not linked to each other. The subreports run independently and do not coordinate their data. Sometimes, however, you need to create reports that coordinate data, such as a detailed customer issues report within a customer report, or a top-ten holdings report within a mutual fund report.

Figure 13-24 shows examples of linked reports.

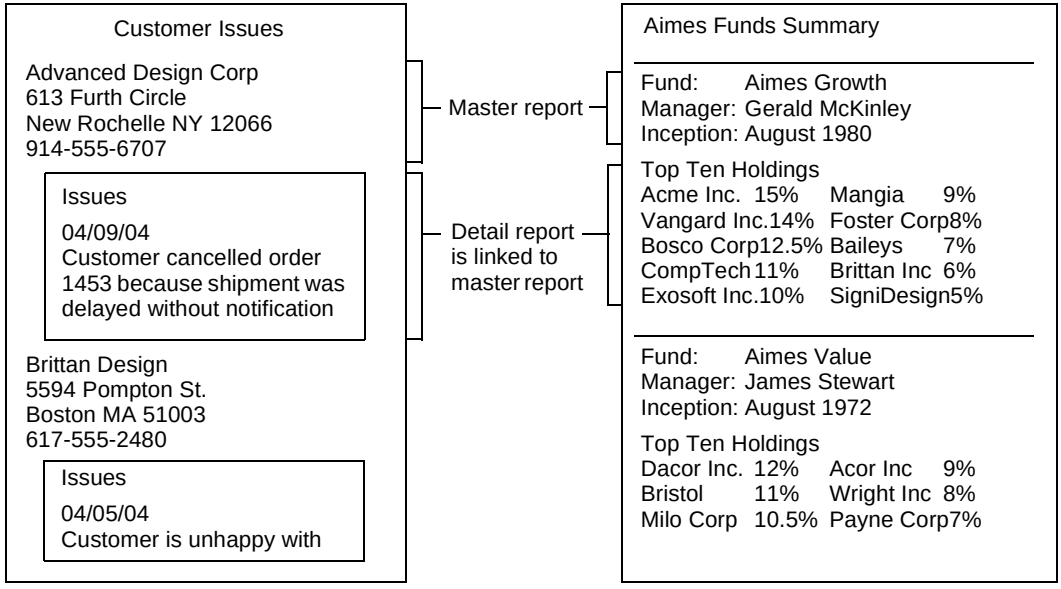


Figure 13-24 Linked subreports

In both examples, one report is nested in another, creating a master report and detail report relationship. Each report accesses data from a different database, and the reports must be linked by a common field value, such as customer ID or mutual fund name. The value of the linking field in the master report determines the data that appears in the detail report. For example, if the customer ID in the master report is 786, the detail report displays the issues for the customer whose ID is 786.

These reports resemble a single report with group sections in which data from two tables are linked, then sorted by a common field. In fact, to create a report that links data from two tables in a single database, you can use either technique—create two linked reports or a single report with two group sections. It is better, however, to use the second technique to improve performance.

Typically, you create linked reports only if your report needs data from tables in different databases.

The rest of this section provides step-by-step instructions for creating a linked report. The master report accesses data from the offices table in the Sfdata sample database to list the regional offices. The detail report accesses data from the OrdersByOffice table in the Quikdata sample database to list the orders for each office. The common field between the two tables is city.

In this tutorial, you perform the following tasks:

- Create a new report.
- Build the query for the master report.
- Group data in the master report by office city.
- Lay out data in the master report.
- Create the detail report.
- Create a parameter for the field that links the reports.
- Write a function to assign values to the parameter.
- Run and view the report.

How to create a new report

1 Choose File➤New.

2 In Create New Report, select Blank Report. Choose OK. A simple report design appears.

3 Expand Content—Report.

4 Delete Content—Frame. The report design appears, as shown in Figure 13-25.

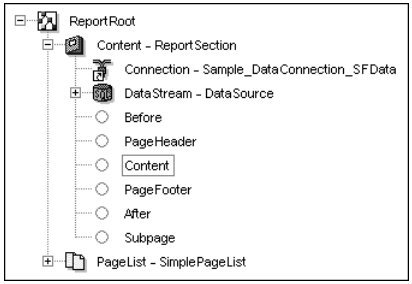


Figure 13-25 Creating a new report

5 Save the report design.

How to build the query for the master report

1 In the report structure, select DataStream—DataSource.

2 Choose View➤Data. Database Login appears. Choose OK. Datasource—Query Editor appears.

3 Drag the offices table from Database Browser to Query Editor, as shown in Figure 13-26.

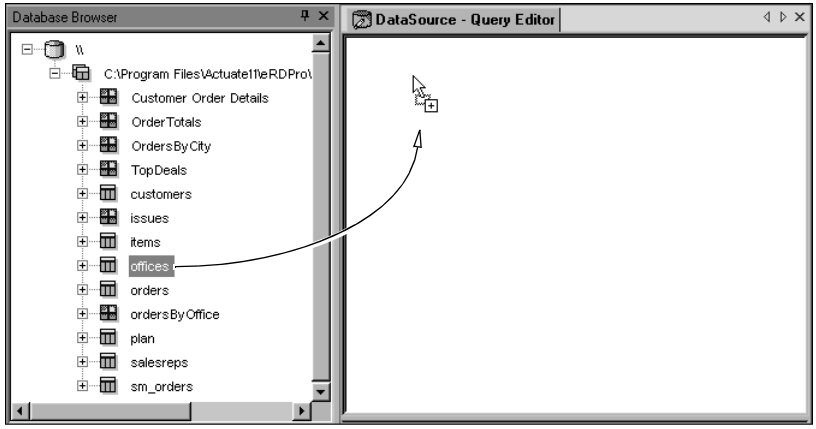


Figure 13-26 Building a query

4 To add all fields from the offices table, drag the asterisk (*) from the offices table to Column Name. All table names appear in Columns, as shown in Figure 13-27.

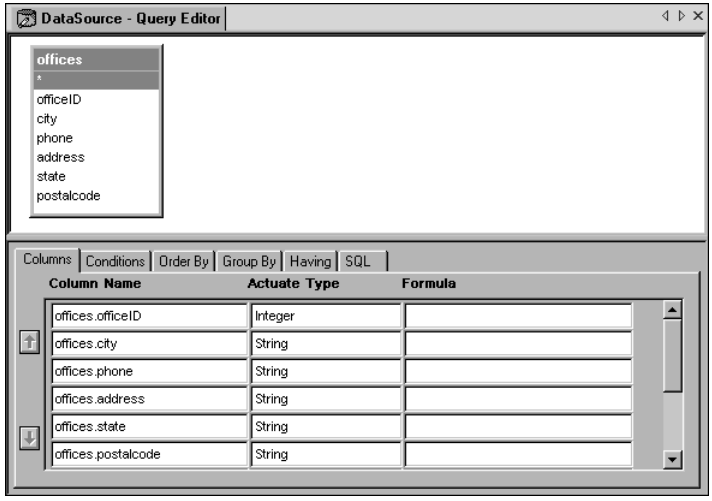


Figure 13-27 Query Editor

How to group data in the master report by office city

1 Choose View➤Design. The report design appears.

2 Choose Tools➛Grouping.

3 In Grouping & Sorting, in Available fields, to group data on the city field, select offices.city. Choose >. Choose OK. A group section appears in the Content slot of the master report, Content—Report.

4 Expand Content—OfficesCityGroup. A frame appears in the Before slot of the Content—OfficesCityGroup group section, as shown in Figure 13-28.

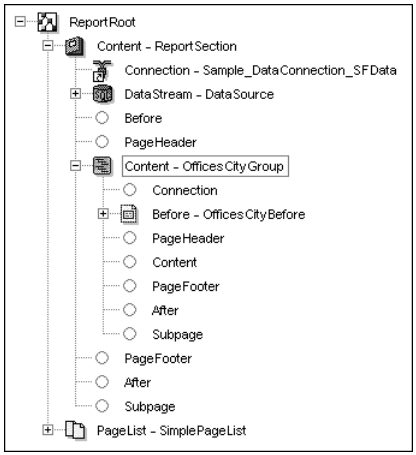


Figure 13-28 Grouping data

How to lay out data in the master report

1 Insert the following database fields in the Before—OfficesCityBefore frame:

- offices.address
- offices.state
- offices.postalcode
- offices.phone

2 Insert a computed field in the same frame above the database fields and use the following expression:

```
[offices.city] & " Office"
```

3 Arrange and format the components in the layout so that they appear, as shown in Figure 13-29.

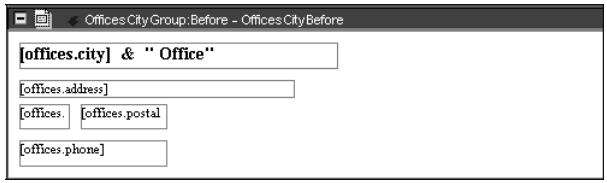


Figure 13-29 Laying out data

How to create the detail report

1 In the report structure, in the Content—OfficesCityGroup group section, select the Content slot. Choose Insert➛Report.

2 In Create New Report, select the Listing Report Wizard to create the detail report using the following information:

1. Connect to the Quikdata sample database.
2. In Report Wizard—Select Fields, select the ordersByOffice table and add these fields in the following order:
   - orderID
   - orderTotal
3. Sort this data by orderID
4. Accept the default layout style and report title. The report structure for both the master and detail reports shows no empty slots, like the example in Figure 13-30.

Figure 13-30 Master and detail report structure

How to specify the field with which to link the reports

The common field between the two reports is city. You use the city field as a parameter in the detail report, so that the detail report can get the current city value from the master report when the report runs.

1. In the report structure, select the detail report's data stream component, DataStream—SqlQuerySource.
2. Choose View➛Data.
3. In SqlQuerySource—Query Editor, choose Conditions.
4. In Conditions, drag the city field from the ordersByOffice table to Column Name. ordersByOffice.city appears in Column Name.
5. Type the following expression in Query Expression, as shown in Figure 13-31:

```
=:CityParam
```

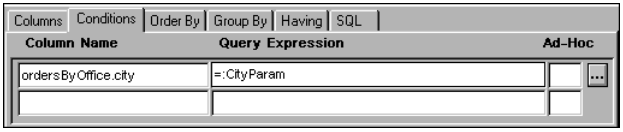


Figure 13-31 Defining parameters for linked reports

You assigned the city field to a parameter called CityParam. Now you need to write code to assign the proper values to this parameter at run time.

How to write a function to assign values to the parameter

1. In the report structure, right-click the detail report, Content—ReportSection, then choose Properties.
2. In the Properties page for the report, choose Methods.

3. In the Methods page, select Function BuildFromRow(row As AcDataRow) As AcBuildStatus. Choose Override.

   For information about the BuildFromRow( ) method, see *Programming with Actuate Foundation Classes.*

4. Type the following code:

```
Function BuildFromRow( row As AcDataRow ) As AcBuildStatus
   If Not row Is Nothing Then
        SqlQuerySource::CityParam = row.GetValue( "city" )
   End If
   BuildFromRow = Super::BuildFromRow( row )
End Function
```

How to hide the parameter from the report user

When you create a parameter, e.Report Designer Professional displays the parameter on Requester by default. Because you set the value of the CityParam parameter through code, you should hide this parameter from your report users so they do not enter a value at run time.

1. Choose Tools➛Parameters.
2. In Parameter Editor, select SqlQuerySource::CityParam. Choose Modify.
3. In Parameters Properties—General, select Hidden. Choose OK.
4. Close Parameter Editor.

How to run and view the report

1. Choose Report➛Build and Run.
2. In Requester—<report name>.rop, to stop Requester from appearing in subsequent runs, select Do not prompt for parameter values. Choose OK. The report appears.
3. Choose View➛Next Page to page through the report. The report looks like Figure 13-32. The master report displays the address and phone number of each office and the detail report displays the orders for each office.

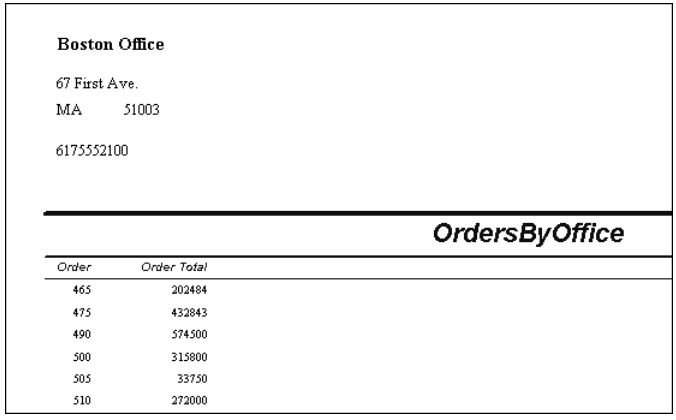


Figure 13-32 Viewing linked reports

Building a report with conditional sections

This section provides step-by-step instructions for building a report that uses a conditional section to display one of three frames when a user runs the report. The first frame displays the list of sales representatives in Boston if the user specifies 1 for an office ID parameter. The second frame displays the list of sales

representatives in New York City if the user specifies 2, and the third frame displays the list of sales representatives in Philadelphia if the user specifies 3.

A conditional section has only two slots, Then and Else, in which to place conditional frames. Figure 13-33 shows the structure of a conditional section.

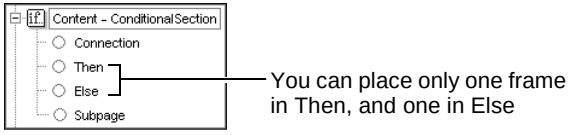
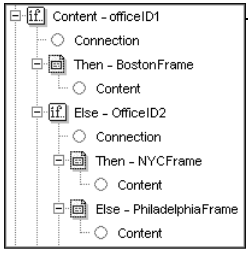


Figure 13-33 Conditional section structure

This report, however, requires three conditional frames. To accommodate three conditions in two slots, you need to use nested conditional sections.

Place a second conditional section in the Else slot of the first conditional section to set up the following condition shown in Figure 13-34.



This report structure sets up the following condition:

```
If officeID = 1 Then Boston
   frame
Else
If officeID = 2 Then NYC
   frame
Else Philadelphia frame
```

Figure 13-34 Nested conditional sections

In this tutorial, you perform the following tasks:

- Create a new report and add the conditional sections and frames.
- Build the query to specify the data to retrieve.
- Create an ad hoc parameter to prompt the user to specify which office information to display.
- Specify the condition that determines which of the three frames is displayed when the user runs the report.
- Lay out data in the three frames.
- Run and view the report.

How to create a new report and add conditional sections and frames

1. Create a new report and delete the frame in its Content slot:
   1. Choose File➛New.
   2. In Create New Report, select Blank Report. Choose OK. A simple report design appears.

3. Expand Content—Report.
4. Delete Content—Frame from the report design.

2. Drag a conditional section from Toolbox—Structure and drop it in the Content slot.
3. Rename the conditional section:

   `IfOfficeID1`

4. Drag a frame from Toolbox—Structure and drop it in the conditional section's Then slot.
5. Rename the frame:

   `BostonFrame`

6. Add another conditional section and two frames to the report structure:
   1. Drag a conditional section from Toolbox—Structure and drop it in the first conditional section's Else slot.
   2. Rename the conditional section:

      `IfOfficeID2`

   3. Drag a frame from Toolbox—Structure and drop it in the second conditional section's Then slot.
   4. Rename the frame:

      `NYCFrame`

   5. Drag a frame from Toolbox—Structure and drop it in the second conditional section's Else slot.
   6. Rename the frame:

      `PhiladelphiaFrame`

7. Save the report design. The report design looks like Figure 13-35.

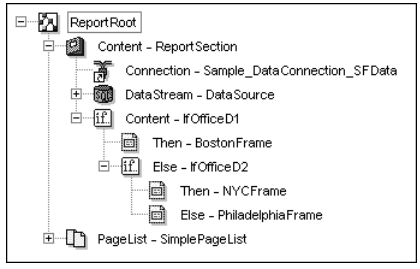


Figure 13-35 Adding conditional sections and frames

How to build the query

1. Choose View➛Data.
2. In Database Login, choose OK without specifying a user name or password. DataSource—Query Editor and Database Browser appear.
3. Drag the following tables from Database Browser to Query Editor:
   - offices
   - salesreps
4. Drag the following fields from the offices table to Column Name:
   - officeID
   - city
5. Drag the following fields from the salesreps table to Column Name:
   - first
   - last

   Query Editor—Columns appears, as shown in Figure 13-36.

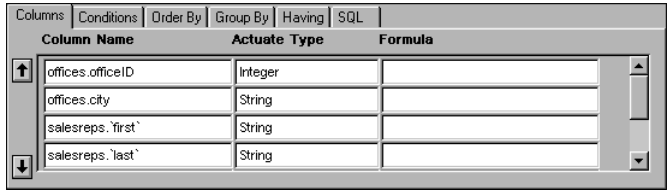


Figure 13-36 Query Editor—Columns

How to create the ad hoc parameter

1. Choose Conditions. Query Editor—Conditions appears.
2. Drag the officeID field from the offices table and drop it in Column Name.
3. Select Ad Hoc. e.Report Designer Professional creates an ad hoc parameter.
4. Choose View➛Design. The report design appears.
5. Choose Tools➛Parameters. Parameter Editor displays the new parameter, DataSource::offices_officeID, as shown in Figure 13-37. Note the parameter name. You need it when you specify the expression for the conditional section.
6. Close Parameter Editor.

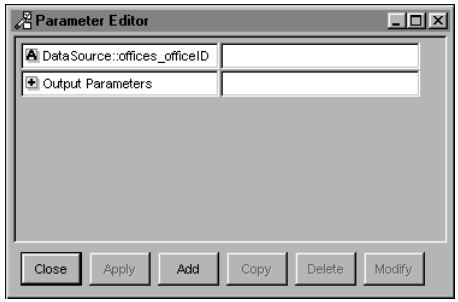


Figure 13-37 Creating an ad hoc parameter

How to specify the conditions to determine which frame to display

1. Right-click Content—IfOfficeID1, then choose Properties.
2. In the Properties page for the frame, type the following expression for the IfExp property:

   ```
   DataSource::offices_officeID = 1
   ```

3. Select Else—IfOfficeID2.
4. Type the following expression for the IfExp property:

   ```
   DataSource::offices_officeID = 2
   ```

How to lay out data in the frames

1. Choose Insert➛Computed Field. Drop the computed field in Then—BostonFrame.
2. In Expression Builder, type the following expression to display the sales representative names:

   ```
   [salesreps.first] & " " & [salesreps.last]
   ```

   Choose OK. Choose Control Type appears.
3. Choose OK to accept the default control type, text. A text control appears in the frame.
4. Increase the width of the text control.
5. Copy the text control to the following frames, as shown in Figure 13-38:
   - Then—NYCFrame
   - Else—PhiladelphiaFrame

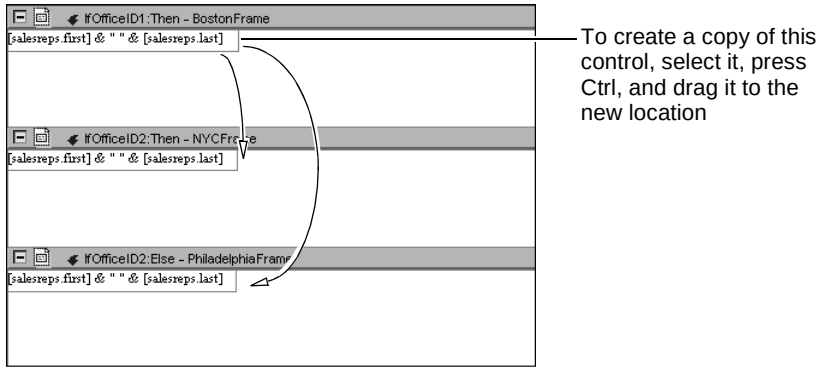


Figure 13-38 Adding a text control

6. Decrease the height of all three frames to reduce the amount of space between the rows of names in the finished report.
7. Set each frame's background color to a different color.

How to add a title that reflects the contents of the displayed frame

1. Delete the default report title from the page layout:
   1. In the report structure, expand PageList—SimplePageList, then expand PageStyle—Page.
   2. Delete Content—ReportTitle.
2. Drag a frame from Toolbox—Structure and drop it in the Before slot of Content—Report.
3. Choose Insert➛Computed Field and drop the computed field in the Before—Frame.
4. In Expression Builder, type the following expression:

   ```
   "Sales Representatives in " & [offices.city]
   ```

   Choose OK.
5. In Choose Control Type, choose OK to accept the default control type, text. A text control appears in the frame as shown in Figure 13-39.
6. Increase the width of the text control.
7. Decrease the height of the frame.

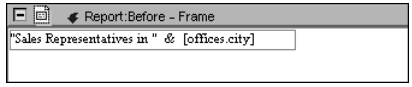


Figure 13-39 Adding a title

How to run and view the report

1. Choose Report➛Build and Run. Requester—<report name>.rop prompts you to provide a parameter value.
2. Type 1 for offices_officeID. Choose OK. The report displays the list of sales representatives in Boston, as shown in Figure 13-40. The title above the list is Sales Representatives in Boston.

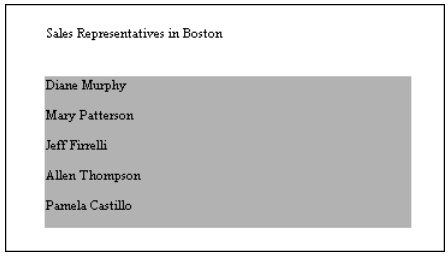


Figure 13-40 Report output of Sales Representatives in Boston

3. Run the report two more times to view the NYC and Philadelphia lists. Type 2 and 3, respectively, for the offices_officeID parameter.

Examining a report containing multiple sections and subreports

This section describes the design of the example report Forecast, which displays multiple parallel and sequential subreports in one report. The first page of this report contains a summary view of the report's contents. There are four mini-subreports on this page. After the summary page, the following subreports appear sequentially:

- Issues
- Top Deals
- Negative Changes
- Positive Changes

Figure 13-41 shows the following report pages:

- Page one, which presents a summary view of the report
- The first page of the Issues subreport, which starts on page 2 of the report
- The first page of the Top Deals subreport
- The first page of the Negative Changes subreport

Notice how the layout of the first page is different from the layout of the other pages.

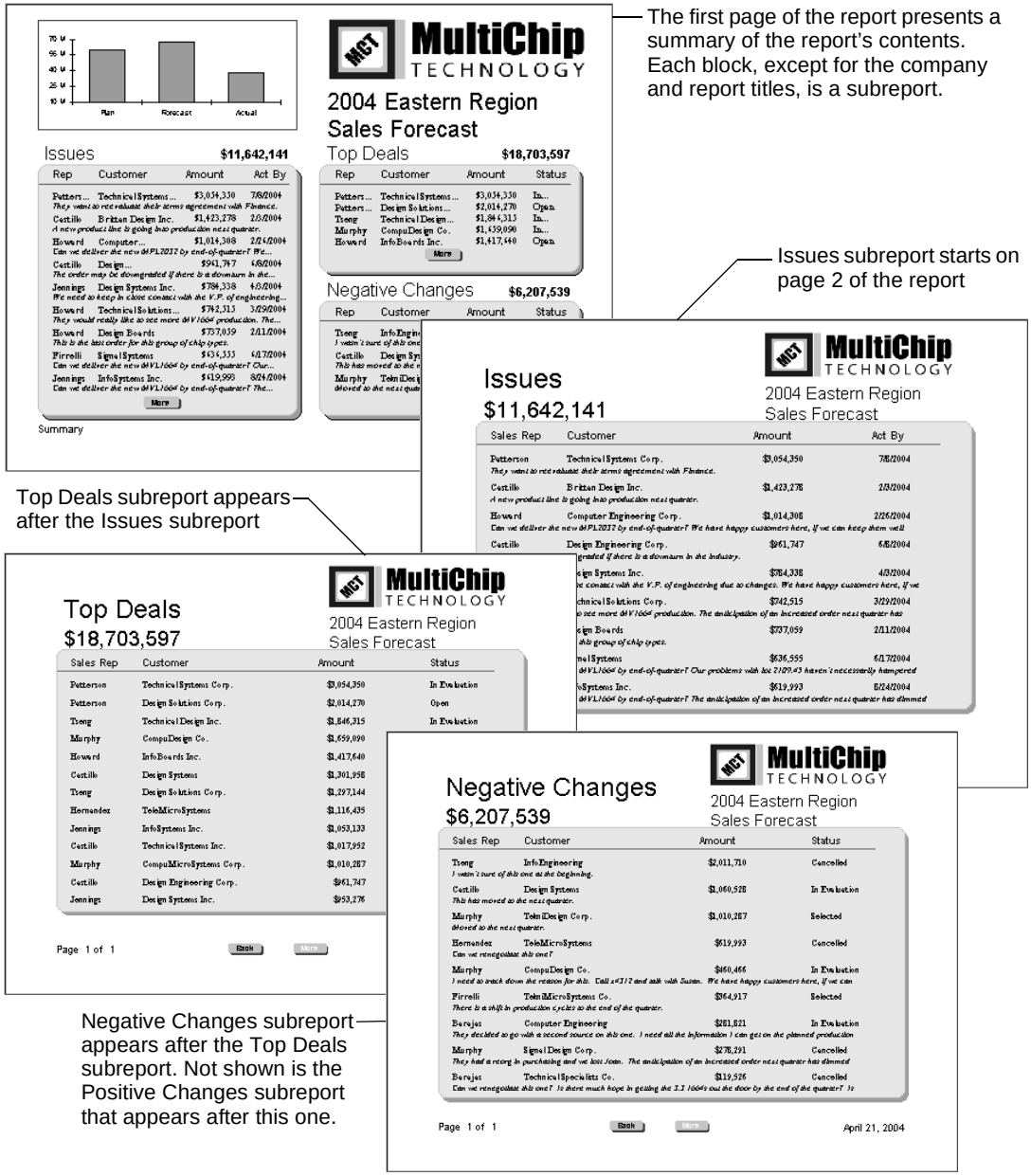


Figure 13-41 Example report Forecast with four subreports

Understanding the structure of the main report

The report design organizes these subreports in the following order:

- In the summary page:
  - A chart
  - An abbreviated Issues subreport
  - An abbreviated Negative Changes subreport
  - An abbreviated Top Deals subreport
- A full Issues subreport
- A full Top Deals subreport
- A full Negative Changes subreport
- A full Positive Changes subreport

The structure of the report reflects the order of the subreports and the relationship among them. First, a sequential section organizes the order in which the subreports appear. For the four subreports that appear in multiple blocks on a single page, a parallel section is used to organize the subreports in multiple flows.

Figure 13-42 shows the high-level report structure.

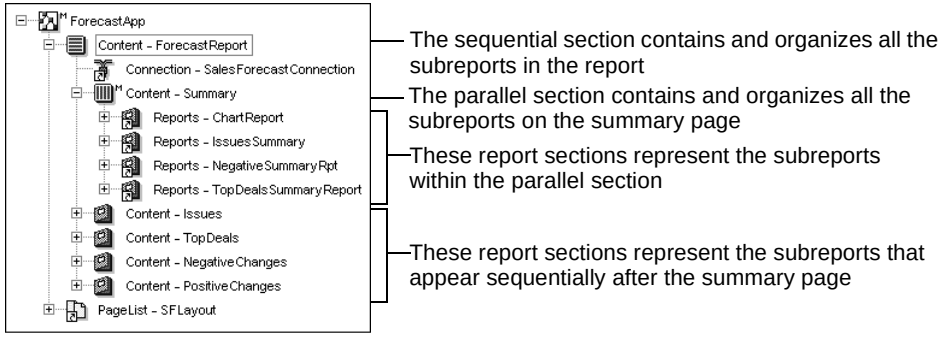


Figure 13-42 High-level report structure

Understanding connections and data streams

Every subreport uses its own data stream, but because all the subreports access data from the Sfdata database, all the subreports share a single connection component. The top-most Connection slot contains this connection component, as shown in the previous illustration.

Understanding the organization of the subreports on the summary page

To generate the summary subreports on the first page, the report design contains four subreports within a parallel section. Figure 13-43 shows the relevant part of the report design and the corresponding results.

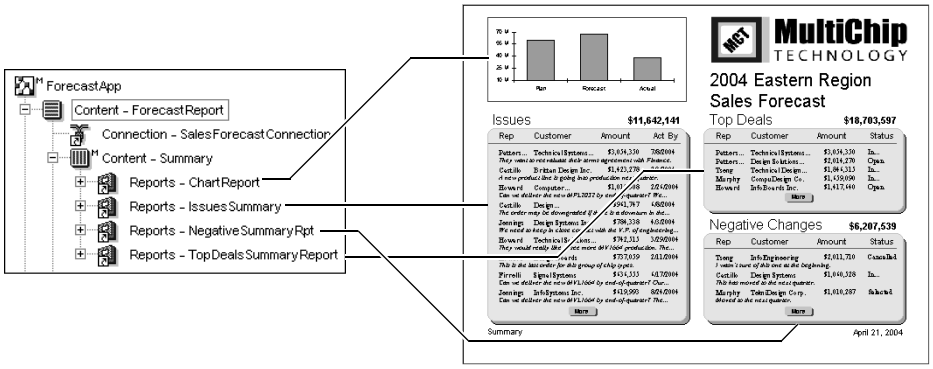


Figure 13-43 Report design components matched to report sections

To display the subreports in the block formation, the report developer completes the following tasks:

- Create a page layout that contains multiple flows, one flow for each part of the page in which a subreport appears.
- Assign a specific flow to each subreport. The report developer sets the FlowName property of each subreport to a specific flow that appears in the page layout.

Understanding the report's page layout

In this report, all the pages in the report have the same layout, except for the first page. To achieve this look, the report developer modifies the report's main page component in the following ways:

- Instead of using the default Simple Page List component, the report design uses a Title and Body Page List component to design the overall page layout. This component enables the report to use one page layout for the first page of the report and a different page layout for the other pages in the report.
- This report design uses a page component in the Title Page slot of the Title and Body Page List component. This page component contains all the flows necessary to display each of the four subreports on the first page. It also contains graphical components to display the company logo, report title, date, and colored rectangles that appear as shadows behind each subreport.
- The report developer arranges and sizes each flow on the first page in the arrangement in which the subreports appear.

- The report developer adds a page component to the Body Page slot of the Title and Body Page List component. This component contains a single flow to display the contents of the other subreports. It also contains graphical components that appear on all the pages, except the first, of the report.

Figure 13-44 shows the components used to create the report's page layout.

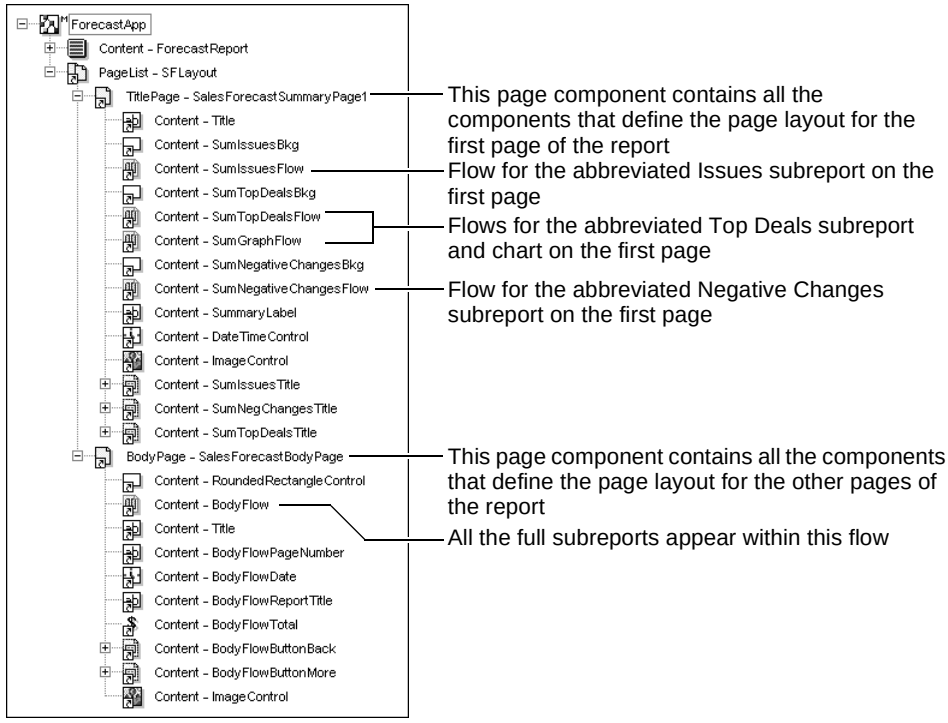


Figure 13-44 Components used to create the page layout

Figure 13-45 shows the layout for the first page. The illustration also shows the four flows selected. Notice how their arrangement reflects what you see in the finished report.

Understanding how to assign flows to subreports on the summary page

To display the subreports in the appropriate flows, each subreport's FlowName property is set to a specific flow. This step is necessary only for subreports that appear on a page with multiple flows.

The subreports that appear on page 2 and later do not have to explicitly specify the flow to use because the subreports appear on a page that contains only one flow, and, therefore, the subreport uses this flow by default.

Page layout for the first page as it appears in the layout pane

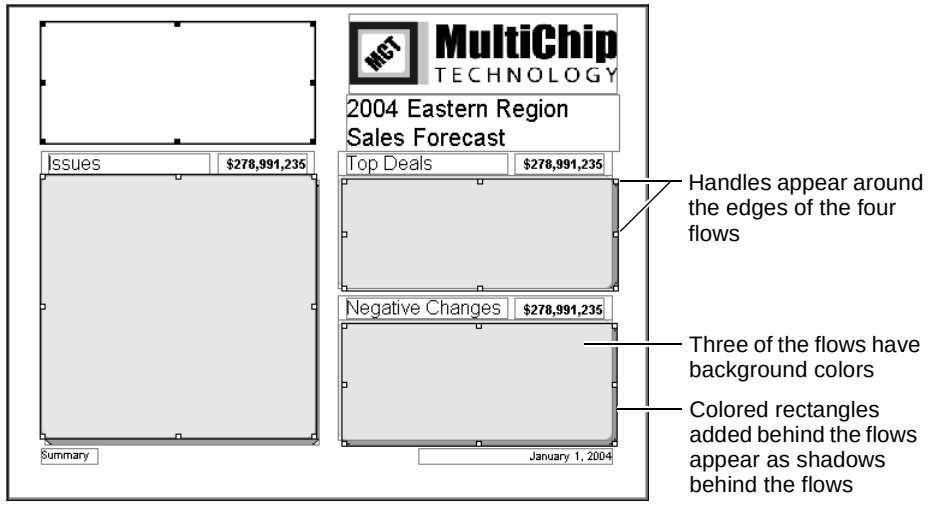


Figure 13-45 Layout for the first page of the report

Figure 13-46 shows how a flow corresponds to the IssuesSummary subreport that appears on the first page of the report. The same procedure applies to the chart, Top Deals, and Negative Changes subreports that appear on the first page.

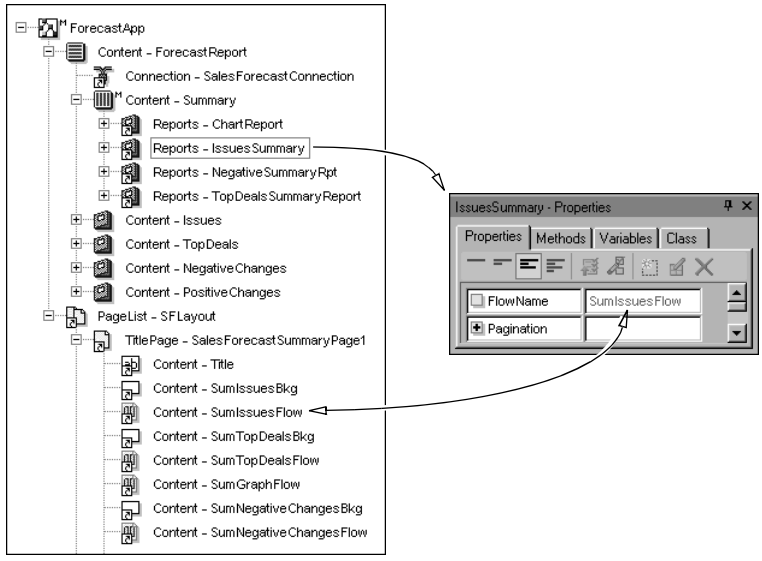


Figure 13-46 Flow relationship of a report to a subreport

Chapter

14

Adding interactive viewing features

This chapter contains the following topics:

- About interactive viewing events
- Providing balloon help
- Providing search functionality in a report
- Developing hyperlink functionality
- Developing a table of contents

About interactive viewing events

An interactive viewing event occurs when a report user views a report and takes an action that results in:

- Displaying balloon help
- Initiating a search for content within a report
- Activating a hyperlink
- Viewing a table of contents for a report

You can design a report that supports any of these events. For example, you can design help text that explains the purpose of a text control.

Providing balloon help

Balloon help text appears when a user hovers a mouse pointer over a component. The text disappears when the user presses a mouse button or moves the mouse pointer. The content of balloon help can be the formatted value of a control or any message that you specify. It is advisable to limit the length of a text message because balloon help appears for approximately five seconds.

You can provide balloon help text in the following ways:

- In a component's BalloonHelp property, type the help text to display.
- Override the BalloonHelp( ) method.

If you type the text in BalloonHelp and also override the BalloonHelp( ) method, the return value of the method appears instead of the property string. For more information about using the BalloonHelp( ) method, see Class AcControl in *Programming with Actuate Foundation Classes*.

Providing search functionality in a report

You can give report users the ability to search for components in a report on the web and in e.Report Designer Professional's view perspective. The report user can search for most types of report components. The components for which you cannot provide search functionality are:

- Connections
- Cross-tab controls
- Data rows

- Data streams
- Flows
- Page lists

To support searching for a component, you must set properties of the component using the Searching property group. The properties that you can set depend on the type of component. The top-level report component, for example, has one Searching property, SearchTag. A frame, by contrast, has five Searching properties. This section explains how to use each search property.

Figure 14-1 shows the search properties for a frame, an integer control, and a group section. To display the Searching properties for a component, choose Advanced Properties on the Properties page.

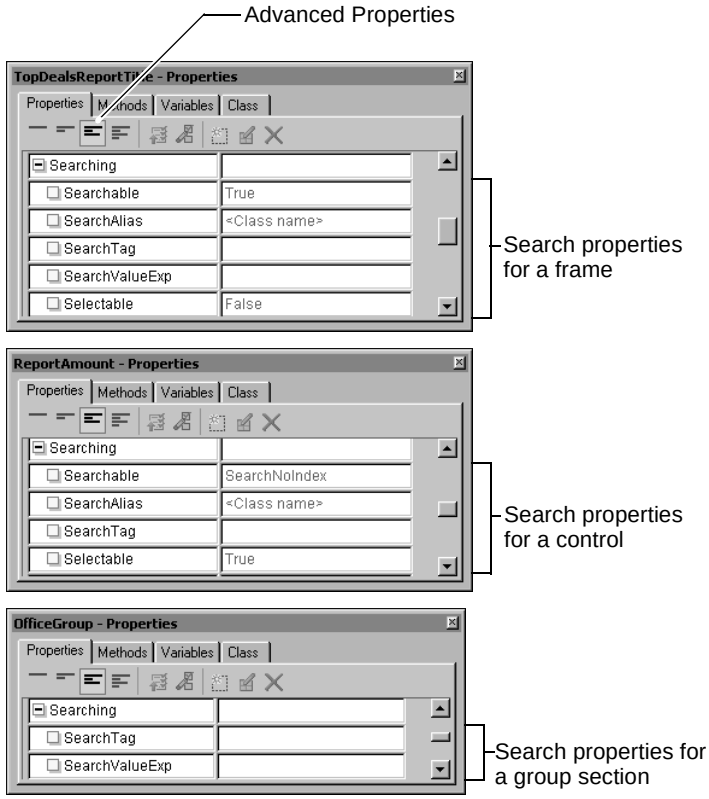


Figure 14-1 Search properties

Table 14-1 describes the properties that you can use to support searching for an object.

Table 14-1 Search property descriptions

Property	Description
Searchable	Determines whether the report user can search a report for the component. Values are: ■ NotSearchable. The report user cannot search for the component. ■ SearchNoIndex. The report user can search for the component. e.Report Designer Professional searches the entire report. SearchNoIndex is the default value. ■ SearchWithIndex. The report user can search for the component using a high-performance indexed search.
SearchAlias	The name to display to the report user in the Search window. The default value is <Class name>. Use SearchAlias only when you do not use SearchTag.
SearchTag	Uniquely identifies one or more searchable components. SearchTag is useful in complex reports where structures such as parallel and sequential sections can otherwise prevent successful searches. Use this property only when you do not use SearchAlias.
SearchValueExp	Specifies the component value for which to search. SearchValueExp can evaluate to a column name or to an expression. You can set SearchValueExp for a frame, a section, or a page.
Selectable	Determines whether a user can select the component to add it to a search. The default value is False.

Naming a component in a search

When a user adds a component to a search, the component name appears in the Search window, which is available in the view perspective. The component name in Search comes from the SearchAlias property. The default value of SearchAlias is the component's class name. You can change the SearchAlias property value to display a different name in the Search window.

Figure 14-2 shows how the default value of SearchAlias appears in the report design and also shows the class name in the Search window.

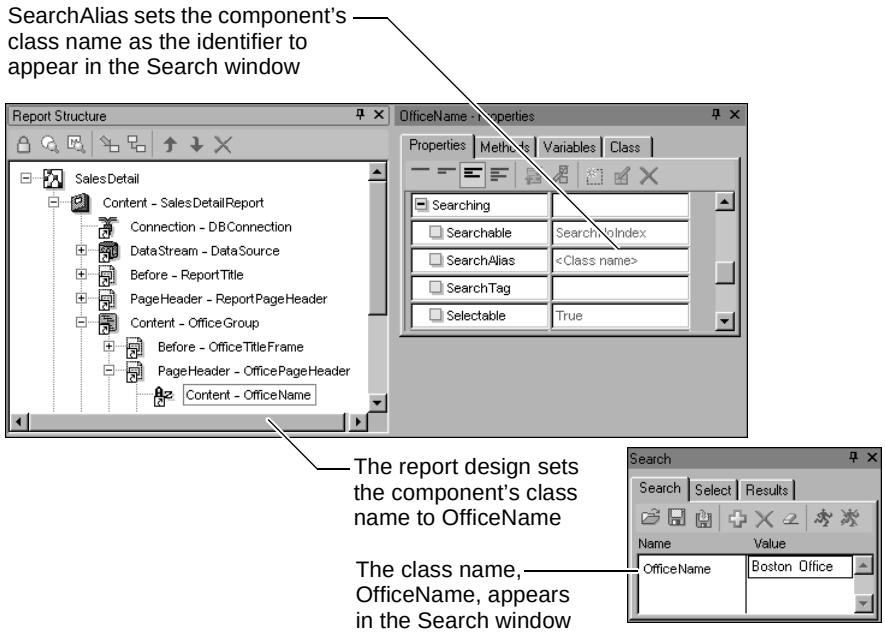


Figure 14-2 SearchAlias property

The value that you specify for SearchAlias does not appear in the Search window if:

- A report user provides a value for the component's SearchTag property and a value for SearchAlias. In this case, the SearchTag value appears in Search.
- A report user adds a component to the Search window and its SearchAlias value already appears as a name in the list. When there are duplicate SearchAlias values, e.Report Designer Professional adds a hyphen (-) and a number to the duplicate name, as shown in the following examples:

```
ForecastAmount
ForecastAmount-1
ForecastAmount-2
```

Creating versatile searches

Using the SearchTag property for a report component, you can enhance the report user's ability to search for data in multiple ways. For example, you can use SearchTag to support:

- Searching for multiple instances of the same class
- Searching for multiple classes
- Retrieving data from controls in multiple parallel or sequential sections

The searches that you create using SearchTag apply to reports that you view in the view perspective and on the web.

The default SearchTag value is an empty string. If you do not specify a SearchTag value, e.Report Designer Professional uses the report's structure to determine which search results to retrieve. If you set a SearchTag value, that value becomes the component's identifier for the purpose of searching. You can no longer use the scoped class name to identify the component in a search.

If you set both SearchTag and SearchAlias, e.Report Designer Professional uses the SearchTag value. To use SearchTag for a component, you must set the component's Searchable property to SearchNoIndex or SearchWithIndex. SearchWithIndex creates an index in the ROI for faster retrieval at view time, while increasing generation time and the size of the ROI. SearchWithIndex is most efficient on fields that are subject to frequent searches.

There are two ways to set a value for the SearchTag property. You can set a static value on the Properties page or you can use the SetSearchTag( ) method for a control to set the value dynamically. This section discusses how to set SearchTag on the Properties page. For information about using SetSearchTag( ), see Class AcReportComponent in *Programming with Actuate Foundation Classes.*

Searching for multiple instances of the same class

If a report design contains multiple instances of the same component, you can use the SearchTag property to uniquely identify those components for a report search. For example, a report can have three nested controls, each named CustomerTotal. One control displays the customer's total for a specific order, another shows the total amount for all the customer's orders, and another shows the customer total for each sales representative. To uniquely identify each CustomerTotal control for a search, set a unique SearchTag property for each control.

How to provide search functionality for multiple instances of a class

The following procedure shows how to search an ROI in the view perspective. You can run the same search for a report on the web.

1. In the report design, set the SearchTag property for each instance of the class that you want to use in a search. For example, set SearchTag to CustomerTotals for the controls that specify the total amount of an invoice, the total sales for a customer, and the total sales for each sales representative.
2. Set the Searchable property of these controls to SearchNoIndex or SearchWithIndex.
3. Run the report, choosing OK on Requester. The report instance appears.
4. Choose View➛Search. Search appears.
5. Choose Select. Select appears.

6 In the report, select one of the components that has the SearchTag property set to the value that you specified in the design.

7 Choose Add. The search tag name appears in Select.

8 Run the search. The search returns values for all the specified components.

Searching for multiple classes

You can use SearchTag to design a search that returns values for multiple classes when the user searches for any one of the classes. For example, you can create a search tag called RepInfo and assign it to the classes that display a sales representative's name, office ID, and total sales. When the user chooses one of these components in the report instance, the search returns values for all of them.

How to group search results for multiple classes

The following procedure shows how to search a report in the view perspective. You can run the same search for a report on the web.

1 In the report design, set the SearchTag property for the classes that you want to group in a search. For example, set SearchTag to RepInfo for the controls that specify sales representatives' names, office IDs, and sales totals.

2 Set the Searchable property of these controls to SearchNoIndex or SearchWithIndex.

3 Run the report, choosing OK on Requester. The report appears.

4 Choose View→Search. Search appears.

5 Choose Select.

6 In the report instance, select one of the components that has the SearchTag property set to the value that you specified in the design. For example, select a sales representative's name.

7 Choose Add. The name of the search tag appears in Select, as shown in Figure 14-3.

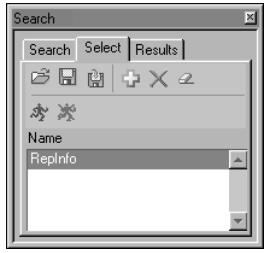


Figure 14-3 Search—Select

8 Choose Run. The values for all controls that have a RepInfo search tag appear in Results, as shown in Figure 14-4.

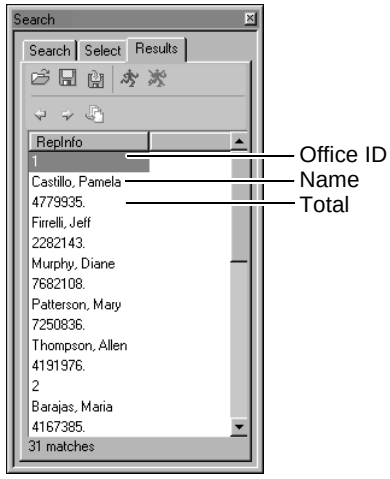


Figure 14-4 Search—Results

Retrieving data from controls in multiple parallel or sequential sections

The default search behavior for reports that contain parallel or sequential sections is to return values for only one section at a time. For example, in a report that displays the hire dates for employees in one section and the dates of salary increases in another section, the report user must search for each date class separately in the report instance. You can change this behavior by setting SearchTag to the same value for the date class in each section. When the report user selects one of the classes in the report instance, the search retrieves all values for both classes.

In a report that uses parallel sections, the first value in the search window is the first value from the parallel section the user selected. The next value is the first value from the next parallel section, and so on. Figure 14-5 shows the results of a search in which the EmployeeName class and the SalaryHistory class have the same SearchTag property. These classes appear in separate parallel sections in the report. The EmployeeName class contains one value for each employee. The SalaryHistory class can contain multiple values for each employee, to show salary changes over time.

In a report that uses sequential sections, the values from the section the user selects appear first, followed by the values from the next section, and so on.

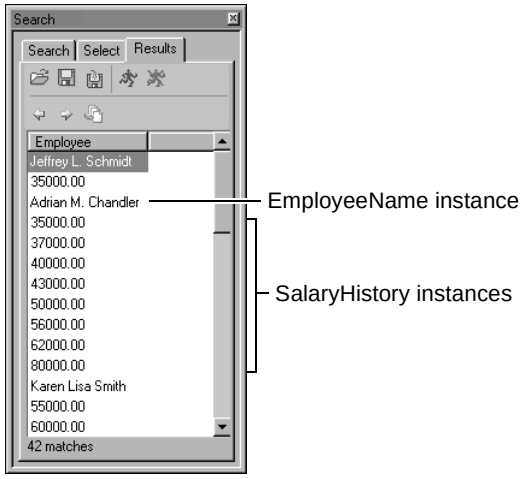


Figure 14-5 Search results of classes in parallel sections

About page headers and page footers in searches

A user can search controls in page headers and page footers. If the page header or page footer content changes from page to page, e.Report Designer Professional uses values from the first page header or page footer. For example, in a dictionary, the alphabetical range that appears in the page header changes for each page.

For the report section, if the Pagination➛ShowHeaderOnFirst property value is NoHeaderOnFirst, which is the default setting, a page header does not appear for a report or group section unless it extends beyond a page break. e.Report Designer Professional uses the invisible page header for that section. The ShowFooterOnLast property determines the visibility of page footers. In searches, page headers and page footers behave similarly.

Hyperlinks in the search results exhibit the following behavior:

- Hyperlinks link to only the first instance of the page header or page footer.
- In reports that use page-level security, the hyperlink goes to the first instance of the page header or page footer to which the user has access.
- If a search result links to the location of an invisible page header, the hyperlink goes to the beginning of the report or group section that contains the invisible page header.
- If a search result links to the location of an invisible page footer, the hyperlink goes to the beginning of the report or group section that contains the invisible page footer.

Designing searchable reports

The following strategies support successful searching:

- Place searchable controls that the user cannot combine in a search on different pages of the report document where the user is less likely try to combine them in a search. The search generates an error message when the user asks the search analysis to draw an impossible relationship across sections. State in the report that a search can only include controls on the same page.
- Include helpful information in the report to prevent users experiencing problems searching.

  For example, include suggestions for searches or display a button to hyperlink to your own Search Help to alert users that they cannot include controls in two nearby sections. If two parallel sections appear next to each other on the same page of the report, they can appear to be conceptually related.
- Always set SearchTag for dynamically created components if they are searchable and can appear in multiple places or multiple times in the same context.
- Do not set SearchTag for references because the scoped class name uniquely identifies references.
- If you set the SearchTag value programmatically, you must set it before calling the parent class' Start( ) method, Super::Start( ). You can set the value in the class's Start( ) method or elsewhere in the report design before Start( ) executes.
- For experienced users, the report's table of contents contains some visual clues about the structure of the report document. Depending on the level of experience of your report users, complex reports can require that you include additional report-specific Search Help.

About search analysis

Before a report search begins, e.Report Designer Professional analyzes the control objects the user adds to the search. Search analysis attempts to ensure that search results relate to each other. Using the default settings, the search analysis includes only those control objects that exist in related structures in the report design.

Report designers can use the SearchTag property to override the default search behavior settings.

Search works with controls that Actuate's search mechanism can organize into a flat, two-dimensional table, such as a spreadsheet. If the analysis of the controls that a user chooses for a search contains a hierarchy that the search mechanism cannot organize into a flat, two-dimensional table, the search cannot continue.

If a user viewing a report on the web adds multiple controls to the search conditions, then chooses Search Now, the search analysis maps the path to each

chosen control in the report structure. If the paths differ, the analysis compares the path differences to determine whether those differences show one-to-one or one-to-many relationships among the control objects. The search mechanism can organize both one-to-one and one-to-many relationships into a flat, two-dimensional table.

The analysis of the paths results in one of the following outcomes:

- If one-to-one or one-to-many relationships exist among the control objects, the search works and search results appear.
- If the path differences indicate that the search results in many-to-many relationships in the search results or if the paths indicate ambiguity, an error message appears. e.Report Designer Professional does not support searches that result in many-to-many relationships or ambiguous relationships in the search results. The search mechanism cannot organize those relationships into a flat, two-dimensional table.

When you use the view perspective to search reports, search analysis maps the paths to each control in the report structure when you add a control to a search. If the control does not match a supported relationship, an error message appears.

For a successful search, structurally related sections must contain all control objects the user adds to the search. Examples of relationships in report designs that support searching include:

- A report has a one-to-one relationship with a report section it contains.
- A report section has a one-to-one relationship with each of the components in its Before, After, PageHeader, and PageFooter slots.
- A report section and a group section have one-to-many relationships with the components in a Content slot subordinate to them in the hierarchy.
- A frame has a one-to-one relationship with each component contained in it, except when the controls are dynamically created. In that case, a frame can have a one-to-many relationship with the control. A search on a dynamic control matches only one instance of the control.
- A parallel section and a sequential section have one-to-one relationships with each component of each Content slot subordinate to them in the hierarchy.
- A control in an After slot that has only one value, such as an account total for a group section, has a one-to-one relationship with any other control in the report in a Before, After, PageHeader, or PageFooter slot that also has only one value. The mapped paths to these slots in the report structure do not prevent adding these unique controls to a search.

Examples of relationships in report designs that prevent searching are:

- Parallel and sequential report sections, which have many-to-many relationships among them. For a search, you cannot combine controls from

two parallel or sequential sections, except the controls in Before, After, PageHeader, or PageFooter slots that have only one value.

- Different Content slots, which have many-to-many relationships among them, if they contain multiple controls that have more than a single value.
- Page decorations and data controls, which are not related. Page decorations appear outside the flow in the page. Page decorations appear subordinate to the PageStyle and PageList slots in Report Structure. In a search, you cannot mix page decorations, such as a DateTime control, with data controls.

How to recognize relationships that support searches

To determine which structures in the report design result in one-to-one and one-to-many relationships, examine the report design's structure pane. Expanding and collapsing the structure indicates the relationships among the components.

1 Open <eRDPro_HOME>\Examples\DesignAndLayout\Detail\Detail.rod. Report Structure shows the first level in the report with + next to a Content—SalesDetailReport section, as shown in Figure 14-6.

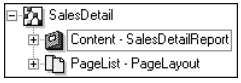


Figure 14-6 First level of a report structure

The SalesDetailReport section has a one-to-one relationship with the SalesDetail report that contains it.

2 Expand SalesDetailReport to display the next level in the report as is shown in Figure 14-7.

The SalesDetailReport section has a one-to-one relationship with ReportTitle, ReportPageHeader, and ReportTotals, which are located in Before, PageHeader, and After slots.

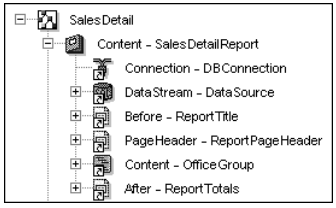


Figure 14-7 Report file structure

A relationship that supports searching exists between components in Report Structure that can expand from or collapse into another component.

3 Expand the OfficeGroup section as is shown in Figure 14-8. The SalesDetailReport section has a one-to-many relationship with the contents of

the OfficeGroup group section, because the group section contains many controls that can each have many values.

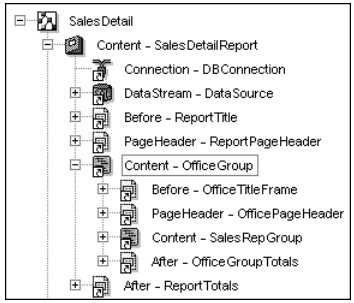


Figure 14-8 Detail file structure of OfficeGroup

4 Close Detail.rod.

5 Open <eRDPro_HOME>\Examples\DesignAndLayout\Forecast \Forecast.rod.

6 Expand the ForecastReport sequential section as is shown in Figure 14-9. Each of the report sections the ForecastReport sequential section contains has a one-to-one relationship with the ForecastReport section. The ForecastApp report has a one-to-one relationship with the contents of the ForecastReport sequential report section.

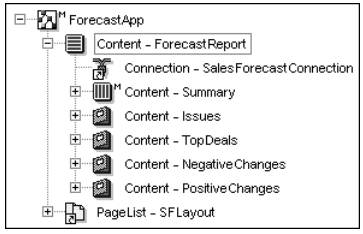


Figure 14-9 ForecastReport structure

The Issues, TopDeals, NegativeChanges, and PositiveChanges report sections and the Summary parallel section do not expand from or collapse into each other. Most of the components of these sections cannot combine for a search. You can combine controls that have only one value from the Before, After, PageHeader, or PageFooter slots in each of these report sections.

7 Expand the following sections as shown in Figure 14-10:

- Issues
- IssuesReportBodyContent
- Top Deals
- TopDealsReportContent

The components of Issues and TopDeals create many-to-many relationships among them. The search functionality does not support many-to-many relationships.

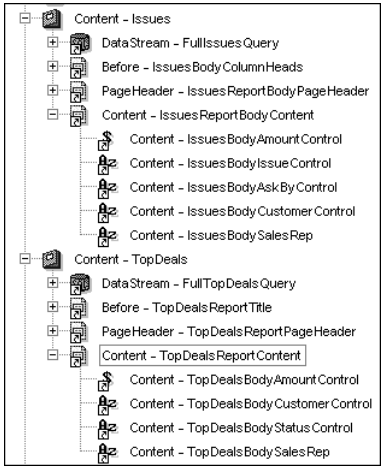


Figure 14-10 Expanding content sections

8 Close Forecast.rod.

Searching reports with parallel and sequential sections

This section illustrates how the proximity of parallel and sequential report sections can lead a user to think that the searchable control objects should result in a successful search.

For example, you create a report of sales history that contains sequential sections including quarterly forecasts for an office in one section and customers' credit ranks sorted by salesreps in another section. Figure 14-11 shows an example of such a report.

In this report document, you can choose to combine the Sum from the Sales Forecast for Boston and the Number of customers for sales representative Diane Murphy. Both controls appear in After sections and each contains only one value.

Because the sequential sections appear close to each other on the same page of the report, a user viewing the DHTML report document might think that the information is related and choose controls from both sections for a search. When the user chooses Search Now, an error message appears and says that the search cannot execute using the current criteria.

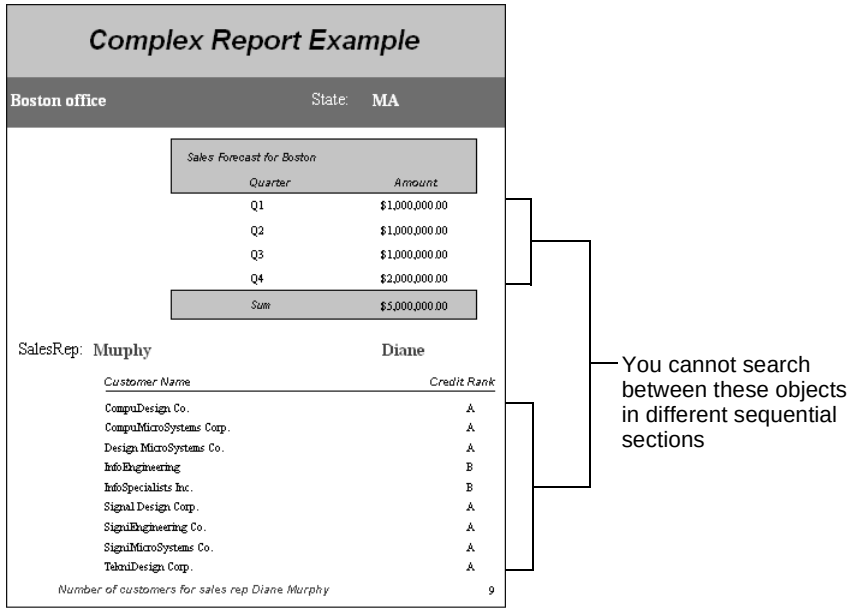


Figure 14-11 Complex report showing sequential sections

To prevent the error message, the report designer can:

- Change the design to avoid sequential or parallel sections. The user can add report sections to a search if there is a one-to-one or one-to-many relationship among them.
- Change the page layout to display only controls from the Sales Forecast for Boston table or only controls from the Customer Name list on a single page.
- Provide the suggestion to include only controls from the Sales Forecast for Boston table or only controls from the Customer Name list for a search.
- Change Searchable to NotSearchable for control objects in one sequential section.
- Use SearchTag to identify controls to support searches that Actuate search does not otherwise support in complex reports.

Developing hyperlink functionality

To define a hyperlink in Actuate e.Report Designer Professional, use a component's LinkExp property. In LinkExp, you type an extended Universal Resource Locator (URL) that identifies the link's destination report and defines the hyperlink's target in the destination report. Because LinkExp is an expression property, you can build expressions that resolve to specific values when the

report user activates the link. The expression can include references to columns in a data row.

About LinkExp syntax

The syntax for the LinkExp URL is:

```
<Destination URL>[#<Search expression>][?<Dynamic report
   parameters>]
```

where

- <Destination URL> is a required element that points to the destination. The destination URL can be relative to the current environment or it can be on the web.
- <Search expression> defines the exact location of the component in the destination report.
- ? indicates the beginning of the dynamic parameters section.
- <Dynamic report parameters> are parameters the user can enter when running the search.

You must specify elements in the order shown. Otherwise, the syntax is flexible. Depending on the purpose of the hyperlink, you can omit parts of an element, omit an entire element, or combine elements in various ways.

LinkExp is an expression property. You must enclose the entire link expression in quotation marks. To refer to specific columns in a report's data rows, concatenate the columns to the link expression string using an ampersand (&) after the expression's closing quotation mark.

Specific column values also must be enclosed in quotation marks if they contain spaces at the beginning, spaces at the end, literal quotes, or relational expressions.

To include a quotation mark (") in a link expression, you must use a set of two quotation marks. For example, the following LinkExp expression specifies a jump to the detailed information for the current office:

```
"FORECAST.ROI#ForecastDetail::OfficeTitleFrame::OfficeName=
   """& DataValue &"""
```

In this example, DataValue is a relational expression that evaluates to the column name. Actuate e.Report Designer Professional replaces DataValue with the value that the column contains when the user clicks on that column.

How to build a hyperlink expression using LinkExp

To create a simple link using LinkExp, enclose the link expression in quotation marks when you define the expression. If you include a reference to a specific column or value in a data row, concatenate the reference using &. To include

literal double quotation marks in an expression, use two sets of double quotation marks.

1 Start with the desired end result, the link expression that you want to build:

```
FORECAST.ROI#ForecastDetail::OfficeTitleFrame::OfficeName=
   "<office>"
```

where <office> is the name of the office for which you want information.

2 Enclose the entire link expression in quotation marks:

```
"FORECAST.ROI#ForecastDetail::OfficeTitleFrame::OfficeName=
   "<office>""
```

where <office> is a placeholder for a column name or data control.

3 To include the current value of DataValue in the hyperlink at run time, replace <office> with &DataValue:

```
"FORECAST.ROI#ForecastDetail::OfficeTitleFrame::OfficeName=
   "&DataValue""
```

When the report user chooses the office name that appears in the generated report document, Actuate e.Report Designer Professional replaces the placeholder DataValue with the current value of the column or data control. In the preceding example, the column contains an office name.

4 Because the value from the column or data control can contain spaces at the beginning or end, quotation marks, or relational expressions, enclose the value in two sets of quotation marks:

```
"FORECAST.ROI#ForecastDetail::OfficeTitleFrame::OfficeName=
   """&DataValue""""
```

The following sections describe the elements of the LinkExp syntax in more detail.

About destination URL syntax

The destination URL indicates the location of the destination report. If you do not specify a destination URL, Actuate e.Report Designer Professional assumes the location is internal to the current report.

The destination syntax is:

```
[<protocol>:]//<path>/<filename>
```

where

- <protocol> is the communication protocol to use to reach the destination report. Supported protocols appear Table 14-2.

Table 14-2 Supported protocols

Protocol	Description
file	The destination is in a file system: `file://C:/Forecasts/Boston/Q3forecast.roi`
http	The destination is on the web or in an Encyclopedia volume. This protocol goes to the browser: `http://Condor:8700//iportal/iportal /activePortal/viewer/viewframeset.jsp? name=/Revenue/Q1revenue.roi`

none	The result depends on the environment. If the source and destination are on the same file system, the destination appears in the view perspective. If the destination is on the web, the destination appears in the browser.
	The following example specifies a destination report in the folder Boston, one level up from the current report's folder:
	`…/Boston/Forecast.roi`
	The following example specifies a destination report in the current folder:
	`Q3salesdetails.roi`

other	The destination appears in the browser. Other protocols include gopher and FTP.

- <path>/<filename> is the destination report's path name. A path can be absolute or relative. You can specify a hyperlink to a report object instance (.roi) file. If the hyperlink runs to a dynamic report, you can link to a report object executable (.rox) file or a report object values (.rov) file. The original and destination reports must be in the same file system, on the same server, or on the web.

Actuate iServer does not support relative paths for temporary reports. To support viewing a temporary report, use an absolute path as shown in the following example:

```
/Folder/Report.roi.
```

Alternatively, Actuate iServer supports ~ syntax for an absolute path, as shown in the following examples:

- To display Myreport.roi in the current user's home folder, set LinkExp to:

```
~/Myreport.roi
```

- To display Myreport.roi in user1's home folder, set LinkExp to:

```
~user1/Myreport.roi
```

Table 14-3 describes examples of several destination URLs.

Table 14-3 Example destination URLs

Destination URL path	Description
`"../Boston/Forecast.roi"`	For a report saved in the Encyclopedia volume, indicates that the destination report is in the Boston folder, one level up from the current report's folder
`"Q3salesdetails.roi"`	Indicates that the destination report is in the current report's folder
`"file://C:/Forecasts/Boston /Q3forecast.roi"`	Indicates that the destination report is in the file system, in the Boston subdirectory of the Forecasts folder on drive C
`"http://gumby:8700/iportal /iportal/activePortal/viewer /viewframeset.jsp?name= /Revenue/Q1revenue.roi"`	Indicates that the hyperlink is to a file on the web server
`"rotp://gumby/salesserver /forecast.rox"`	Indicates that the destination report is on Actuate iServer

About search expression syntax

The search expression defines the hyperlink's specific destination in the destination report. To display the first page of the destination report, specify only the report URL. You do not need a search expression to display the first page.

You can use one of the following options to identify a component in the search expression:

- The fully qualified name of the component. Actuate software uses the structure of the report to find a component.
- The SearchTag property value. If you set a value for the SearchTag property, you cannot use the fully qualified name of a component class to locate the component.

Using the fully qualified name of a component in a search expression

The search expression syntax using the fully qualified component name is:

```
#<Class[::class…]> [[.<Variable>]=<Value>] [;<…>]
```

where

- # indicates the beginning of the search expression.
- <Class[::class…]> is the fully qualified name of a component class, such as a control class.
- <Variable> is the class variable on which to search. If you do not specify a variable name, the search is on the default value for the specified class. If you do not specify a value for the class, the search looks for all instances of the class in the report.
- <Value> is the value on which to search. You must enclose the value in quotation marks when the value has any of the following characteristics:
  - Includes one or more spaces at the beginning or end
  - Contains a relational expression, such as <, >, or =
  - Contains a special character, such as &, ;, @, or #

For example, to search for the literal string “Gumby Special Pack”, use the following syntax:

```
"#ForecastApp::ProductName=""Gumby Special Pack"""
```

To create a hyperlink to the ForecastApp report’s product name, use the following syntax:

```
"#ForecastApp::ProductName=" & [values.ProdName]
```

where

- The product name control is ProductName.
- ProdName’s ValueExp is [values.ProdName].

Adding search conditions to a fully qualified component name

You can use special characters to add search conditions to a fully qualified name. For example, you can specify several conditions and separate them by semicolons. Do not include spaces after the semicolons. Specifying multiple search conditions is like constructing a Boolean AND statement. Controls or data must match all the search conditions to be found by the hyperlink. If the controls or data do not match, the hyperlink does not work. If more than one match exists, the hyperlink goes to the first occurrence.

For example, to search for customers in the NorthEast sales region who have a credit ranking of A, add the credit rank criteria to the name of the component for which you are searching, using the following syntax:

```
"#ForecastApp::CustInfo.Region=NE;CustInfo.CreditRank=A"
```

To search for the Positive Changes report in the Q3Sales report, use the following syntax:

```
"Q3Sales.roi#RevenueDetails::PositiveChanges"
```

To search for any instance of customer ID in a CustomerNameControl in the current report, use the following syntax:

```
"#ForecastApp::CustomerNameControl.CustomerID"
```

To search for customer ID 1562, use the following syntax:

```
"#DetailReport::CustomerFrame::CustomerID=1562"
```

To search for order number 1340, use the following syntax:

```
"#SalesDetail::OrderNumber=1340"
```

Using the SearchTag property value in a search expression

When you use the SearchTag property, the search expression syntax is:

```
#($<SearchTag>)=<Value>
```

where

- # indicates the beginning of the search expression.
- ($…) identifies the SearchTag.
- <SearchTag> is the value of the SearchTag property that you set in Properties.
- <Value> is the data value on which to search. If you do not specify a value, the search finds the first instance of the specified SearchTag value in the report. You must enclose the value in quotation marks when the value has any of the following characteristics:
  - The value includes one or more spaces at the beginning or end.
  - The value contains a relational expression, such as <, >, or =.
  - The value contains a special character, such as &, ;, @, or #.

To search for order number 1340, use the following syntax in the search expression:

```
"#($Search_for_OrderNumber)=1340"
```

where Search_for_OrderNumber is the SearchTag value that you set in Properties.

About dynamic report parameters syntax

You can create a new dynamic report object instance (.roi) file from the current ROI, using LinkExp to pass specific parameters to the report object executable (.rox) or report object value (.rov) file. You can specify whether to overwrite any previous version of the report or create a new version of the report. You can also search in the dynamic report after you generate it.

Creating dynamic reports is useful for short-run reports, those reports that are of interest right now but that are not needed long-term. For example, you can define a dynamic report to retrieve the details for positive deals on a specific order number, based on the data in the current report. Users can examine the positive deals details, then throw the report away.

The dynamic report parameters syntax is:

```
?<param>=<value>;<param>=<value>[;<…>]
```

where

- ? indicates the beginning of the dynamic report parameters section.
- <Param> is the name of a parameter in the report or the keyword _ _overwrite.
- <Value> is the parameter value. When the parameter is the keyword _ _overwrite, the parameter value is

  ```
  old | new
  ```

  Old and new are case-sensitive. You must specify these values in lowercase. Old causes the generated report to overwrite any previous version. New creates a new version of the generated report. The default setting is to create a new version.
- [;<…>] and other special characters specify additional parameter value pairs, separated by semicolons.

For example, to examine the details for all customers in the Forecasts report who are located in the Northwest region and who have a credit ranking of C, use the following syntax:

```
"FORECAST.ROX?Region=NW;CreditRank=C"
```

For more information about specifying a value for an ad hoc parameter and using Query by Example (QBE) syntax in an expression, see *Using Information Console.*

To examine the same details for an ROX that generates a report to display on the web, use the following syntax:

```
"rotp://Gumby/ToySales/Forecasts FORECAST.ROX?
   Region=NW;CreditRank=C"
```

To generate the details report and overwrite any existing version of the report, use the following syntax:

```
"FORECAST.ROX?Region=NW;CreditRank=C; _ _overwrite=old"
```

How to create a hyperlink that runs a report using a parameter value

The following procedure describes how to create a hyperlink that generates a dynamic report. In this procedure, the link expression generates a report that contains customers who have a credit rank of C.

1 Start two e.Report Designer Professional instances, one to display the source report and the other to display the destination report.

2 Open the report object design (.rod) file for the destination report.

3 Choose Tools➛Parameters. Parameter Editor appears, as shown in Figure 14-12.

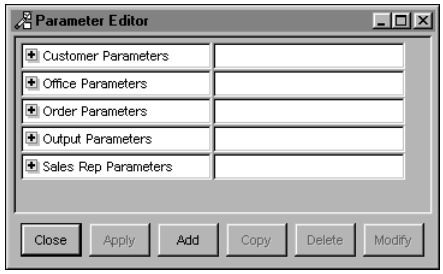


Figure 14-12 Parameter Editor

4 Expand a parameter group, such as Customer Parameters.

5 Select a parameter, such as customers_creditrank. Choose Modify. Parameter Properties—General appears, as shown in Figure 14-13.

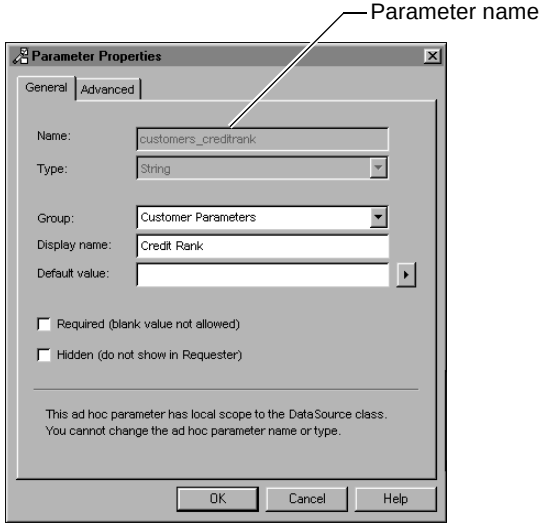


Figure 14-13 Parameter Properties

6 In Parameter Properties—General, record the parameter name. Choose OK.

7 To create the destination report executable file, run the report. The destination report appears in the view perspective.

8 Close the destination report and the design file.

9 To create the hyperlink in the source report:

1. Open the source report design file.
2. Add a label control in which to create the hyperlink.
3. Right-click the new label control, then choose Properties. The Properties page for the control appears.
4. In Linking➛LinkExp, type the following hyperlink expression:

   ```
   "<report name>.rox?<parameter_name>=<value>"
   ```

   where

   - <report name> is the report name.
   - ? indicates the beginning of the dynamic report parameters section.
   - <parameter_name> is the name of the parameter, taken from Parameter Properties—General.
   - <value> is the parameter value.

   For example, to generate a report called Detail, displaying customers with a credit rank of C, type:

   ```
   "Detail.rox?customers_creditrank=C"
   ```

10. Run the report. The source report appears.
11. To generate and link to the destination report, double-click the label control that you created in the source report.

Creating a hyperlink that uses a dynamic parameter value

To create a hyperlink that generates a destination report using a parameter value from a data control in a source report, complete the following tasks:

- Create a destination report design that contains an ad hoc parameter value.
- In a source report design, create a data control that contains a link expression. The link expression passes a parameter value to the Factory process to generate a destination report.

Use the following syntax to create a hyperlink that uses a parameter value from a data control in a source report to generate a destination report:

```
"<report name>.rox?<param>='" & [<table_column name>] & "' "
```

where

- <report name> is the name of the destination report.
- <param> is the name of a parameter in the destination report.
- <table_column name> is the name of a database column that appears in a control in the source report. The <table_column name> value of that control

replaces the [<table_column name>] variable in the link expression for that control in the generated source report document.

When a user chooses the linked data control in the source report document, that data control's value passes to the destination report as a parameter value. The destination report runs, using the value from the source report, and the specified data appears in the destination report.

For example, a destination report, Report_2.rod, contains a text control that displays the database value customers.phone. The report also contains an ad hoc parameter, customers_phone. In Requester, you can specify a value for customers_phone when you run the report. You can also use a link expression to provide a dynamic parameter value from another report document to generate Report_2.roi.

In Figure 14-14 Report_1.rod contains a text control that displays the database value customers.phone. The text control's Linking➤LinkExp property value is:

```
"Report_2.rox?customers_phone='" & [customers_phone] & "' "
```

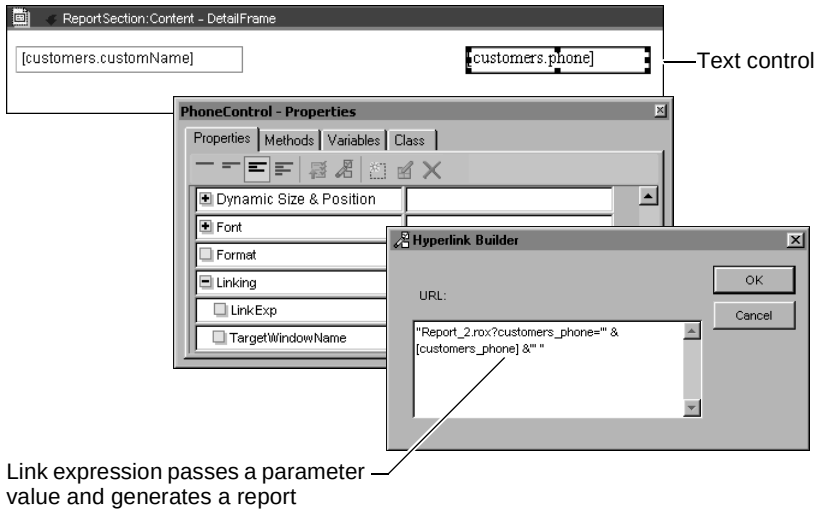


Figure 14-14 Creating a linked text control

Because the customers.phone database column contains hyphens in its string values, quotation marks enclose single quotes in the LinkExp value.

Creating this link expression requires:

- Enclosing the following string in quotation marks to include the single quote in the string:

  ```
  Report_2.rox?customers_phone='
  ```

- Concatenating the parameter name using the following syntax:

  ```
  & [customers_phone] &
  ```

- Enclosing the final single quote in quotation marks to pass this single quote to the Factory process.

The completed LinkExp property value is:

```
"Report_2.rox?customers_phone='" & [customers_phone] & "' "
```

If you hover the mouse pointer over the text control in the view perspective, the status bar displays the link expression, including the single quotes that must enclose the telephone number as indicated in Figure 14-15.

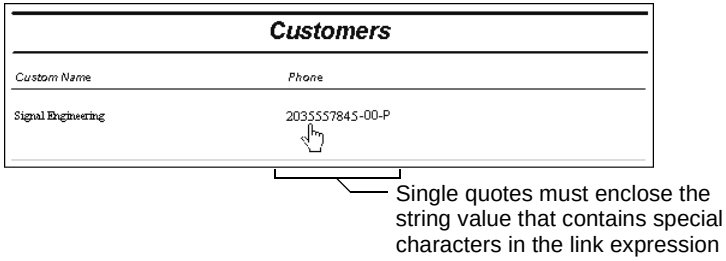


Figure 14-15 Displaying a linked expression

Double-clicking a linked data control in the source report generates a destination report using the parameter value from the data control.

Calling JavaScript functions with LinkExp

You can use the LinkExp property to call a JavaScript function. Use the following syntax for the LinkExp property value:

```
javascript:function(parameters)
```

For example:

```
javascript:history.back()
```

The parentheses pass to the web browser. If the function name or the parameters contain parentheses or certain other characters, however, these characters are escaped. To pass these characters to a function, you must unescape them.

Defining universal hyperlinks

Defining universal hyperlinks, or hyperlinks that work in a variety of environments, supports viewing reports in various environments. Suppose users run reports on a variety of systems and view reports on the web, as well as locally. For example, reports can be in the file system or an Encyclopedia volume folder, with users viewing them in the view perspective, and reports can be in a

directory published by a web server, with users viewing them using a web browser.

To define universal hyperlinks do the following:

- Ensure that the source and destination reports are in the same environment: they are both in the file system, on an Encyclopedia volume, or on a web server.
- Ensure that the reports' locations are fixed and are simple. They should be in the same directory or folder or in the same directory or folder structure.
- Use a consistent directory or folder structure for each environment. For example, put all the forecast reports in a structure similar to \Forecasts\Region\Q2.
- Use relative links to locate the destination report. You can specify locations such as \Q3\Report.roi to reach the destination report.

Designing a hyperlink URL

You can design an Actuate Information Console URL that links to a report, where the URL is external to the report.

The syntax for this Actuate Information Console hyperlink URL is

```
http://<app server>:<port>/iportal/<path><page>.jsp?
   <parameter=value>
```

where

- <app server> is the name of the machine running the application server or JSP/servlet engine.
- <port> is the port on which you access the application server or JSP/servlet engine.
- iportal is the default context root for accessing the Actuate Information Console JSPs.
- <path> is the directory containing the JSP to invoke.
- <page> is the name of the JSP.
- <parameter=value> specifies the parameters and values that the JSP requires.

For example, to view a report document, Actuate Information Console uses a URL with the following format:

```
http://druid:8700/iportal/iportal/activePortal/viewer
   /viewframeset.jsp?name=/Test%20Folder/condtnl2.roi
   &format=DHTML
```

where

- viewer is the directory containing JSPs that provide viewer functionality.
- viewframeset.jsp is the JSP that displays the report with the navigation bar.
- name=/Test%20Folder/condtnl2.roi specifies the path and name of the report object instance (.roi) file.
- format=DHTML specifies the viewing format.

To be included in a URL, special characters in the search conditions, such as spaces, must be encoded. For example, the encoding for a space is %20. Encoding is not necessary if you use the LinkExp syntax to link among reports and report locations.

For more information about Actuate Information Console, see *Information Console Developer Guide.*

Developing a table of contents

The table of contents provides an effective way for a report user to learn about a report's organization and easily navigate to grouped information in reports. The table of contents and search provide complementary navigational tools. After navigating to the information group, the report user can browse forward or use the search feature to find detailed information.

The table of contents should be of a reasonable size. A large table of contents can increase report-generation time and cause usability issues.

About table of contents properties

Actuate e.Report Designer Professional table of contents properties default values are set to include the levels of information needed in the table of contents. The values for the TOC➤TocAddComponent and TOC➤TocAddContents properties determine whether the components and subcomponents appear in the table of contents. The information groups that should appear in the table of contents are usually represented by section components. The default values for TocAddComponent and TocAddContents on lower-level components such as frames and controls suppress table of contents entries. Usually these components are not presented in the report's table of contents.

Developing a table of contents for a grouped report

Table of contents entries for group sections are generated because the default value for the TOC➤TocAddComponent property is TOCIfAllVisible. By setting TocAddComponent to TOCIfAllVisible, one entry for the group section and one entry for each unique value of the group section's key property appear in the

table of contents only if the user has access to all of the pages produced by the group section. To disregard any access considerations, set TocAddComponent to TOCIfAnyVisible.

If the number of detail rows generated for the group section is large, break up the group section into smaller group sections to limit the number of table of contents entries.

Figure 14-16 shows the structure of the Detail report in the <eRDPro_HOME>\Examples\DesignAndLayout\Detail folder. Instead of having a single group section with a key consisting of multiple columns, this report has four nested group sections. Each nested group section has a single column sort key. The groups are as follows:

- OfficeGroup sorted by office ID
- SalesRepGroup sorted by sales representative's last name
- CustomerGroup sorted by customer name
- OrderGroup sorted by order ID

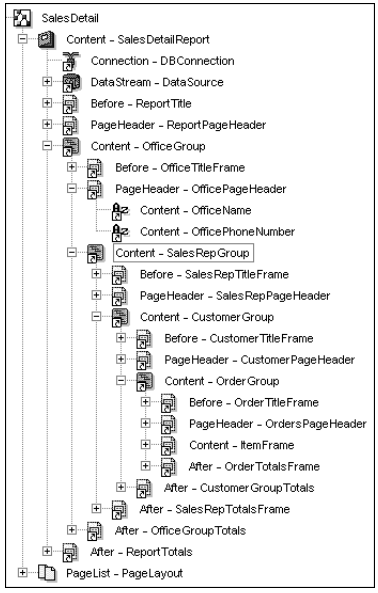


Figure 14-16 File structure of the Detail report

Figure 14-17 shows the default settings for the TOC properties group for the group section SalesRepGroup. To display the TOC properties, choose Advanced Properties on the Properties page.

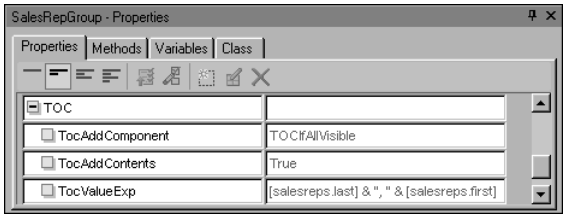


Figure 14-17 Default settings for TOC properties group

The table of contents built from the nested group sections appears in Figure 14-18. The table of contents allows for easy navigation because a manageable number of entries are present at each level of the table of contents.

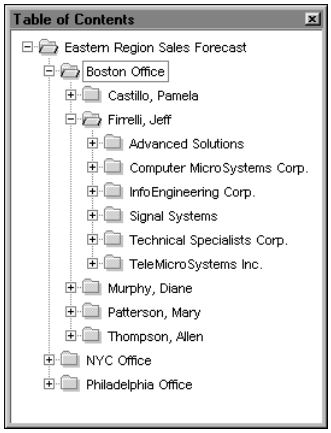


Figure 14-18 Detail of a TOC

Developing a table of contents for other reports

Report section components automatically generate a single table of contents entry, because e.Report Designer Professional's default settings for TOC➤TocAddComponent is TOCIfAllVisible and TOC➤TocAddContents is True.

The Forecast report's table of contents contains only one entry for each of the report sections in the report design, because the TOC➤TocAddContents property is set to False for the Issues, TopDeals, Negative Changes, and Positive Changes reports. For the Negative Changes report, Figure 14-19 shows the TocAddContents property set to False. The Forecast report is in the <eRDPro_HOME>\Examples\DesignAndLayout\Forecast folder.

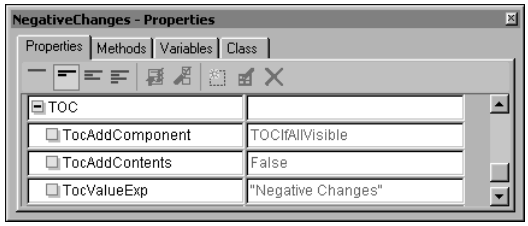


Figure 14-19 TocAddContents property

Figure 14-20 shows the Forecast report's table of contents, displaying only one entry for each of the report sections.

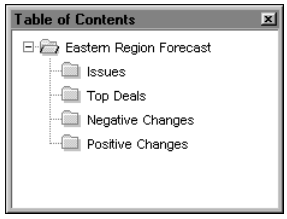


Figure 14-20 Forecast report's table of contents

Chapter

15

Designing a report for multiple platforms and environments

This chapter contains the following topics:

- Designing a report to display and print data across platforms
- Understanding font issues
- Designing a report for multiple viewing environments

Designing a report to display and print data across platforms

The Factory prints and displays a report to match your design specifications as closely as possible. Differences in appearance can nevertheless occur when report users print or view the report on platforms other than those that you use to develop the report. Although you cannot control the printing and viewing environments of report users, you can optimize the report's appearance across platforms by adopting the strategies this topic outlines.

The Factory uses font information to determine character widths and heights, which it uses to compute line breaks, text truncation, fill characters, and so on. The Factory requires font information for the following tasks:

- Displaying report output in DHTML format
- Displaying report output in PDF format
- Displaying report output in PowerPoint format
- Generating dynamic text control data
- Generating Excel data using AFC Excel API code in the report design

Actuate iServer also uses font information for displaying and printing reports. For more information about Actuate iServer's use of fonts, see *Configuring BIRT iServer.*

Understanding viewing and printing requirements

Viewing an Actuate report on a monitor requires the Windows display subsystem. The Windows system depends on:

- The capabilities of the graphics card the monitor uses
- The video driver
- Available Windows software fonts

If you require extremely accurate printed output, you must design for printing in a specific environment. Printing depends on the following technologies:

- The capabilities of the printer, such as color support and resolution in dots per inch (dpi)
- Available printer fonts
- Downloadable fonts
- The printer language
- The printer driver

Designing for the delivery environment

The appearance of the report online and in print depends on the hardware and software the report user uses. Because exact fidelity between printing and viewing is not always possible, decide whether printing or viewing is most important and design for that output. If possible, understand the capabilities of your report users' print and viewing equipment, including monitors, printers, and display and printer drivers.

Be aware that monitor resolution can affect how much of a report is visible on a monitor. In general, a monitor set to a lower resolution is capable of displaying less of a report than a monitor set to a higher resolution. For example, if you have a report with a wide or long format set at 100%, a monitor set to 640x480 resolution can display less of the report than a monitor with a 1024x768 resolution.

Allow for differences in text layout and font heights and widths on different platforms by enlarging controls that contain text. For example, the amount of space between lines on a multiline control differs depending on whether the report prints on a UNIX or Windows printer.

Designing for overlapping controls

Avoid defining BalloonHelp for overlapping controls. If you use BalloonHelp for overlapping controls, the report user sees the help string for each control in the view perspective. Other viewing platforms, such as a DHTML Viewer, are less predictable. A chart's BalloonHelp string does not appear in the view perspective but does appear in DHTML.

Designing for the print environment

Minimize the size and color depth of graphics so that they print clearly in black and white.

If report users view reports online and print them on UNIX PCL printers, minimize the use of charts. Charts do not print in UNIX PCL printers.

Understanding font issues

The Factory uses fonts and font information when generating and printing reports. When you design a report in e.Report Designer Professional, you can specify any combination of Font properties for a control. Users can see text in the fonts that you use only if their machines support your fonts. If the Factory cannot find your fonts when displaying the report in the view perspective, PDF Viewer, or DHTML Viewer, it substitutes fonts. For example, when a report user views a report object instance (.roi) file, any unavailable fonts map to the Windows fonts.

Similarly, font substitution occurs if you use a font that is unavailable in the user's print environment.

There are three ways to make fonts available to a report:

- Use the fonts that master_fonts.rox defines.
  Every report design can use the fonts this file makes available.
- Define fonts in a file named customized_fonts.rox.
  Create this file. Fonts must be available in your environment before you can define a customized font file. Then, register the custom file.
- Define fonts in the report object executable (.rox) file.
  A font you use in the report design is embedded in the ROX that generates from the design, except for fonts in charts and images and fonts the user sets dynamically at run time.

Depending on the environment in which the Factory is running, the Factory looks for font information in your report object executable (.rox) file, a custom font file, a master font file, or the system's operating system. The order in which the Factory searches for font information depends on the system and the system's configuration.

For information about how Actuate iServer searches for fonts, see *Configuring BIRT iServer.*

If Actuate iServer includes the Actuate Query option, Actuate Information Console users can create queries from information objects. You can use e.Report Designer Professional to specify the fonts that Actuate Query uses for displaying query output. For information about specifying fonts displayed in Actuate Query, see *Accessing Data using e.Report Designer Professional.*

Getting font information on Windows and UNIX servers

For rendering DHTML reports and PDF reports and for generating Excel and dynamic text control data, the Factory gets font metrics from the following sources:

- Font information defined in a custom font file stored on the server.
- Font information defined in a master font file stored on the server.
- Font information embedded in the report object executable (.rox) file.
- Default fonts for the controls.

Getting font information on desktop-only environments

If a report design contains dynamic text controls, the Factory gets font information for generating and rendering the contents of dynamic text controls from the following sources:

- Font metrics embedded in the report object executable (.rox) file
- Font metrics in the Windows operating system
- The default font for the control

If a report design uses the AFC Excel API, the Factory gets font metrics for generating the Excel data from the following sources:

- Font metrics embedded in the ROX
- Font metrics defined in a custom file stored on the desktop

The look-up order depends on how the desktop system configures the custom font file.

About the master fonts file

The master fonts file, master_fonts.rox, which ships with Actuate iServer, contains font information for many standard fonts. e.Report Designer Professional and Actuate iServer refer to this file to display and print reports.

Table 15-1 lists the fonts in master_fonts.rox.

Table 15-1 Actuate iServer list of fonts

Plain font	Bold font	Italic font	Bold Italic font
Arial	Arial Bold	Arial Italic	Arial Bold Italic
Arial Unicode MS	Arial Unicode MS Bold	Arial Unicode MS Italic	Arial Unicode MS Bold Italic
Batang	Batang Bold	Batang Italic	Batang Bold Italic
Batang 1	Batang Bold 1	Batang Italic 1	Batang Bold Italic 1
Courier New	Courier New Bold	Courier New Italic	Courier New Bold Italic
Ming LiU	Ming LiU Bold	Ming LiU Italic	Ming LiU Bold Italic
Ming LiU 1	Ming LiU Bold 1	Ming LiU Italic 1	Ming LiU Bold Italic 1
MS Gothic	MS Gothic Bold	MS Gothic Italic	MS Gothic Bold Italic

(continues)

Table 15-1 Actuate iServer list of fonts (continued)

Plain font	Bold font	Italic font	Bold Italic font
MS Gothic 1	MS Gothic Bold 1	MS Gothic Italic 1	MS Gothic Bold Italic 1
MS Mincho	MS Mincho Bold	MS Mincho Italic	MS Mincho Bold Italic
MS Mincho 1	MS Mincho Bold 1	MS Mincho Italic 1	MS Mincho Bold Italic 1
Sim Sun	Sim Sun Bold	Sim Sun Italic	Sim Sun Bold Italic
Sim Sun 1	Sim Sun Bold 1	Sim Sun Italic 1	Sim Sun Bold Italic 1
Symbol	Symbol Bold	Symbol Italic	Symbol Bold Italic
Times New Roman	Times New Roman Bold	Times New Roman Italic	Times New Roman Bold Italic

If you use a font that does not appear in master_fonts.rox, you can create a custom executable file that defines your fonts.

Embedding font information in a report object executable file

By default, Actuate e.Report Designer Professional embeds font information in the report object executable (.rox) file for every font that you use in controls, except in the following cases:

- Fonts in charts and image controls
- Fonts the user sets dynamically when running the report, either programmatically or using dynamic text controls

How to explicitly embed a font in a report object executable file

1. For each font you want to specify, add a label control to the report design.
2. Set the Font property of each label control to a different font. If your report design uses bold or italic versions of a font, you must include them separately. For example, create separate labels for Arial Bold and Arial.
3. Hide each label control by setting the Visibility property to False.

Defining custom fonts

As an alternative to using the master font file or explicitly embedding font information in a report object executable (.rox) file, you can define fonts in a custom font file. The advantage of using a custom file is that you can specify all

fonts your report object design (.rod) files use. A possible disadvantage is that this custom file can be very large, and can take longer to process.

To create a custom font file, complete the following tasks:

- Install the custom fonts on your machine.
  You must have all the fonts installed on your system for the file to compile with the correct font metrics. If you do not, the report compiles but the executable file does not have complete information for all fonts.
- Design a report that contains a label control for each font. Set Font properties for each control.
- Save the ROD as:

  `customized_fonts.rod`
- Compile the executable file.
- Save the executable file to Actuate iServer's \etc folder.
  If you work in a cluster environment, copy this file to every machine in the cluster.

For information about using a custom fonts file for reports that appear in multiple locales, see *Working in Multiple Locales using Actuate Basic Technology.*

How to define fonts in a custom font file

1. In Actuate e.Report Designer Professional, create a blank report with a sequential section:
   1. Choose File➛New.
   2. In Create New Report, select Blank Report. Choose OK. A blank report appears in the design perspective.
   3. In Report Structure, select the Content—ReportSection component. Press Delete.
   4. Drag a sequential section from Toolbox—Structure and drop it in Content.

      Content—SequentialSection appears in Report Structure as shown in Figure 15-1.

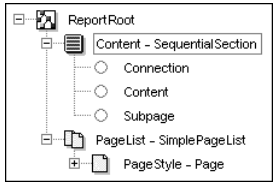


Figure 15-1 Content—SequentialSection

2 Drag a frame from Toolbox—Structure and drop it in the blank Content slot. Content—Frame1 appears.

3 To add fonts to the design:

   1 Drag a label control from Toolbox—Controls and drop it in the frame.

   2 Right-click the control, then choose Properties.

   3 In the Properties page, set the font properties.

   For example, create a label for Arial Unicode MS, as shown in Figure 15-2.

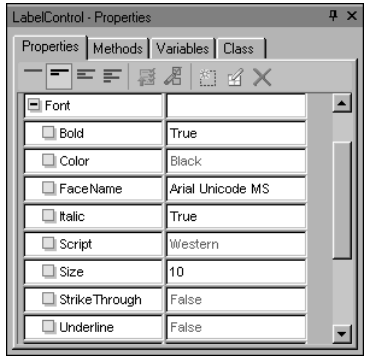


Figure 15-2 Setting font properties

4 Repeat steps 1 through 3 to add additional fonts. You must create a separate label for plain, bold, italic, and bold italic fonts of the same family. For example, Arial Bold is not the same as Arial.

Figure 15-3 shows the design for an example fonts file.

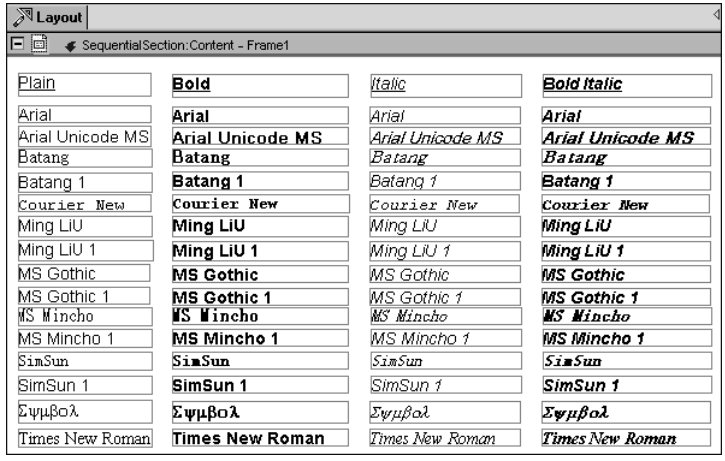


Figure 15-3 Fonts file

4 Save the report as:

```
customized_fonts.rod
```

5 Run the report. The font information appears in the view perspective.

To use the custom fonts that you define in this report, place the executable file, customized_fonts.rox, in Actuate iServer's \etc directory. In a cluster environment, place the executable file on every Actuate iServer node that has the View or Factory service enabled.

You must also register the custom font file to use it in a desktop environment.

Using a custom font file in a desktop environment

Actuate desktop software looks for font information in a custom font file if you configure the desktop software to use the file. To specify that the Factory use a custom font file, you must register the file by setting environment variables and specify how the Factory should use the file.

How to register a custom font file

1 Set the AC_MASTER_FONT_FULLPATH_NAME environment variable to the location in which the custom font file is saved. Use the full path name.

2 To specify how the Factory should use the custom font file, complete one of the following tasks:

- If you want the Factory to look for fonts in the custom font file before checking the report executable file, set the environment variable AC_USE_DEFAULT_FONT_FILE to:

  ```
  primary
  ```

- If you want the Factory to look for fonts in the executable file first, set the environment variable AC_USE_DEFAULT_FONT_FILE to:

  ```
  secondary
  ```

  For information about setting environment variables, see Windows Help.

Designing with fonts on multiple platforms

To ensure that the fonts in your report adhere to your design specifications when report users view or print the report in different environments, consider the following techniques:

- Work with standard fonts that users are likely to have on their systems. Distributing custom fonts to users is typically a costly option because of font copyrights and licenses.
- Minimize the number of fonts in the report.

- Use scalable fonts rather than the raster version of fonts. For example, on Windows, use Courier New, not Courier, as the monospaced font.
- Do not use fonts designed for use with English and other Western European languages, such as Arial and Times New Roman, on systems running Japanese, Chinese, or Korean locales. The PDF converter cannot recognize the characters so it uses a Latin character set. Acrobat Reader also cannot display the correct text. In Asian locales, use fonts designed for Japanese, Chinese, or Korean, such as MS Gothic, MS Song, and Gulim. For more information about designing reports for multiple locales, see *Working in Multiple Locales using Actuate Basic Technology.*
- For non-TrueType fonts, font substitution yields incorrect results when you use an invalid size, such as Courier 1. For this reason, a string might not appear correctly in the DHTML Viewer or in Acrobat Reader, even if the specified font is available in the viewing environment. The font must be available in the design environment or the font information in the executable file describes the substituted font.

If you set fonts dynamically for report generation or viewing, embed font information in the report object executable (.rox) file for all fonts a report user can select.

Designing a report for multiple viewing environments

Reports can be viewed in various types of viewing environments.

Displaying different controls in DHTML and PDF output

You can display different controls in a report's DHTML and PDF output by stacking one control on top of another in the report design and setting the ShowInDHTML property for one control and the ShowInPDF property for the other.

For example, you can display the output of custom browser code when the report appears in a web browser and display an image control when it prints in PDF format. Place a browser scripting control on top of an image control in the report design. For the browser scripting control, set ShowInDHTML to true and ShowInPDF to false. For the image control, set ShowInPDF to true and ShowInDHTML to false.

Setting whether a control appears while printing or viewing

To prevent a control from appearing in the report's PDF output, set the control's ShowWhenPrinting property to false. If ShowWhenPrinting is true, the setting of ShowInPDF determines whether the control appears in the PDF output. Likewise, if a control's ShowWhenViewing property is false, the control does not appear in the DHTML output. If ShowWhenViewing is true, the setting of the ShowInDHTML property determines if the control is displayed in the report's DHTML output. The default setting for a control's ShowWhenPrinting and ShowWhenViewing properties is True.

Chapter

16

Debugging a report

This chapter contains the following topics:

- About debugging
- Understanding the types of errors
- About e.Report Designer Professional's debugging tools
- Controlling code generation
- Starting a debugging session
- Examining and modifying variable values

About debugging

Debugging is the process of locating and eliminating errors in a program. Typically, debugging involves executing specific portions of a program and analyzing the operation of those portions. Actuate e.Report Designer Professional contains debugging tools to identify and fix compile-time and run-time errors in a report. Using e.Report Designer Professional, you can debug the following items in a report:

- An Actuate Basic source file (.bas) in the report design
- An Actuate Foundation Class method

Using the debugging tools in e.Report Designer Professional, you can perform the following tasks:

- Set a breakpoint to generate code up to a specified point, then stop and view the result.
- Watch a variable to check its values.
- Execute a method one line at a time or step over methods that you know are error-free.
- View the current execution stack.

e.Report Designer Professional also provides a tool to set parameters for successive debugging runs of a report.

The following sections describe the debugging tools and how to use them.

Understanding the types of errors

The first step in fixing bugs is understanding the following three kinds of errors your code can contain:

- Compilation, or syntax, errors
- Run-time errors
- Logic errors

About compilation errors

Compilation errors result from incorrectly constructed statements. Typical errors include mistyping a reserved word, omitting necessary punctuation, or failing to balance pairs of statements such as If and End If.

A compilation error can occur because of an error in an Actuate Basic statement, a custom Actuate Basic source (.bas) file, or the ValueExp property of a report

component. As a report generates, the Actuate Basic compiler detects compilation errors in Actuate Basic methods or Actuate Basic source (.bas) files and generates a list of error descriptions in the Output window.

About run-time errors

A report design can be free of compilation errors but still fail to generate a report because it contains run-time errors. These errors occur while the code executes. A run-time error occurs when e.Report Designer Professional tries to perform an impossible task, such as opening a file that does not exist or trying to divide a number by zero. Like a compilation error, a run-time error is easy to find and fix. When Actuate Basic encounters a run-time error, it stops report generation and displays the method containing the error. An arrow points to the line where the error occurs.

About logic errors

If your code is syntactically correct and runs without errors, it can still produce incorrect results. For example, you can find that your code calculates an incorrect value. These errors, called logic errors, occur if you use the wrong operator or function, forget to initialize a variable, or assign an incorrect value to a variable.

Actuate Basic cannot identify logic errors. Much of a report developer's debugging effort goes into solving logic errors. Debugging tools help you identify logic errors by monitoring the values of a report's variables and objects as the report generates.

About e.Report Designer Professional's debugging tools

Actuate e.Report Designer Professional provides several tools to support debugging. Many of them are available using the Debug menu. Also, when you run a report to a breakpoint, e.Report Designer Professional displays the debug perspective, which presents the principal debugging tools, as shown in Figure 16-1.

These tools include the following windows:

- Method editor, which displays the method or source file to debug
- Call Stack, which lists the stack of method calls
- Output, which presents error messages that help with debugging tasks
- Variables, which lists the variables available to the report in context
- Watch, which supports watching one or more variable values or array elements as the report runs

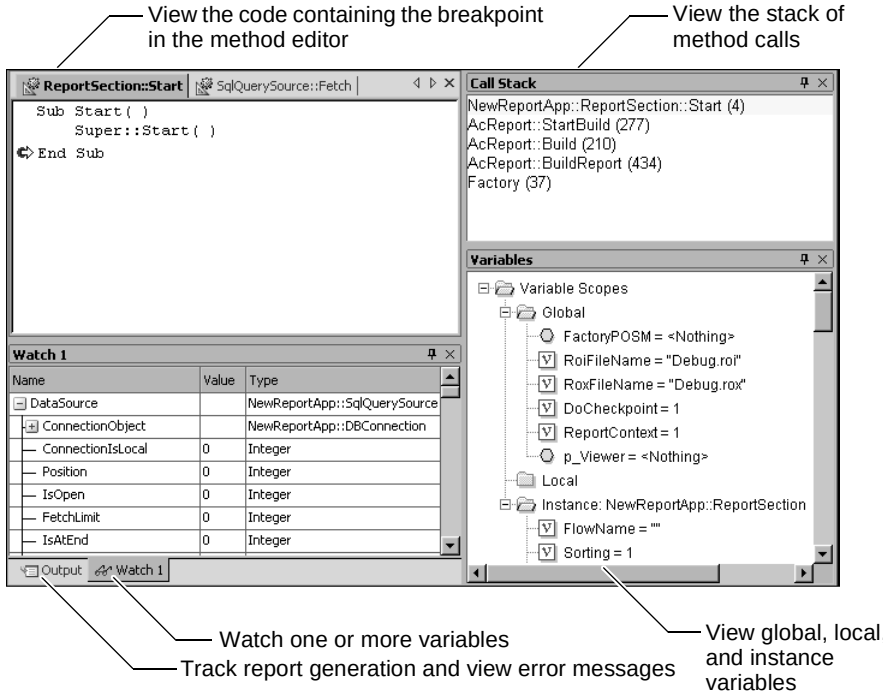


Figure 16-1 Debug perspective

About the Debug menu

The Debug menu, shown in Figure 16-2, provides access to the principal debugging tools and allows you to control code generation.

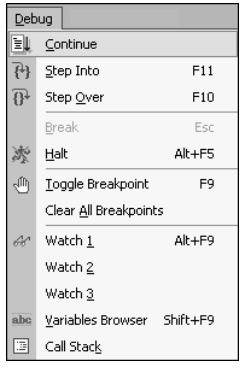


Figure 16-2 Debug menu

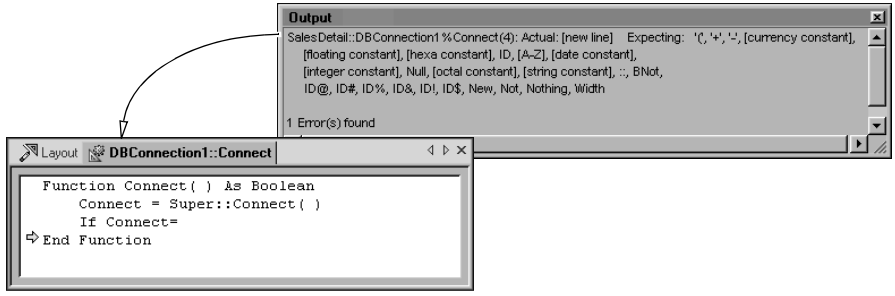
Table 16-1 describes the options available from the Debug menu.

Table 16-1 Debug menu options

Menu option	Description
Continue	Resumes report generation after a break or breakpoint.
Step Into	Supports line-by-line code generation in a method.
Step Over	Skips line-by-line code generation in a method.
Break	Pauses code generation in a large report and displays the source code, indicating where generation paused.
Halt	Stops code generation. You cannot resume code generation or debugging after you choose Halt.
Toggle Breakpoint	Sets a new breakpoint or clears an existing breakpoint in a method.
Clear All Breakpoints	Removes all breakpoints.
Watch 1, Watch 2, and Watch 3	Display the value of one or more report variables.
Variables Browser	Displays the Variables window, which shows the names and values of all variables in a report or in a specific portion of a report.
Call Stack	Displays a list of method calls up to and including the most recent call.

Viewing errors in the Output window

Output shows the progress of report generation and displays a message if an error occurs. When you double-click an error message in Output, e.Report Designer Professional displays the method that contains the error and flags the erroneous statement for review, as shown in Figure 16-3.



Double-clicking an error message in Output displays the statement that causes the error

Figure 16-3 Identifying errors in Output

Output also can identify obsolete properties. Properties become obsolete for many reasons, including changes to the data source and changes that occur from one Actuate release to another. Obsolescence also can occur when you change the parent class of a component. When you run the report, Output displays error messages about the obsolete properties. To remove references to these properties from the design, choose Report➛Verify Design.

About Variables

To monitor a variable during code generation, use the Variables window. The Variables window is available only during a debugging session. To use Variables, set a breakpoint in the method that contains the variable that you want to examine, then run the report. At the first breakpoint, Variables appears in the debug perspective.

Variables shows the values of all variables in context, organized by scope:

- Global. A global variable is accessible to the whole report. For example, the variables that store the names of the report object executable (.rox) file and report object instance (.roi) file have global scope.
- Local. A local variable is accessible within the method currently executing.
- Instance. An instance variable is accessible to an object. For example, in Figure 16-4, the SQLQuerySource object has the instance variables SelectClause, FromClause, WhereClause, and so on.

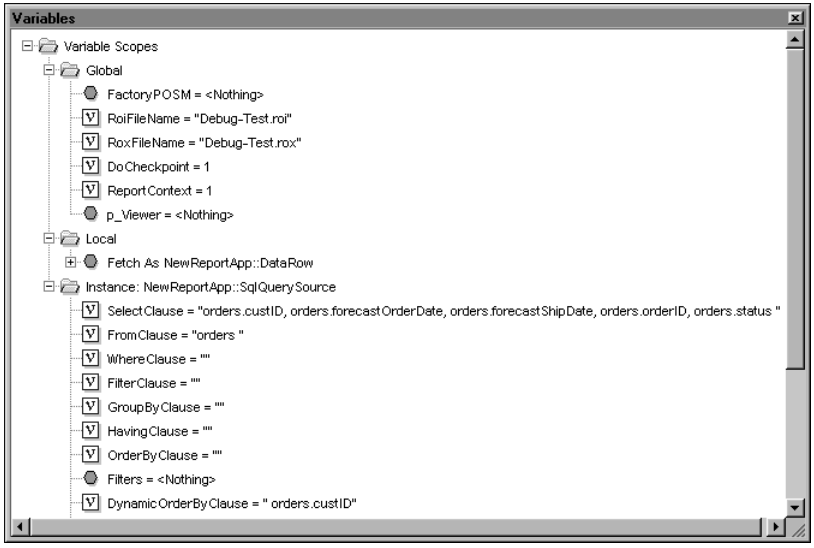


Figure 16-4 Debugging with Variables

As you step through code, variables information updates continuously. The following icons in Variables indicate the types of variables:

- Simple variable of a native data type, such as String or Integer.

- Structure. A data type that consists of a defined set of named simple variables.

- Object reference variable. An object reference variable gets its value by referring to another object. For information about object reference variables, see *Programming with Actuate Foundation Classes.*

- Array of objects of a native data type. A report can contain a number of arrays. For example, the controls in a frame typically appear in Variables as elements of an array.

About Call Stack

As you step through code generation, Actuate Basic keeps track of the method calls up to the current line. You can view this information in the Call Stack window. Each line in Call Stack indicates the method name and line number of the method in the source file. Call Stack tracks the method calls up to the current instruction. You can double-click any line in Call Stack to view the method in the source code.

In Figure 16-5, Call Stack displays the methods called before the first breakpoint, starting with the first call to the internal Factory( ) method.

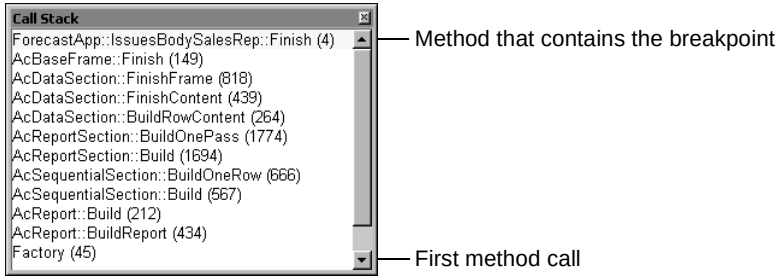


Figure 16-5 Debugging with Call Stack

A line in Call Stack that is highlighted in yellow shows where code generation suspended. A line highlighted in green shows the current context, where you last double-clicked to display the method in the source code. The methods in this window include private methods that you should not override.

Figure 16-6 and Figure 16-7 show how the contents of the Call Stack, source code, and Variables windows are related when you suspend code generation.

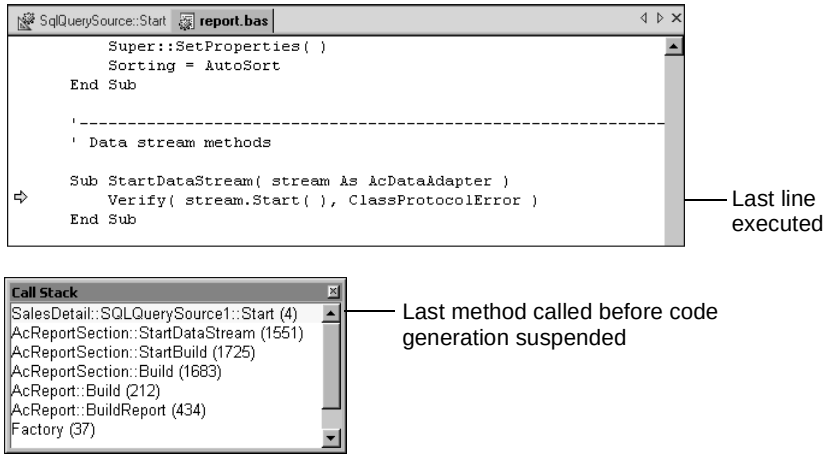


Figure 16-6 Call Stack and source code in code generation suspension

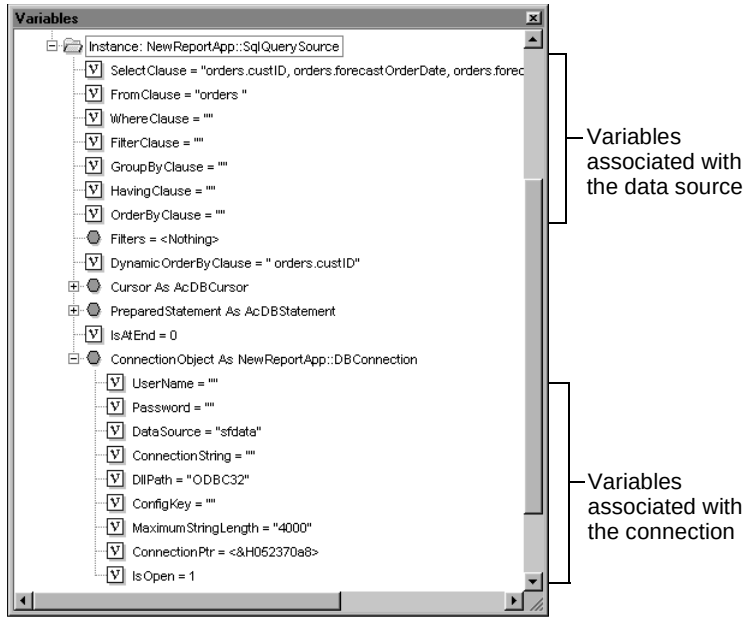


Figure 16-7 Variables window in code generation suspension

Actuate Basic sets a run-time stack limit of 200. A report that exceeds this limit causes a fatal error. A report using recursion with a large number of iterations can exceed this limit.

About the Watch window

Using the Watch window, you can:

- Monitor the value of one or more variables or array elements in the debug perspective.
- Modify the value of one or more variables or array elements at run time.

You can open up to three Watch windows at a time. In each window, you can monitor multiple variables or array elements.

As shown in Figure 16-8, Watch consists of three columns, Name, Value, and Type. You can widen or narrow these columns.

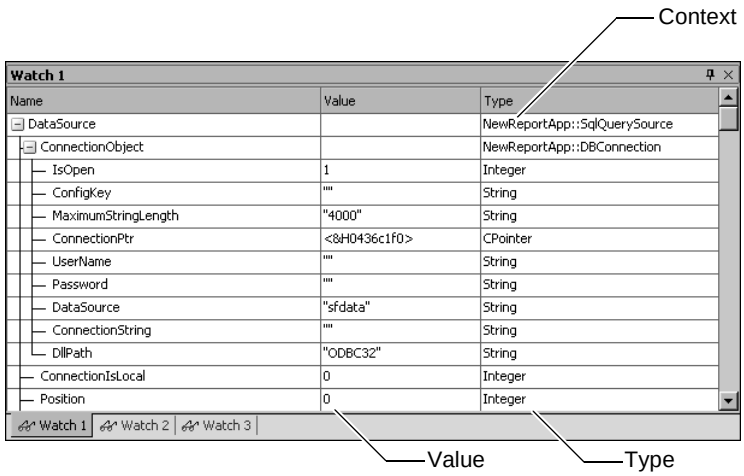


Figure 16-8 Debugging with Watch

In Name, type the name of the variable to watch. You also can drag the variable name from a source file or method editor and drop it in Name. If the variable is a structure, instance, or array variable, you can view the values assigned to its members by choosing the plus sign to expand it.

In Value, you can view the value of the variable.

In Type, you can view the data type of the variable. You cannot change the type. You can also view the context within the report for a structure, instance, or array variable.

Saving parameters for successive debugging runs

You can avoid having to enter parameters for successive test runs of a report by saving the test parameters to a report object value (.rov) file. You can save as many sets of parameter values as you want, each in an ROV. After saving

parameter values to a file, you specify that the report reads parameter values from a particular ROV.

Starting a debugging session

You can debug Actuate Basic methods and Actuate Basic source (.bas) files. The following sections provide step-by-step instructions for starting a debugging session for each type of file.

How to start a debugging session for a method

1. Select the method to debug and display it in the layout window:
   1. In Report Structure, right-click the class containing the method that you want to debug, then choose Properties. The Properties page for the class appears.
   2. Choose Methods. The list of methods associated with the class appears. The number of methods that you see depends on the filter setting for this report. You can choose filtering settings from the toolbar in Methods. Figure 16-9 shows the possible settings.

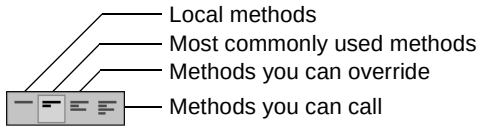


Figure 16-9 Filtering the methods in a report

   You can debug only methods that are declared or overridden in the class. To debug any of the inherited methods that appear in this list, access them from the class in which they are declared.

   3. Select a filter option. Methods displays the methods for the option that you selected.
   4. Double-click the method that you want to debug. The code for the method appears in the layout window.
2. To set a breakpoint at the line of code where you want to stop generation, place the cursor at the beginning of the line. As shown in Figure 16-10, a red dot indicates where a breakpoint is set to occur. Choose Debug➤Toggle Breakpoint.

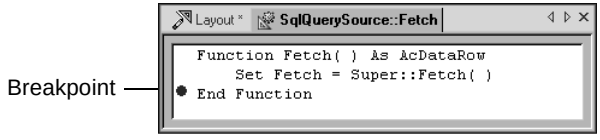


Figure 16-10 Setting a breakpoint

3 Choose Report➤Build and Run to start code generation. When Actuate Basic reaches the breakpoint, it stops code generation and displays the code in the debug perspective. An arrow appears at the beginning of the line of code where the breakpoint occurs, as shown in Figure 16-11.

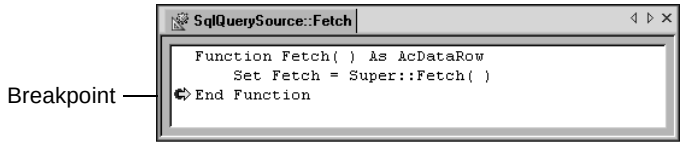


Figure 16-11 Breakpoint results

You can then use Actuate's debugging features to manipulate code generation or inspect variable values.

How to start a debugging session for an Actuate Basic source file

1 In e.Report Designer Professional, choose File➤Open. Select File appears.

2 Navigate to the Actuate Basic source (.bas) file that you want to debug. Choose Open. The code appears in the layout window, as shown in Figure 16-12.

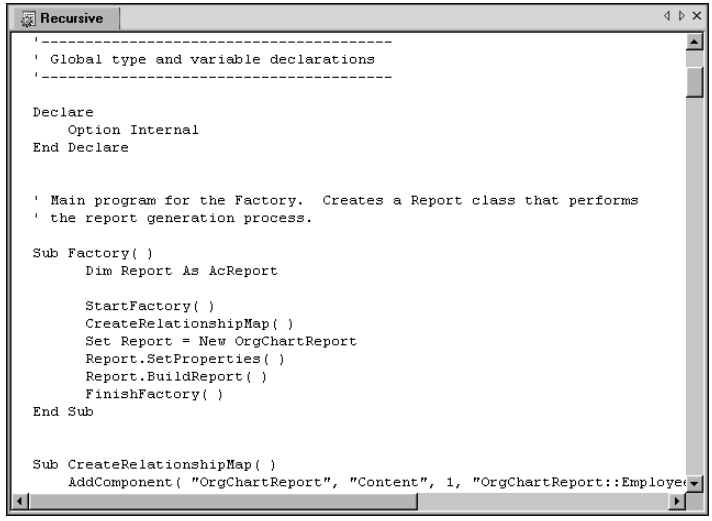


Figure 16-12 Actuate Basic source file

3 Set a breakpoint by placing the insertion point at the beginning of any line of code that performs an action. Choose Debug➤Toggle Breakpoint.

4 Close the BAS file.

5 Open the related report object design (.rod) file.

6 Choose Report➤Build and Run to start code generation. When Actuate Basic reaches the breakpoint, it stops code generation and the debug perspective opens. You can then use Actuate's debugging features to manipulate code generation or inspect variable values.

Controlling code generation

Using e.Report Designer Professional's debugging tools, you can set a breakpoint to control whether Actuate Basic executes a single line of code, an entire method, or a larger code block. By specifying when the report runs and when it pauses, you can accept areas that you know work correctly and focus on the code that requires attention. The following sections describe how to control report generation.

Running to a breakpoint

Use a breakpoint to pause report generation at a specific line of code. For example, if you do not want to trace through each line of a complex method, set a breakpoint just before the code that you want to examine. When Actuate Basic encounters a breakpoint, it suspends generation just before the line with the breakpoint. You can then use the other debugging tools to inspect the state of the method or trace generation line by line from that method.

When you close the report design, e.Report Designer Professional removes the breakpoints.

How to set a breakpoint

1 In a method or an Actuate Basic source (.bas) file, place the insertion point at the beginning of the line of code where you want to set or clear a breakpoint.

2 Choose Debug➤Toggle Breakpoint. A red dot appears to the left of the line of code to indicate the breakpoint, as shown in Figure 16-13.

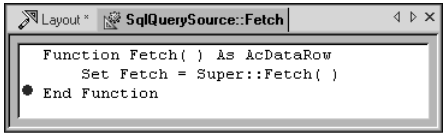


Figure 16-13 Debugging using a breakpoint

How to clear a breakpoint

In a method or an Actuate Basic source (.bas) file with a breakpoint set, choose Debug➤Toggle Breakpoint to clear the current breakpoint. To clear every breakpoint, choose Debug➤Clear All Breakpoints.

Stepping through code

Setting a breakpoint takes you to the location where you suspect a problem occurs. Then, you can step through your code and execute it line by line to monitor the effect of each statement. When you step through code, Actuate Basic executes the current statement, advances to the next statement, and suspends generation.

You can step through code in the following two ways:

- Step into each line of code, including code in called procedures.
- Step over code in called procedures.

In a debugging session, you typically use both techniques, depending on which portions of code you want to analyze at any given time. For example, if you step through method A and method A calls method B, you can step over method B if you know that method B works correctly. When you step over method B, Actuate Basic executes it as one unit instead of stopping after each line in method B. Conversely, you step into a called procedure if you are not sure the code is executing properly.

By stepping through code, you can observe how the Factory generates a report. You can step through the Actuate Basic source (.bas) files that make up the Actuate Foundation Classes and determine the sequence of method calls the Factory makes to create the report.

How to step into a method

Take one of the following actions to step into a procedure:

- Press F8.
- Choose Debug➛Step Into.
- Choose Step Into on the toolbar.

How to step over a method

Take one of the following actions to step over a procedure:

- Press F10.
- Choose Debug➛Step Over.
- Choose Step Over on the Debug toolbar.

Resuming code generation

After suspending code generation at a breakpoint, you can resume code generation until the next breakpoint or until the end of the program.

To resume code generation, take one of the following actions:

- Press F5.
- Choose Debug➛Continue.

- Choose Continue on the Debug toolbar.

Stopping code generation

You can stop code generation at any time. When you stop, Actuate Basic clears the program from memory. You cannot resume code generation for this debugging session once you stop.

To stop code generation, take one of the following actions:

- Press Shift+F5.
- Choose Debug➛Halt.

Examining and modifying variable values

The ability to examine the value of a variable is essential in debugging a report. For example, you can override the Start( ) or OnRow( ) methods of a text control to display an alert message when there is suspicious activity on a credit card account. By tracking the text control's value as the report generates, you can verify that the alert message appears only when you expect it to appear. Similarly, you can track the value of a variable in a data row as each Fetch( ) method executes to ensure that custom processing occurs correctly in each data row a report section receives.

To examine and modify variable values, you use two tools, the Variables window and the Watch window, in conjunction with the method editor. In Variables, you can view global, local, and instance variables in a tree structure that shows the variables in context. In Watch, you can monitor the values of variables as they change during report generation. You also can modify the run-time value of a variable to test the effects of changes without having to recompile the entire report.

Tracing object reference variables

In an Actuate report, an object typically contains object reference variables that refer to other objects. For example, a control has an object reference variable called Container that refers to the object that contains the control. A frame has several object reference variables, including Container, DisplayContainer, and Contents, that refer to the frame's container object in the content hierarchy, the frame's container in the display hierarchy, and the frame's contents, respectively.

Using the Variables window, you can view an object reference variable to understand how the variable points to other objects as the report runs.

Figure 16-14 shows how to expand successive levels of variables to view an object's container, the container's variables, and the variables for the contents of the container. In this illustration, a label control, CustIDLabel, has an object reference variable, Container. The container is the detail frame of the report. The container, in turn, refers to the components in the frame as an array of objects. This array includes CustIDLabel. Each component has a number of simple variables.

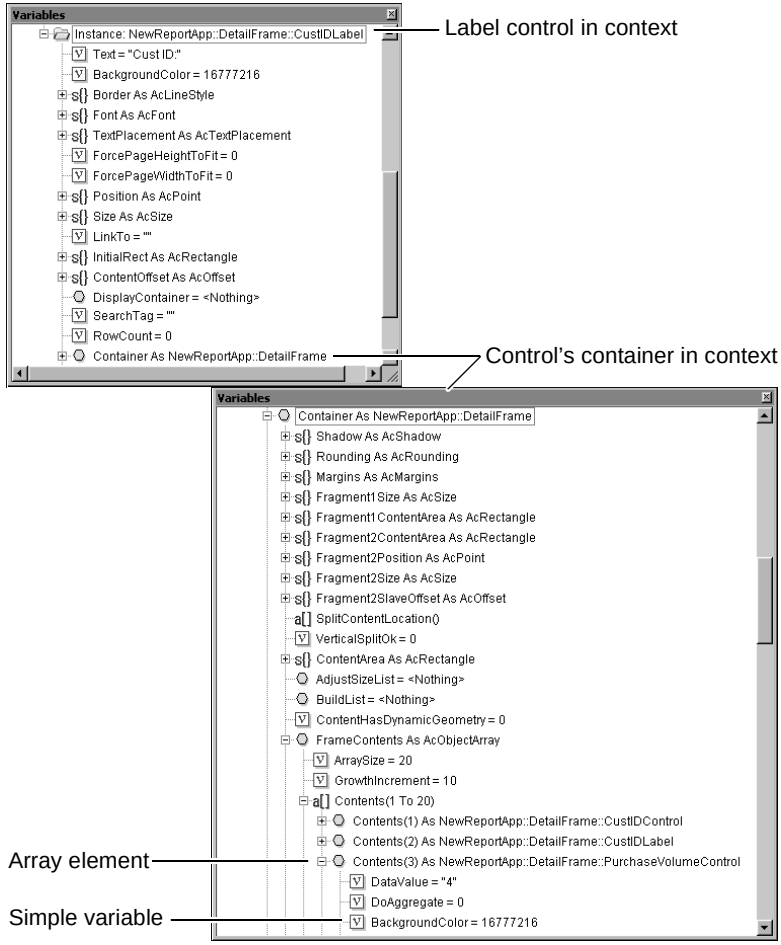


Figure 16-14 Viewing containers in the Variables window

Monitoring and changing the run-time value of a variable

To monitor the value of a variable during a debugging session, use the Watch window. You also can use Watch to change the run-time value of a variable or an array element. Watch displays variables in a tree structure similar to the one in the Variables window, as shown in Figure 16-15.

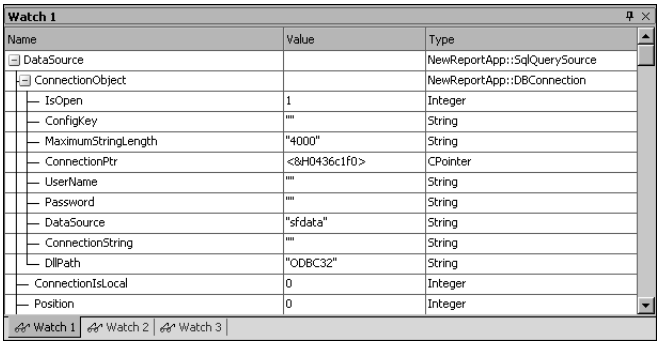


Figure 16-15 Monitoring variables using the Watch window

As you step through code, the list of variables in Watch remains the same, whether or not those variables apply to the current object. For example, as you monitor the DataValue variable for a data control, you can step into code that does not use the DataValue variable. DataValue remains in the Watch list but Actuate Basic does not recognize the variable for the current context and the value field looks like the one in Figure 16-16.

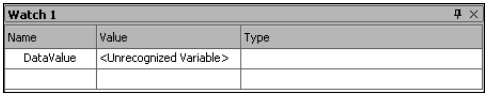


Figure 16-16 Example of an unrecognized variable

Inserting and arranging variables in Watch

There are two ways to insert a variable in the Watch window. You can type the variable name in the Name column and press Tab. The value and type of the variable appear in the window. You also can drag a variable from Variables and drop it in Watch.

You can change the order of variables in Watch by dragging a variable from one position and dropping it in another position.

How to monitor a variable value using Watch

1 In the design perspective, set a breakpoint in a method that uses the variable that you want to watch.

2 Run the report. At the first breakpoint, the debug perspective opens. A blank Watch window appears at the lower left of the perspective.

3 In the Name column, type the variable name and press Tab. Watch displays the value and type of the variable.

   If the variable is a structure, instance, or array variable, + appears beside the name. You can view the members of the variable by choosing the plus sign.

4 After you view the first occurrence of a variable value, you can run to the next occurrence by choosing Debug➤Continue. When the report runs to completion, the view perspective appears briefly. Then, the debug perspective returns. You can switch to the view perspective to see the report.

Editing a variable value

Using Watch, you can change the value of a native Actuate data type such as Integer, String, Double, Boolean, and so on. You cannot change the value of an instance or an array. For example, if LogoFont is of type AcFont and it has the members Size and FaceName, you can change the value of LogoFont.Size or LogoFont.FaceName but not the value of LogoFont.

When you change a variable value in one Watch window, the value changes in any other Watch window that lists the variable and in the Variables window. In Watch, your edits appear in red. When you change a variable value in Watch, you do not change the value of the variable in the design perspective. For example, if you change the size of a control in Watch, the control retains its original size in the design perspective.

A string value in Watch is enclosed in quotation marks. When you select a string variable value to edit, the quotation marks disappear. The quotation marks return when you finish editing and press Tab.

A change to a value lasts for the duration of the debugging session or until the value changes again.

How to modify a variable's run-time value using Watch

1 In the design perspective, set a breakpoint in a method that contains the variable that you want to watch. For example, in the Finish( ) method of an integer control, type the following code:

```
Sub Finish( )
   Super::Finish( )
   DataValue = 67890
End Sub
```

   Then set a breakpoint at the beginning of the last line of code, as shown in Figure 16-17.

Figure 16-17 Editing a variable value using Watch

2. Run the report. At the first breakpoint, the debug perspective opens. A blank Watch window appears at the lower left of the perspective.
3. Type the name of the variable to watch in the Name column. For example, using the code in step 1, type:

   ```
   DataValue
   ```

4. Press Tab. The value that you set in the Finish( ) method appears beside the variable name. The data type of the variable appears in the Type column.
5. Choose Debug➛Continue. The second occurrence of the variable appears. The value and type are the same as the first occurrence.
6. For the second occurrence, double-click the value. Type a new value. For example, type:

   ```
   12345
   ```

   The new value appears in red.
7. Choose Debug➛Continue. The third occurrence of the variable appears. The value and type are the same as the first occurrence. The value appears in red to indicate the change in the earlier occurrence.

   When the report runs to completion, you can switch to the view perspective to see the effects of your edits. The second occurrence of the variable value is different from the other occurrences, as shown in Figure 16-18.

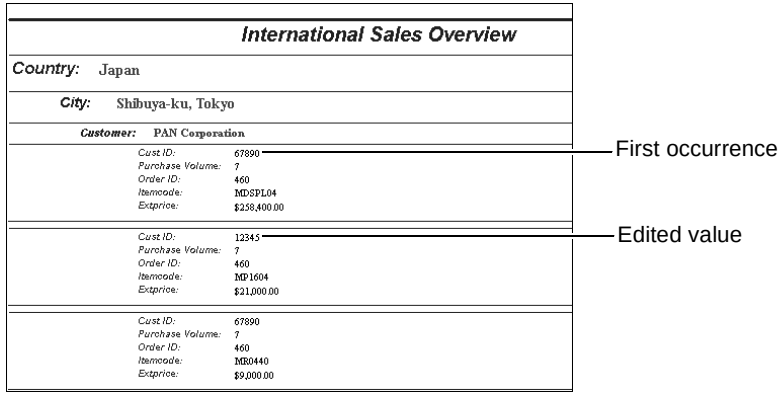


Figure 16-18 Viewing an edited variable value

Part Three

Displaying multiple reports from a single report design

Chapter

17

Designing for ad hoc reporting

This chapter contains the following topics:

- About ad hoc reporting
- Planning to use parameters
- Specifying how parameters appear to the user
- Deleting parameters
- Testing parameters
- Saving and reusing parameter values
- Suppressing parameter prompting
- Verifying parameter values that the user supplies

About ad hoc reporting

When you create a report, you build a query that extracts specific data from the data source to use in the report. Then you lay out and format the data in a way that best presents the data to report users. The report user sees the information that you selected in the format that you designed.

As users become familiar with the report and recognize its potential as an analytical tool, they will want to view the data in different ways. For example, in a sales report that contains nationwide data, a report user might want to view only sales in a single state, or sales over a certain amount, or sales completed in the last 30 days.

The simplest solution for this type of ad hoc reporting requirement is to ask the report user to determine which data to retrieve and display when the report runs. A report developer makes this solution available by creating parameters.

About parameters

Report parameters determine which data the report displays. Instead of retrieving all data from a data source, a report that uses parameters retrieves a specific subset of data. For example, if you are working with an inventory database, you can use parameters to retrieve data about heavy equipment only, or data about all inventory purchased from a specific vendor, or data about office furniture purchased between two dates. The report developer creates parameters during the design process. The report user can set parameter values by typing or selecting values on the Requester dialog box. If the user does not set parameter values, e.Report Designer Professional uses a default value the report developer defines.

Report parameters can be optional or required. Optional parameters do not require that the report user enter values when running the report. Optional parameters appear in the parameter list without a trailing asterisk, as Figure 17-1 shows.

If the report contains required parameters, the user must provide a value in order for the report to run. For instance, accessing a database typically requires a user name and password. Requester indicates that a parameter is required by displaying an asterisk after the parameter name. A required parameter typically has a default value that the developer provides. In Figure 17-1, two output parameters are required and default values are supplied so that the report runs. If you change one of the default values, such as the Output File Name value, the new output file name overrides the default value.

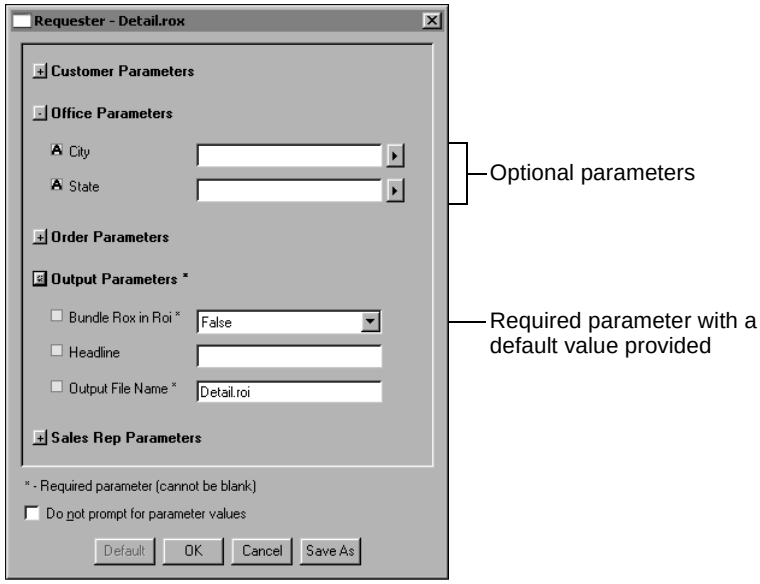


Figure 17-1 Optional and required parameters in Requester

Using parameters in a report provides the following benefits:

- Parameters generate on-demand reports.

  By using parameters in your report design, you can create a single report that generates specialized reports, on demand, to meet the specific needs of each report user.

- The report developer designs one report, which report users can run multiple times to suit different requirements.

  Parameters are essential for time-sensitive reports. Consider the report that does not use parameters:

  To provide sales results for each month, a report developer generates a sales report each month. When the report developer first creates the report, he builds a query to show only sales data for the month of January. The next time the report developer needs to generate a report for February's numbers, he must modify the query to get only sales data for February. Without parameters, the report developer must modify the query every month before distributing the report to users. By creating a parameter that prompts for the month for which to view sales information, the report developer creates one report and runs it each month to refresh data.

- Parameters support managing large reports

  A report design can generate a detailed report that is several hundred pages long. It can display, for example, all itemized sales orders for all customers in all cities in every country. To reduce the size of the report and let the user

select what the user wants to view, you can use parameters to generate only parts of the original large report. You can, for example, create parameters that ask the user to supply a customer, or range of customers, a city, or a country.

- Parameters enable the report user to decide the format of a report.
  In addition to using parameters to determine what data to retrieve from the data source, you can create parameters that ask the user how to format a report. For example, you can make the background color property or the visibility property of a component a parameter. When the user runs the report, the user can select a color or choose to display certain information.

Understanding how a report prompts the user for input

When you create parameters to prompt the user for input, e.Report Designer Professional displays the parameters on Requester each time the user runs the report.

Figure 17-2 shows Requester for the example report Detail.rod.

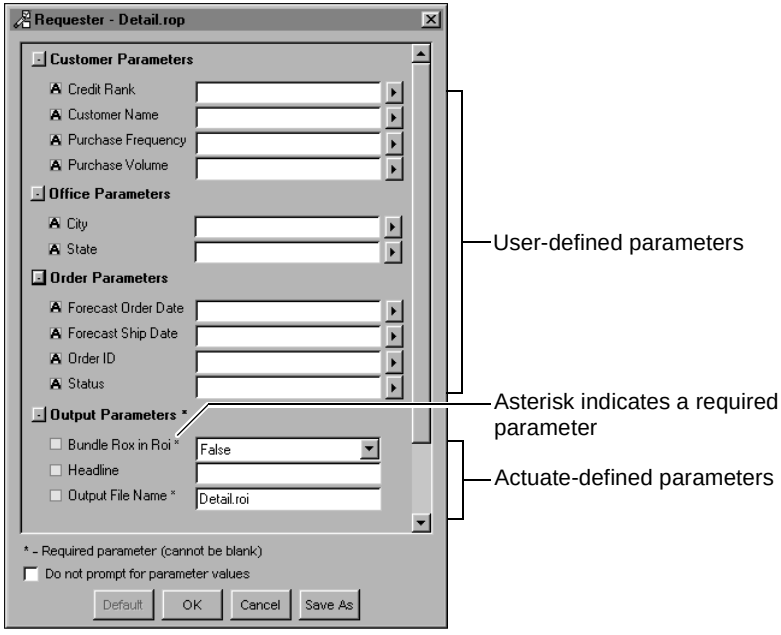


Figure 17-2 Parameters displayed on Requester

The output parameters are defined by the software. These parameters prompt the user to specify:

- Whether to bundle the report executable with the generated report.

- A headline, which appears in Actuate Information Console if you publish the report to an Encyclopedia volume.
- A name for the generated report file. The default name is the same name as the report design file name.

Requester's default behavior is to appear only if there are user-defined parameters. If you want Requester to appear when there are no user-defined parameters, to prompt the user to specify values for the output parameters, choose Report➤Design Properties and select Always prompt for values.

Planning to use parameters

When you create parameters for a report design, consider the following:

- Think of the different ways a user needs to filter the information. Create parameters to meet those requirements.
- If you create many parameters, organize them into logical groups. Look at the Detail report example that ships with the product. It uses four parameter groups, Customer parameters, Office parameters, Order parameters, and Sales Rep parameters.
- Use short but descriptive text prompts. For example, Customer State is clearer than State.
- Avoid forcing the user to type information. If the report design requires a parameter value, supply a default value.
- Do not assume the report user knows how the data source represents data. For example, a user might not know that an Orders.Status field takes three values, Open, Closed, and In Evaluation. Without this knowledge, the user does not know what value to supply for an Order. Status parameter. To improve usability, create a drop-down list or series of radio buttons for the user to select a parameter value, instead of having the user type the value.

There are two types of parameters, ad hoc parameters and static parameters. Both types can narrow the scope of the data that a SQL query retrieves. Static parameters also can change the appearance or behavior of report output, without modifying the SQL query.

About ad hoc parameters

A report developer designs a SQL query to retrieve data from a database. By selecting or typing values for ad hoc parameters in Requester, the report user modifies the WHERE clause of the SQL statement to retrieve data that meets certain criteria. For an ad hoc parameter, a user can provide a single value, a range of values, a list of values, or no value at all. Ad hoc parameters can be either optional or required.

An ad hoc parameter must be associated with a field in the query. The report developer can create this association using Query Editor, as shown in Figure 17-3.

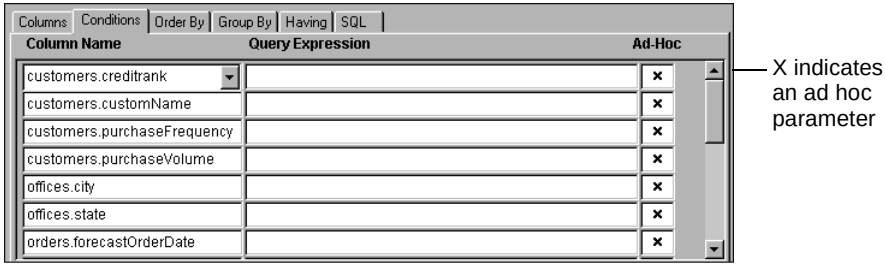


Figure 17-3 Creating an ad hoc parameter

In Requester, an A next to the parameter name indicates that a parameter is ad hoc. Figure 17-4 shows two optional ad hoc parameters, City and State, in Requester.

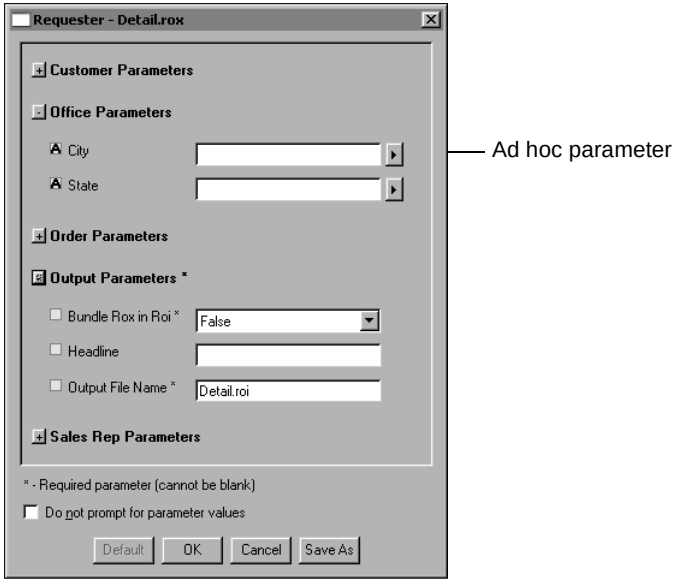


Figure 17-4 Ad hoc parameter values in Requester

About static parameters

A static parameter is a global variable that a report developer creates using the Parameter Editor. As with ad hoc parameters, the report user can set an initial value when generating a report, thereby modifying the WHERE clause of the SQL query. The user can provide one and only one value for each static parameter.

For static parameters, e.Report Designer Professional always looks for a run-time value. If the user does not provide a value, the software uses the default value that the report developer provided. If the default value is Null, the report output might not be what the user expects. For example, if a parameter filters state data and the parameter value is Null, the report runs but does not display data. When you create a static parameter, therefore, you should require a value from the report user.

Unlike ad hoc parameters, static parameters do not have to be associated with a database field. You can, therefore, use static parameters to change the appearance or behavior of a report at run time. For example, you can create a Summary parameter to indicate whether the report output is summary data. If the value of Summary is True, then the report output is a summary. If the value is False, then the report output shows all the details.

Specifying how parameters appear to the user

When you create a parameter using the procedures that are described in the previous sections, e.Report Designer Professional adds the parameter to Requester to prompt a user to supply information when the user runs the report. You do not need to do anything else to make a parameter work. Typically, however, you change how the parameter appears on Requester.

You can design user-friendly parameters using the following techniques:

- Use a descriptive prompt that states clearly what values a user can supply. This approach is especially important for data-filtering parameters where the value that a user supplies must match the corresponding data in the data source. For example, if a state field in the data source stores state names by their two-letter code, rather than by their full name, you need to convey this to the report user. Instead of using State as the text prompt, you could use one of the following strings:

  ```
  State, for example, CA
  ```

  ```
  Supply the state abbreviation
  ```

  With these prompts, it is clear what values the user must type.

- Provide default values that result in a well-presented report, in case a user does not supply a value. It is good practice to specify a default value if you make the parameter required. For example, provide a default value, CA for California. If a user does not change that value, the report runs and displays a report document that contains data for California.

- If you created many parameters, organize them into logical groups. For example, show the parameters for customer name and credit rank in a customers parameters group.

You can enhance the usability of parameters further by using the following advanced techniques:

- Create a drop-down list or series of radio buttons that provide values from which a user can select.
- Create check boxes from which a user can select true or false.
- Provide ranges of numbers or dates from which a user can select.
- Mask text values that the user supplies. You can use this technique for a password parameter. When the user types the information, asterisks (*) appear.

Specifying a parameter's display name, group, and default value

To make parameters user-friendly, you typically change a parameter's default display name, assign the parameter to a group, and specify a default value. For example, Requester can display a default value for the Credit Rank and Customer Name parameters in a Customer Parameters group, as shown in Figure 17-5.

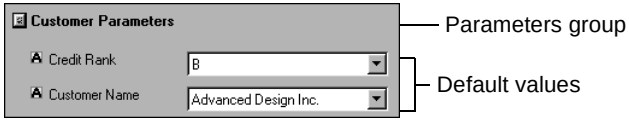


Figure 17-5 Default values for Customer Parameters group

How to set a parameter's display name, group, and default value

1. Choose Tools➛Parameters. Parameter Editor displays all the parameters that you created and the predefined output parameters.
2. Select a parameter to modify. Choose Modify.
3. In Parameter Properties—General, as shown in Figure 17-6, you can specify the following properties for the parameter:
   - To set the parameter prompt that appears on Requester, type a descriptive prompt in Display name. If the text is too long, Requester truncates it.
   - To list the parameter in a group of related parameters, specify a group name in Group.
   - To specify a default value for the parameter, type a value in Default value.

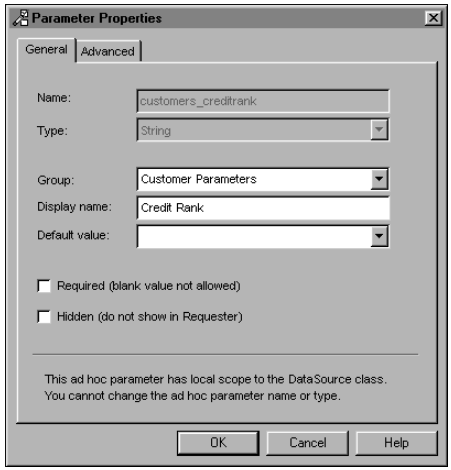


Figure 17-6 Specifying parameter display name, group, and default value

Choose OK.

Providing the user with a list of choices

Drop-down lists and radio button series are tools that provide a list of choices to a user. In both cases, a user can make only one selection from multiple choices.

The following list describes the differences between these user interface elements:

- Drop-down lists save space and are less visually distracting to a user, because they take up only one line on Requester. Drop-down lists do not require a default value. For usability, a drop-down list should contain fewer than 100 entries.
- Radio button series require as many lines as there are choices, and you must specify one as a default value. For usability, a set of radio buttons should contain fewer than 10 entries.

You create a list of values for a drop-down list or a series of radio buttons by typing the values or by creating an Actuate SQL query that retrieves the values.

Figure 17-7 shows an example of a drop-down list on Requester.

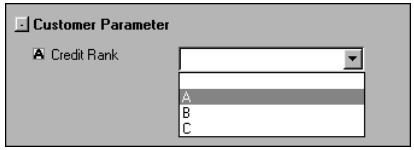


Figure 17-7 A drop-down list of parameters

Figure 17-8 shows a series of radio buttons on Requester.

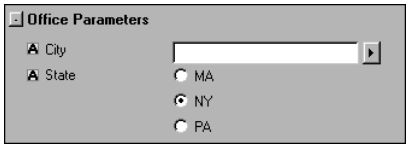


Figure 17-8 Radio buttons on Requester

Typing a list of parameter values

You can create a list of values for a drop-down list or a set of radio buttons by typing the values. A value can appear as it appears in the data source, or you can supply an optional display name that appears on Requester.

How to type a list of parameter values

1. Choose Tools➛Parameters.
2. In Parameter Editor, select a parameter to customize. Choose Modify. Parameter Properties—General displays optional properties that you can modify.
3. For Display name, you can type a descriptive prompt for the parameter. This text appears on Requester. For example, instead of the default value for the data source column name parameter, offices_state, you can type:

   ```
   Select a state:
   ```

   The display name appears on Requester, as shown in Figure 17-9.

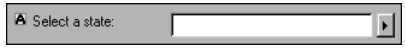
   

   Figure 17-9 Example of a customized display name
4. Choose Advanced.
5. In Parameter Properties—Advanced, select Drop-down list or Radio buttons.

   If you create a drop-down list for a parameter other than a date or ad hoc parameter, you can select Allow user to type a value. An ad hoc parameter accepts a QBE expression as its value.

6. In Parameter Properties—Advanced, choose New.
7. In Values and display names, complete the following tasks:

   - In Value, type a value that matches a value for this parameter in the data source.

     This value appears in the drop-down list or next to a set of radio buttons if you do not provide an optional display name. Depending on the type of data-filtering parameter, you might be able to type a QBE expression, such as 1–10 for parameters of integer type, or you might need to supply a specific value. If you supply a specific value, the value must match exactly the field data in the data source. For example, if the parameter asks for an

order status, and the values in the data source are Closed and Open, you must use those values. If you type a different value, such as Shipped or Pending, your report runs, but it contains no information, because the data source contains no matching values.

- In Display name, you can type an optional name for the value.
  This display name appears in the drop-down list or next to a radio button.

8 To supply the next value, choose New. Repeat this step until you finish supplying all the values for the drop-down list or set of radio buttons. The order in which the values or display names appear in this list is the order in which they appear in the drop-down list or set of radio buttons on Requester.

You can rearrange the order in which the values appear by dragging-and-dropping the values, by using the up and down arrows, or you can choose Sort to alphabetically sort the values. Alphabetical sorting applies to the display names if you provide them.

9 Set one of the values as the default by selecting the value then choosing Default, as shown in Figure 17-10.

Values and display names

✓	Value	Display name
	CT	Connecticut
	MA	Massachusetts
	NJ	New Jersey
	NY	New York
✓	PA	Pennsylvania

Figure 17-10 Specifying a default parameter value

Choose OK.

Creating a query that retrieves a list of parameter values

You can create a list of values for a drop-down list or a set of radio buttons by writing an Actuate SQL query that retrieves the values from an information object (.iob) or data source map (.sma) file. If you create a query to retrieve the values, you do not have to modify and recompile the report design when the values change. e.Report Designer Professional also supports using a query that retrieves a value using a display name. For example, you can create a query that retrieves a customer ID value when a user chooses a customer name from a drop-down list or set of radio buttons. The Actuate SQL query must meet the following requirements:

- The query must retrieve data from an information object (.iob) or data source map (.sma) file that is stored in an Encyclopedia volume.
- The query can retrieve values of one of the following types:
  - A single column whose data type matches the parameter's data type.

- A column whose data type matches the parameter's data type and another column whose values appear as the display names in a drop-down list or set of radio buttons

- The query must not contain a WITH clause.
- The query must not return more than 500 rows.

You can create an Actuate SQL query by selecting an information object or data source map and a column in Dynamic List of Values Builder, as shown in the following example:

```
SELECT DISTINCT CATEGORY
FROM "AQ/Information Objects/io_items_agr_group_sql.iob"
ORDER BY 1
```

In Dynamic List of Values Builder, you can also create an Actuate SQL query that retrieves a column value and a display-name column value from an information object or data source map. For example, the following Actuate SQL query retrieves the custID column value as the parameter value. The second column name in the query, customName, specifies the column value that appears in the drop-down list or set of radio button labels. In this example, ORDER BY 2 sorts the values that appear in the user interface:

```
SELECT DISTINCT custID, customName FROM "/MyHomeFolder
   /MyProject/Data Sources/MyDatabase/dbo.customers.sma" ORDER
   BY 2
```

You must modify the default query if either of the following conditions are true:

- The information object defines parameters, and you want to provide values for these parameters, as shown in the following example:

  ```
  SELECT DISTINCT CATEGORY
  FROM "AQ/Information Objects/io_items_agr_group_sql.iob" [5,
     "CA"]
  ORDER BY 1
  ```

- You want to change the sort direction for the list of values, as shown in the following example:

  ```
  SELECT DISTINCT CATEGORY
  FROM "AQ/Information Objects/io_items_agr_group_sql.iob"
  ORDER BY 1 DESC
  ```

e.Report Designer Professional does not validate the query.

The values that the query returns appear when a user supplies a value for the parameter in Actuate Information Console. The values do not appear, however, when you specify a value for the parameter on Requester. In Requester, you must type the value in a text box.

For more information about Actuate SQL, see *Accessing Data using e.Report Designer Professional*. For more information about information objects, see

Accessing Data using e.Report Designer Professional. For more information about running a report in Actuate Information Console, see *Using Information Console.*

How to create a query that retrieves a list of parameter values

1 In Parameter Properties—Advanced, select Dynamic list of values. The Dynamic list of values pane appears in the lower half of Parameter Properties—Advanced.

2 Choose Ellipsis, as shown in Figure 17-11.

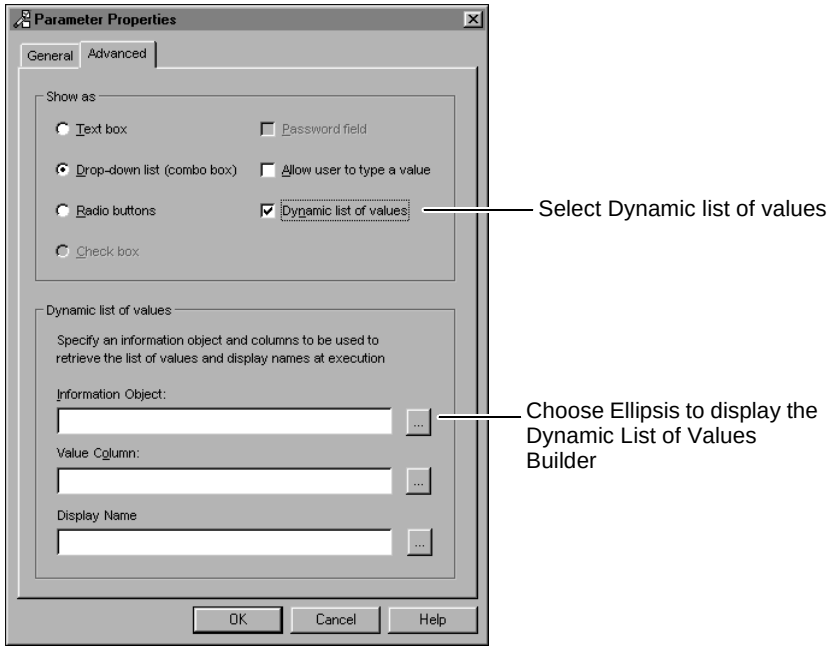


Figure 17-11 Parameter Properties—Advanced

If you have not selected a default iServer profile, Dynamic List of Values Builder—iServer appears. If you select a default profile, Dynamic List of Values Builder—Information Object appears, and you can proceed to step 4.

3 Select the iServer profile for the Encyclopedia volume that contains the information object or data source map. Choose Next.

If the iServer profile does not appear on Dynamic List of Values Builder—iServer, choose Create a new iServer profile to create it, as shown in Figure 17-12.

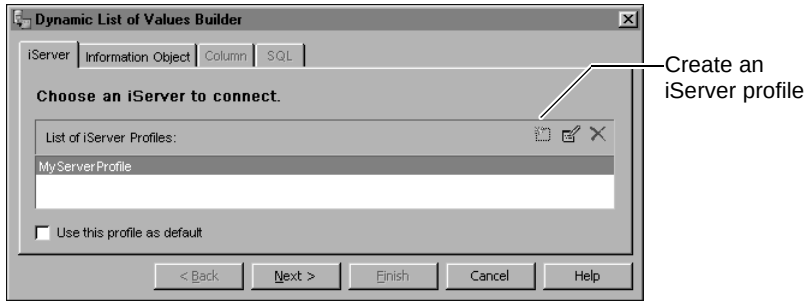


Figure 17-12 Creating a new iServer profile

4 In Dynamic List of Values Builder—Information Object, navigate to the appropriate information object or data source map, and select it, as shown in Figure 17-13. Choose Next.

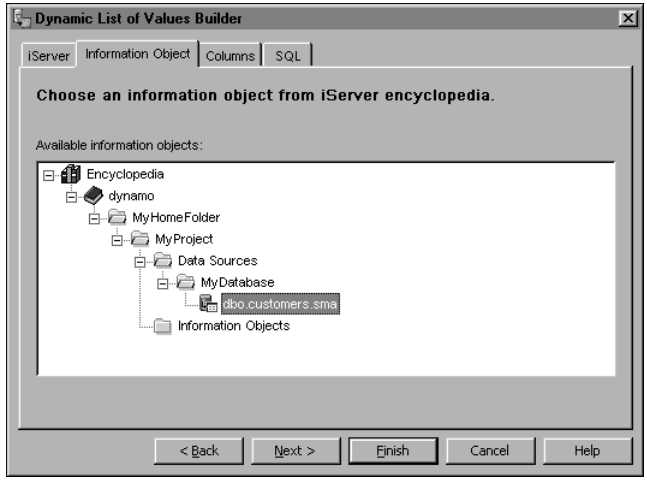


Figure 17-13 Dynamic List of Values Builder—Information Object

5 In Dynamic List of Values Builder—Columns, complete the following tasks:

- In Available value columns, select a column.
- In Available display name columns, you can select a column that contains the display names that appear in the drop-down list of values.

  For example, Figure 17-14 shows how to set up the query for the custID column value using a display name from the customName column. When a user runs the report, that user chooses a customer name to retrieve a customer ID value for the parameter.

  Choose Next.

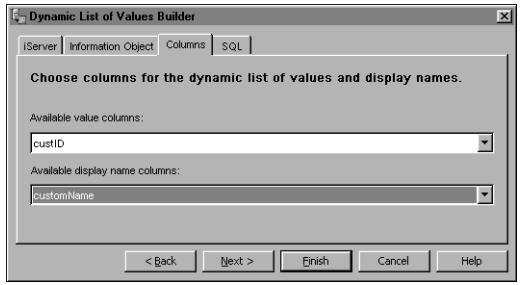


Figure 17-14 Dynamic List of Values Builder—Columns

6 In Dynamic List of Values Builder—SQL, complete one of the following tasks:

- If you do not want to modify the Actuate SQL query, choose Finish. Figure 17-15 shows an Actuate SQL statement that retrieves and sorts the custID column values for the drop-down list or radio button labels.

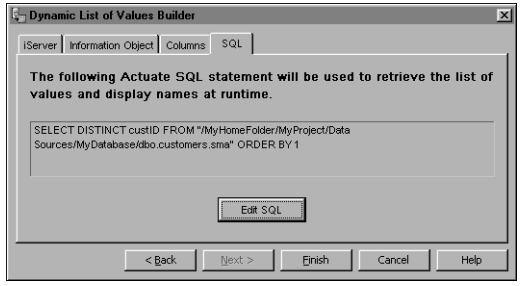


Figure 17-15 Dynamic List of Values Builder—SQL

- If you want to modify the Actuate SQL query, choose Edit SQL, and modify the query. Then, choose Finish.

Figure 17-16 shows an Actuate SQL statement that retrieves a custID column value using the customName value as the display name in the drop-down list or radio button labels. ORDER BY 2 sorts the display name values.

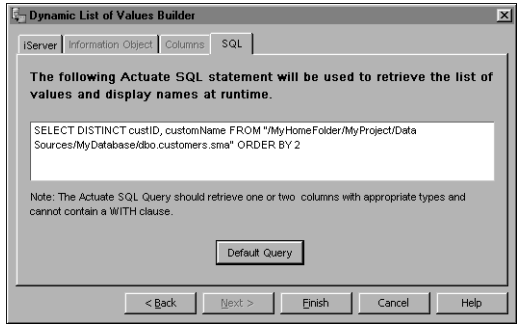


Figure 17-16 Example of an Actuate SQL statement

Parameter Properties—Advanced displays the Actuate SQL statement, as shown in Figure 17-17.

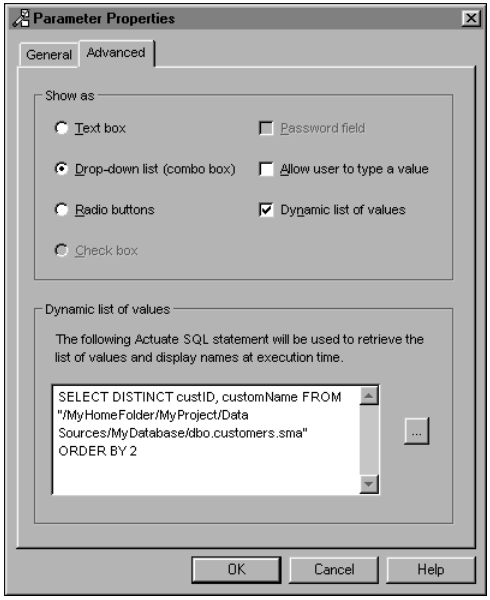


Figure 17-17 Parameter Properties—Advanced

Providing the user with true or false options

Use a check box to prompt a user to choose True or False for a parameter of Boolean type. The check box is only available for parameters of Boolean type. Figure 17-18 shows an example of a check box on Requester.

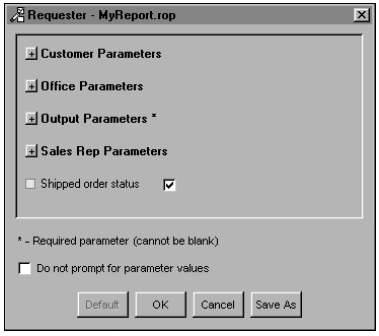


Figure 17-18 Check box example

How to set up a check box

1 Choose Tools➛Parameters.

2 In Parameter Editor, select a parameter to modify. The parameter must be of type Boolean. Choose Modify.

3 In Parameter Properties—General, for Display name, type a descriptive prompt for the parameter. This text appears on Requester. Choose Advanced.

4 In Parameter Properties—Advanced, you can set values for the true and false labels, as shown in Figure 17-19. You also can indicate the default value to display on Requester:

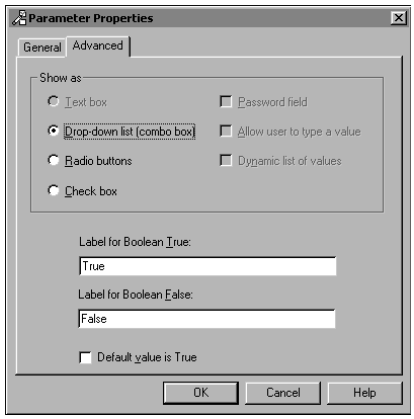


Figure 17-19 Setting up a check box

Select Check box. The bottom half of the window displays the default value for the parameter, as shown in Figure 17-20. You can change this value.

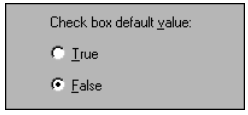


Figure 17-20 Setting the default value for the check box

Choose OK.

Specifying the order in which parameters appear

Requester's default behavior is to display parameters by parameter name in ascending alphabetical order. Note that the parameter name is different from the parameter display name. If you provide a display name, Requester's default behavior is to display parameters by display name in ascending alphabetical order. If you create parameter groups, Requester sorts the parameters by group name first, then by display name or parameter name within each group.

Figure 17-21 shows the default order in which parameters appear.

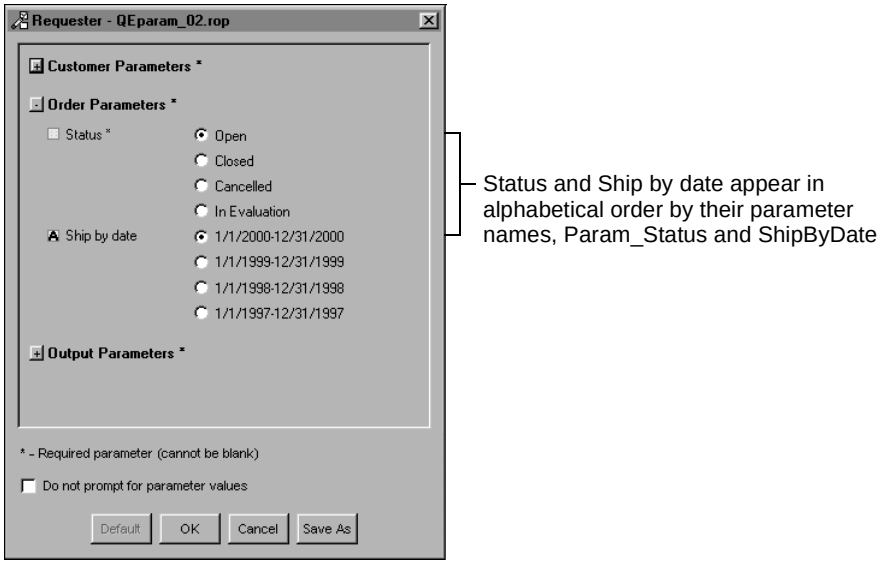


Figure 17-21 Default sort order of parameters

Because a user does not see the actual parameter names, the default sort order appears random to the user, especially if there are no groups to organize related parameters. A more intuitive sort order is to display parameters alphabetically by their display names.

To sort parameters by their display names, on Report Structure, right-click the top-level report component. Choose Properties. The Properties page for the report component appears. Set SortParamsByAlias to True.

Hiding a parameter

You can specify that a parameter does not appear on Requester. Use this approach when you want to supply a value programmatically, rather than through user input.

For example, you can link two subreports in a master-detail relationship based on a common field. The value of the common field in the master report controls the data that appears in the detail report. You need to make the common field a parameter in the detail report so that, at run time, the detail report can get the current value from the master report. Because the parameter value is passed programmatically, you should hide it from Requester so that the user does not see it.

How to hide a parameter

1 Choose Tools➛Parameters.

2 In Parameter Editor, select the parameter that you want to hide. Choose Modify.

3 In Parameter Properties—General, select Hidden. Choose OK.

Deleting parameters

You use the same tool to create and delete a parameter. For example, if you create a parameter using Query Editor, you must delete the parameter in Query Editor. If you create a parameter using a component's Properties page, you must delete the parameter on the Properties page.

If you are not sure how a parameter was created, use the Parameter Editor to view information about the parameters in a report.

Parameter Editor lists all parameters, as shown in Figure 17-22. To open Parameter Editor, choose Tools➛Parameters.

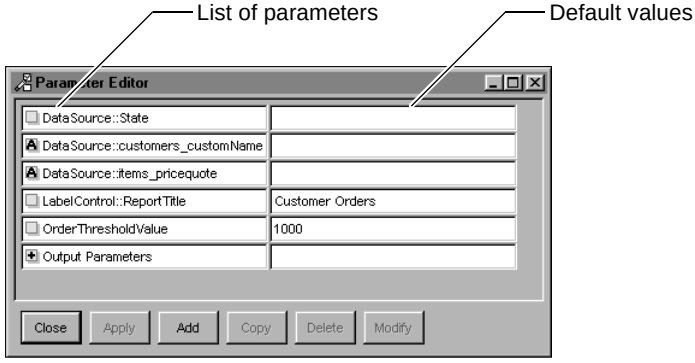


Figure 17-22 Parameter Editor

Parameter Editor displays parameters using the following convention:

```
<component name>::<parameter name>
```

The component name typically indicates which tool created the parameter. For example:

- DataSource::State indicates that DataSource is the component. When DataSource is the component, the developer used Query Editor to create the parameter.
- LabelControl::ReportTitle indicates that LabelControl is the component. When LabelControl is the component, the developer used the Properties page of LabelControl to create the parameter. To confirm this fact, find the component in Report Structure.

- OrderThresholdValue, without a component name, indicates that this parameter is not associated with a component. In this case, the developer used Parameter Editor to create the parameter. To confirm this fact, select the parameter. If the Delete button is available, the developer used Parameter Editor.

When you know which tool a developer used to create a parameter, you can go to the source to delete the parameter.

You can use Parameter Editor to delete a parameter only if you created it using Parameter Editor. Although you can view and modify any parameter using Parameter Editor, the Delete functionality is unavailable for deleting parameters that were created on the Properties page or in Query Editor.

How to delete a data-filtering parameter

1. Choose View➛Data.
2. In Query Editor, choose Conditions.
3. Select the parameter to delete. Press Shift+Delete, as illustrated in Figure 17-23.

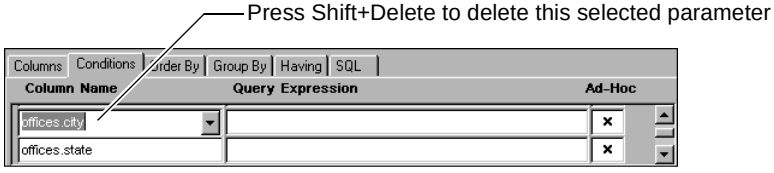


Figure 17-23 Deleting a parameter

e.Report Designer Professional removes the parameter from the list of conditions. When you run the report, the parameter does not appear on Requester.

How to delete a parameter that sets the value of a component property

1. Right-click the component that uses the parameter that you want to delete. Then, choose Properties, as shown in Figure 17-24.
2. In Properties, click either the property or parameter name.

3. Choose Parameter.

Figure 17-24 Deleting a component's parameter

e.Report Designer Professional removes the parameter from the property value. When you run the report, the parameter does not appear on Requester.

Testing parameters

It is important to test all the parameters you create to verify that they work the way you intend and meet user needs. Testing entails running the report, supplying different parameter values, and checking the generated report carefully. You can test parameters using the following techniques.

Test each parameter as you create it

If you create many parameters, it is best to test each parameter as you create it. If you wait until you create all the parameters before you begin testing, it is much harder to debug errors in the output because it is not immediately clear which parameter causes the problem.

Think like a user

Try to look at the parameters from a user's perspective. For each parameter, ask if it is obvious what a user needs to supply. For example, you have a parameter called Customer Credit Rank. Is it clear to a user that he needs to supply A, B, or C? The answer is no. Unless the user knows how data appears in the data source, the user does not know what to supply. The parameter description needs to be clearer.

Parameters that take date values are another error-prone area. For example, if you have a parameter that appears as Order Ship Date, what date format must the user use? If the data source stores dates in mm/dd/yyyy format, the report displays nothing if the user types 2/20/04 because there are no matches. If the report will be used in multiple locales, you might consider creating a date prompt that is locale-neutral and converting, through code, user values to the data source format.

Run the report without specifying any parameter values

The result of running the report without specifying any parameter values should be a report with information in it. If you get an empty report, it probably means that one or more of the parameters requires a default value. Make sure you provide a default value for each required parameter, or avoid required parameters altogether. If possible, do not force the user to provide values.

Check that the default values make sense

For parameters, such as page color, that change the look of a report, the result of a default value is obvious, and you can make adjustments easily. For parameters

that filter data, inspect carefully what information is generated. Generally, you want the report to display all information specified by your original query.

Test each selection in a drop-down list or set of radio buttons

If you created a drop-down list or series of radio buttons for the user to choose values from, test each selection, and check that the output is correct.

For example, if you created a drop-down list for an order status parameter, and the choices are Closed, Open, and In Evaluation, you should run the report four times. First, run it without selecting any of the choices, in which case the report should display records for all order statuses. Then, run the report once for each possible choice.

If the report appears with no data, it means no records matched your selection. For example, if you select Closed and no data appears when you run the report, it could indicate one of the following three possibilities:

- There are no closed orders, which is unlikely.
- Closed is not a valid value. Perhaps the value in the data source is Shipped, not Closed.
- Another parameter causes the problem.

To debug the first two possibilities, check the data in the data source. Debugging the third possibility requires more effort if you have many other parameters. You can avoid this possibility altogether if you always test parameters, one at a time, as you create them.

Saving and reusing parameter values

If you run a report regularly to test the results of various configurations of parameter values, it is tedious to type values over and over again. To solve this problem, save the parameters values that you use frequently to a report object value (.rov) file.

You can save as many sets of parameter values as you want, each in an ROV. After saving values to a file, you can specify that the report reads parameter values from a particular ROV.

How to save a set of parameter values to file

1. Choose Report➛Build and Run.
2. In Requester, from the list of parameters, supply the parameter values to save to a report object value (.rov) file. Choose Save As.

3 In Specify Parameter File Name, in File name, type a name for the report object value (.rov) file. Choose Save. The parameter values that you supplied are saved to the specified ROV.

4 In Requester, complete one of the following tasks:

- Choose Default to reset all parameters to their default values.
- Choose OK to generate a report with the current parameter values.
- Choose Cancel to return to the report design without generating a report.
- Supply a new set of parameter values to save, then choose Save As to save the values to another ROV.

How to specify the report object value (.rov) file to use

1 Choose Report➛Design Properties.

2 In Design Properties—Report Settings, select Read from a file.

3 Type the name of the report object value (.rov) file that the report design reads at run time to get parameter values, as shown in Figure 17-25. You can also choose Browse to find and select the ROV.

Choose OK.

The next time that you run the report, e.Report Designer Professional generates a report using the parameter values specified in the ROV. Requester does not appear.

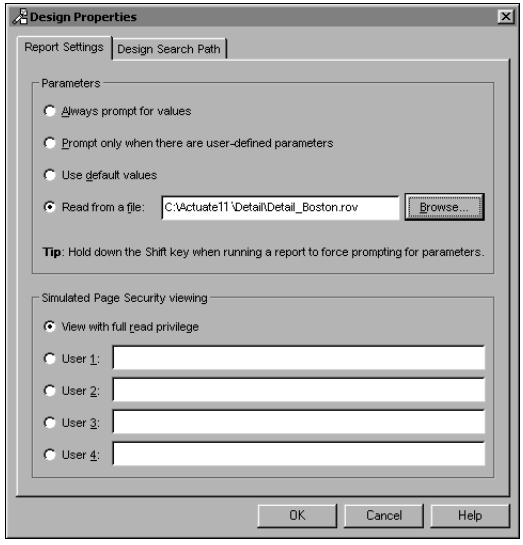


Figure 17-25 Design Properties—Report Settings

Suppressing parameter prompting

The default behavior for Requester is to prompt for parameter values when the report contains user-defined parameters. If you are in testing mode and do not want Requester to appear when you run the report, you can take one of the following steps:

- Specify that the report use default parameter values.
- Specify that Requester not appear.

When you specify either of these options, the default behavior is for Requester to not appear. To override this behavior for individual report runs, press the Shift key when you choose Report➤Build and Run. This action forces Requester to appear.

How to specify that the report use default parameter values

1. Choose Report➤Design Properties.
2. In Design Properties—Report Settings, select Use default values, then choose OK. When you run the report, it uses the default parameter values and Requester does not appear.

   To override this behavior and display Requester for a particular report run, press the Shift key when you choose Report➤Build and Run.

How to specify that Requester not appear

1. Choose Report➤Build and Run.
2. In Requester, select Do not prompt for parameter values. The next time that you run the report, Requester does not appear. To override this behavior and display Requester for a particular report run, press the Shift key as you choose Report➤Build and Run.

Verifying parameter values that the user supplies

e.Report Designer Professional does not verify the parameter values that a user supplies. If a user supplies an invalid value or does not supply a value for a required parameter, the generated report is empty. You can verify values that a user supplies by writing error-checking code.

The best place to write this code is in the Start( ) method of the report component. The report component is the top-level component in the report and it functions as the entry point to the report. The Start( ) method is called after the report object is created but before report generation begins, which makes it the ideal place to add error-checking code.

For more information about the report component and Start( ), see *Programming with Actuate Foundation Classes.*

The following code checks whether the user supplied a value for a required parameter. If the user did not supply a value, a message appears.

```
Sub Start( )
   Super::Start( )

   If DataSource::OfficeState = "" Then
      MsgBox "You must supply a value for Office State"
   End If

End Sub
```

DataSource::OfficeState is the fully scoped parameter name. You must use fully scoped names for parameters with local scope. All data-filtering parameters have local scope. To check the scope of a parameter, look at the parameter's properties.

How to write code that verifies parameter values

1. In Report Structure, right-click the top-level report component. Choose Properties.
2. In Properties, choose Methods.
3. In Methods, select Sub Start( ). Choose Override. The method editor displays the required lines of code for overriding the method.
4. Add the error-checking code to the method.

Chapter

18

Controlling user access through page-level security

This chapter contains the following topics:

- About page-level security
- Designing a report that uses page-level security
- Creating a virtual security ID
- Customizing an access control list

About page-level security

Page-level security is a technique for controlling user access to a report on a page-by-page basis. In a report that uses page-level security, a report user can view, search, and print only pages to which he has access.

Page-level security works by comparing the report user's security IDs (SIDs) to the access control list (ACL) for a page in the report. An ACL is a list of security IDs that pertain to a report component such as a section. If any of the report user's security IDs matches any security ID in the section's ACL, the report user has access to pages e.Report Designer Professional generates from the section.

For example, a report developer can control access to data by creating a report like the one shown in Figure 18-1. The report consists of six pages. Some report users can see all pages and some users can see only a subset of pages.

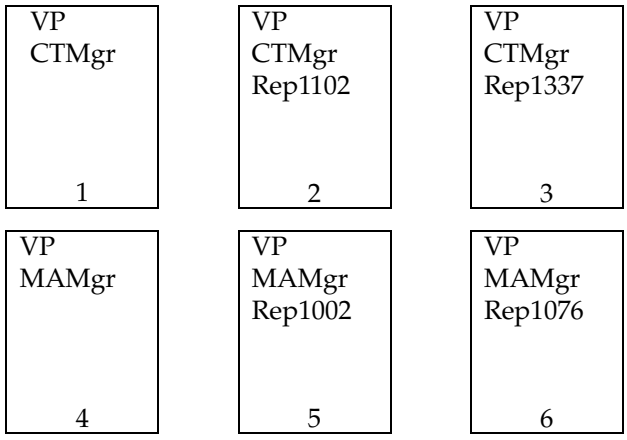


Figure 18-1 Example of a report with controlled access

This report contains a list of customers in a sales region. The VP user can access every page in the report and can see the name of every customer in the region. The state managers, CTMgr and MAMgr, can see only the names of customers in their states. CTMgr can access pages 1, 2, and 3. MAMgr can access pages 4, 5, and 6. Each sales representative can access only a single page.

Page-level security is available only for a report a developer publishes to Actuate iServer for viewing in DHTML format using the secure read privilege. If the report is on the client machine or the report user has the standard read privilege, page-level security is not in effect. For page-level security to take effect, the report user must have secure read privilege to the report. Secure read provides the ability to open, read, print, but not download or edit, a report in the Encyclopedia volume. Using page-level security requires the Actuate Page Level Security option on Actuate iServer.

About security IDs

A security ID provides information to Actuate iServer about a report user's access privileges to pages in a report that uses page-level security. A security ID can be based on any of the following identifiers:

- The user's name
- The user's Encyclopedia volume security roles
- A user name the Report Server Security Extension (RSSE) provides
  For more information about Report Server Security Extension, see *Using BIRT iServer Integration Technology.*
- A virtual security ID
  A virtual security ID can provide greater access restrictions to a section. For example, you can concatenate Encyclopedia volume security roles to create a virtual security ID.

Actuate iServer maintains a list of user security IDs, other than virtual security IDs, for the entire login session.

Defining page-level security rules

e.Report Designer Professional encodes a report's security rules as a list of security IDs. Typically, the developer supplies security IDs as an Actuate Basic expression in the Security➤GrantExp property of a section. When a report runs on Actuate iServer, the Factory calls the SetSecurity( ) method to evaluate the GrantExp expression and generate the access control list (ACL) for the section. Actuate iServer determines which pages to display by comparing the user's security IDs to the security IDs in each section's ACL.

GrantExp must be an Actuate Basic expression and must evaluate to a string. If necessary, use the CStr function to convert the expression to the String data type, as shown in Figure 18-2.

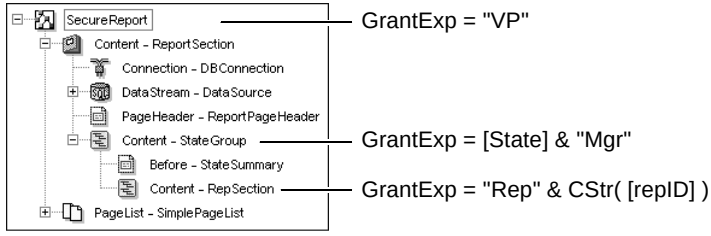


Figure 18-2 Page-level security structure

When GrantExp is blank, there are no security restrictions in place for the section, except those restrictions that the section inherits from its container sections.

To grant access to a section for all of a report user's security IDs, use a virtual security ID.

Understanding cascading security rules for nested sections

In a report that uses page-level security, the default behavior is that a section inherits its container's security IDs. If you nest a section within another section, the nested section's ACL includes its own security IDs and the security IDs for the container section. This behavior is called cascading security.

You set cascading security rules using the Security➤CascadeSecurity property of a section, as shown in Figure 18-3.

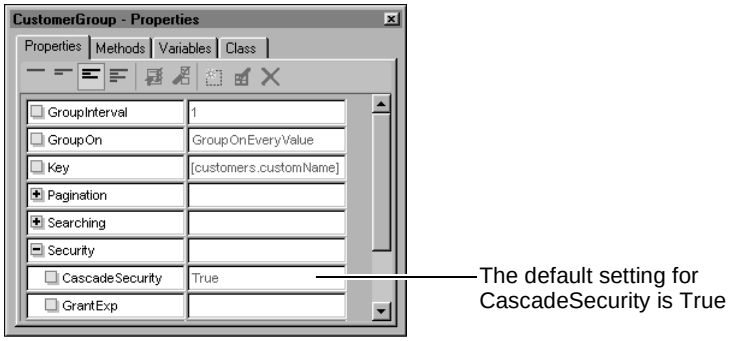


Figure 18-3 Cascading security rules

When the report runs, e.Report Designer Professional places a page break between sections that use different ACLs, ensuring that report users view only the pages their security IDs permit.

Designing a report that uses page-level security

To design a report that uses page-level security, you can perform the following tasks:

- Determine which report users can access each section of the report:
  - Determine the security requirements for the pages in each section.
  - Identify the security IDs that grant access to these pages.
  - Determine how to correlate these security IDs with values in the database that contains the report data.
- For each section, set Security➤GrantExp to return the security IDs required to view pages the section generates.

- Set Security➛CascadeSecurity to enable or disable cascading security rules.
- Number the pages in the report based on either the set of pages visible to the report user or the actual number of pages in the report.
- Test the report design by choosing Report➛Design Properties to access Design Properties—Report Settings, as shown in Figure 18-4.

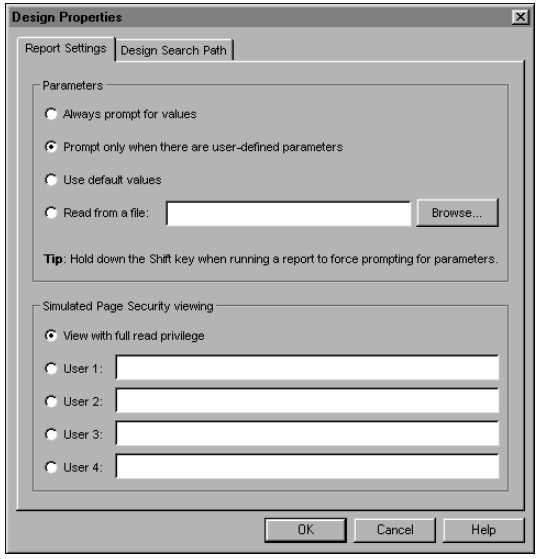


Figure 18-4 Design Properties—Report Settings

Using this page, you can assign security IDs to simulated users, then run the report to verify that each user can access the correct pages.

How to grant access to the pages in a section

1. Right-click a section in Report Structure, then choose Properties. The Properties page for the section appears.
2. Type a value for Security➛GrantExp.

If you do not see the property, choose Advanced Properties on the Properties page.

A valid value for GrantExp is an Actuate Basic expression that evaluates to a single security ID, a comma-separated list of security IDs, or a virtual security ID.

For example, as shown in Figure 18-5, type the following expression in GrantExp:

```
[offices.city] & "Office Manager"
```

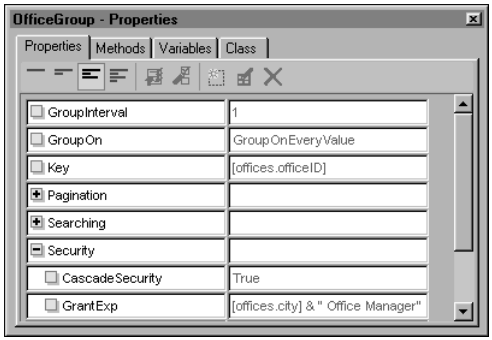


Figure 18-5 OfficeGroup—Properties

For each value of [offices.city] in the database, this expression returns a security ID that corresponds to a security role in the Encyclopedia volume. For example, for Boston, GrantExp returns:

```
Boston Office Manager
```

When the value of [offices.city] changes between rows, the ACL of the section changes and a new page begins.

Displaying page numbers in a report that uses page-level security

In a report that uses page-level security, you can number pages sequentially in the order in which they are visible to the report user. For example, if the report user can access pages 12, 23, 48, and 126 of a report, the page numbers visible to the user are pages 1, 2, 3, and 4.

You also can number pages relative to the total number of pages in the report. Relative numbering ensures that a page displays the same page number for every report user. Relative page numbering is useful when many users need a convenient way to refer to report pages.

To number the pages in a report sequentially, place a page number control in the page and set PageNumberType to VisiblePageNumber. This setting displays the page number relative to the set of visible pages for the report user.

To number report pages relative to the total number of pages, set the page number control's PageNumberType property to ActualPageNumber. ActualPageNumber displays the same page number on a page for every report user.

Testing a report that uses page-level security

You can test a report design that uses page-level security by creating a simulated report user and assigning the user one or more security IDs. When you run the

report in e.Report Designer Professional as a simulated user, you view only those pages accessible to the security IDs that you assign to that user.

This topic explains how to view a report without page-level security and how to view the same report as a simulated user to test the security settings. To clarify which pages the user is viewing, the illustrations in this topic show the ACL for the page, the user security IDs required to view the page, the visible page number, and the actual page number. By adding this information to a report design, you can verify that a user who should view a single page views only that page.

Setting up a report design to test page-level security

To test security rules, you compare the ACL of the page and the security IDs of the current user. This topic explains how to create controls that display this information.

How to display the ACL of the current page

1. In the page layout view, drag a label control from Toolbox—Controls and drop it in the page.
2. Change the label text to:

   ```
   Access Control List:
   ```

3. Drag a text control from Toolbox—Controls and drop it beside the label control. Component Properties appears.
4. As shown in Figure 18-6, set ValueExp to:

   ```
   GetPageList( ).GetCurrentPageACL( )
   ```

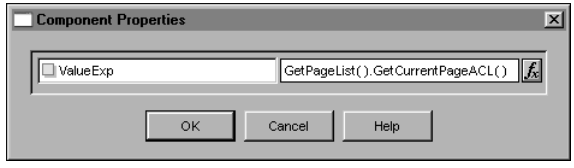


Figure 18-6 Component Properties—ValueExp

GetPageList( ).GetCurrentPageACL( ) returns the ACL for the current page in the list. This text control displays the security IDs that can access the current page.

5. Choose OK. The label and text control appear in the layout, as shown in Figure 18-7.

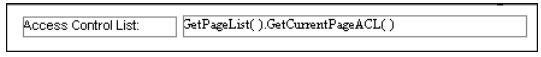


Figure 18-7 Label and text controls

How to display the report user's security IDs

1. Drag a label control from Toolbox—Controls and drop it in the page.
2. Change the label text to:

   ```
   Report User Security IDs:
   ```

3. Drag a text control from Toolbox—Controls and drop it in the page. Component Properties appears.
4. Choose OK without setting a value expression for this text control.
5. In the page layout view, right-click the text control, then choose Properties. The Properties page for the text control appears.
6. Choose Methods. The Methods page for the text control appears.
7. Override the text control's GetText( ) method as the following example shows:

   ```
   Function GetText( ) As String
      GetText = GetValue( GetReport( ), "CurrentUserACL" )
   End Function
   ```

   Overriding GetText( ) in this manner sets the text control's value to the value of the CurrentUserACL variable. If a user with the secure read privilege views this report, GetText( ) returns a comma-separated list of security IDs for the current user. The commas in the list represent the OR operator, so a user with any of the security IDs can access the page. If a user views the report with standard read privilege, GetText( ) returns nothing.

   Next, create the CurrentUserACL variable for GetText( ) to call.
8. Right-click the top-level report component, then choose Properties. The Properties page for the report component appears.
9. Choose Variables. The Variables page for the report component appears.

10. Choose New. Class Variable appears.
11. Create the variable CurrentUserACL, as shown in Figure 18-8.

    Choose OK.
12. Choose Methods. The Methods page for the report component appears.
13. Override the report component's GetUserACL( ) method to create a custom ACL for the current user:

    ```
    Function GetUserACL( acl AS String ) As String
       GetUserACL = Super::GetUserACL( acl )
          CurrentUserACL = GetUserACL
    End Function
    ```

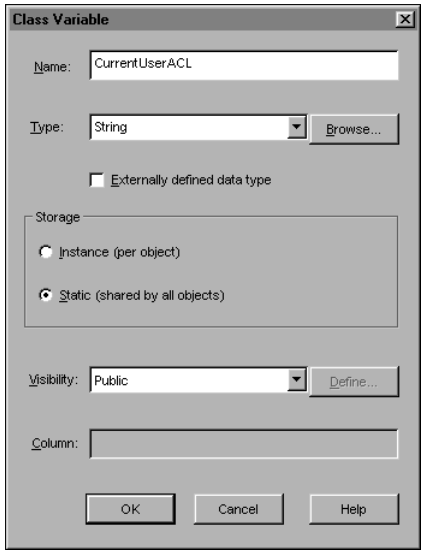


Figure 18-8 Creating a class variable

In the layout view, the controls look like the ones in Figure 18-9.

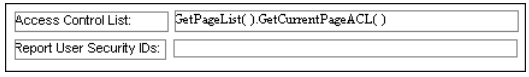


Figure 18-9 Creating a custom ACL

How to display visible and actual page numbers

When you create a blank report, e.Report Designer Professional places a page number control at the bottom of the page. This default page number control's PageNumberType property setting is VisiblePageNofM.

1. In the layout window, scroll to the bottom of the page.
2. Move the default page number control up and to the right, as shown in Figure 18-10.

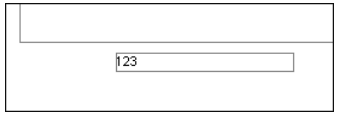


Figure 18-10 Adjusting the default page number control

3. Change the control's PageNumberType property to:

   `VisiblePageNumber`

4. Place the following controls at the bottom of the page:

- A label control containing the following text:

  `Visible:`

  Place this control beside the default page number control.
- A label control containing the following text:

  `Actual:`
- A page number control with PageNumberType set to:

  `ActualPageNumber`

5 Align the controls to look like those in Figure 18-11.

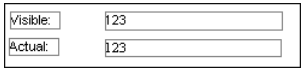


Figure 18-11 Aligning page number controls

Testing page-level security

In this topic, you first run a report without page-level security. Then you run the same report as each of two simulated report users. In these procedures, User 1 has a single security ID, MAMgr. User 2 has a single security ID, Rep1102. By comparing the pages these users can view, you can determine whether the security rules are correct.

How to view a report without page-level security

1 Choose Report➛Design Properties. Design Properties—Report Settings appears.

2 To view the report without page-level security, select View with full read privilege, as shown in Figure 18-12.

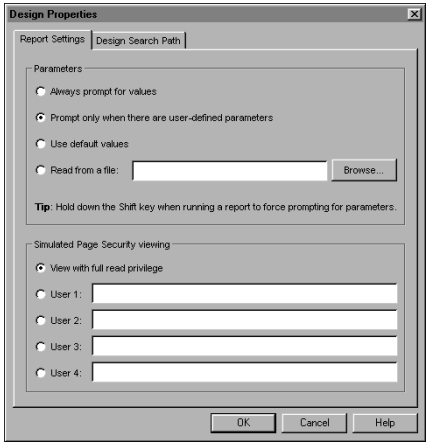


Figure 18-12 Viewing a report without page-level security

Choose OK.

3 Run the report. When you view the report, the visible page number is the same as the actual page number, as shown in Figure 18-13.

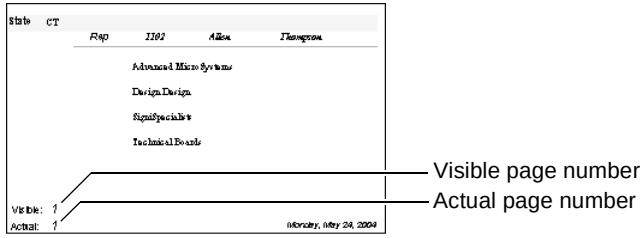


Figure 18-13 Displaying visible and actual page numbers

How to test page-level security as a simulated user

1 Choose Report➛Design Properties. Design Properties—Report Settings appears.

2 To view the report as User 1, select User 1.

3 Type a security ID for this simulated user, as shown in Figure 18-14.

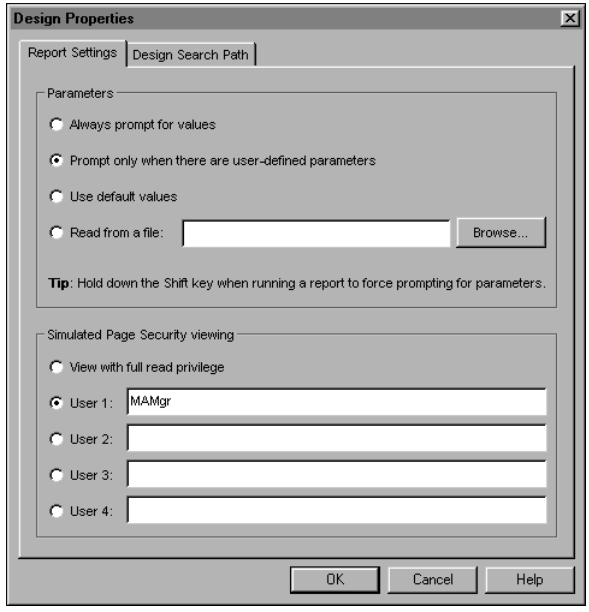


Figure 18-14 Page-security viewing as simulated User 1

Choose OK.

4 Run the report. The report contains only pages for which the access control list includes the security ID MAMgr. The security ID appears at the top of each page. The visible page number is less than the actual page number, as shown in Figure 18-15.

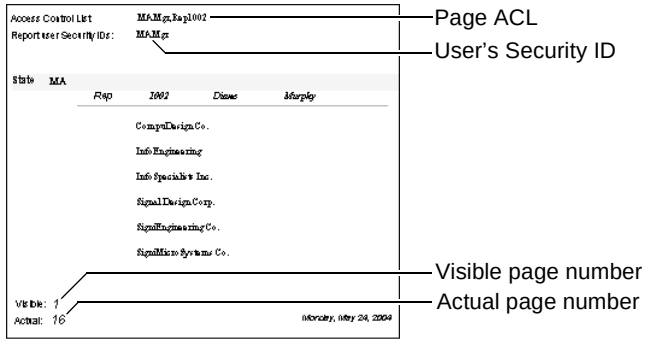


Figure 18-15 Viewing the report with page-level security as User 1

5 To view the report as User 2, select User 2.

6 Type a security ID for this simulated user, as shown in Figure 18-16.

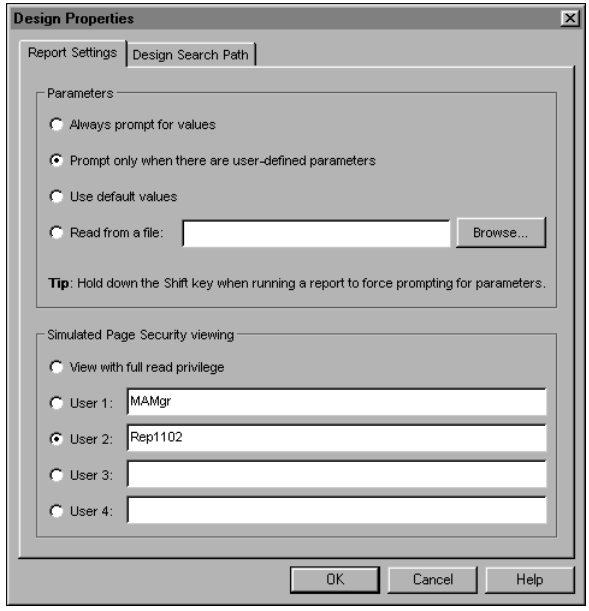


Figure 18-16 Page-security viewing as simulated User 2

Choose OK.

7 Run the report. User 2's security ID appears at the top of the page. The report contains only the page for which the access control list includes the security ID Rep1102.

The visible page number is less than the actual page number, as shown in Figure 18-17.

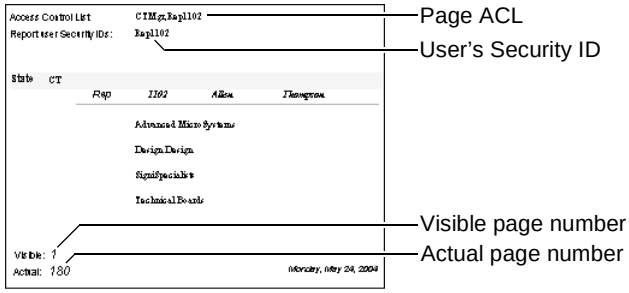


Figure 18-17 Viewing the report with page-level security as User 2

Creating a virtual security ID

You can refine the security rules for a section by concatenating security roles, user names, or other identifiers in a virtual security ID. When you create a virtual security ID, you tighten restrictions on the page.

For example, a page displays total sales for sports utility vehicles. The sales manager role and product category role are security roles in the Encyclopedia volume. By combining these roles in a virtual security ID and setting GrantExp to the virtual security ID, you make the page accessible only to sales managers who are responsible for selling sports utility vehicles.

A virtual security ID is valid only for the report that defines it. You cannot store a virtual security ID in the Encyclopedia volume.

Creating a virtual security ID requires the following steps:

- Override GetUserACL( ) to create the virtual security ID and include it in the user's ACL.
- Set up the page-level security restrictions on the section. To restrict access to pages, set the section's GrantExp property to the virtual security ID.

How to build a virtual security ID

The following code shows how to build the virtual security ID and assign the result to a variable, in this case the ACL variable. Note that when you build the virtual security ID, you ensure that the comma in the ID reads as an AND operator. This step is necessary because in the comma-separated GrantExp list,

the commas read as OR operators. Note, too, that this example assumes a user has only one security ID of the type Manager.

```
Function GetUserACL( acl As String ) As String
   Dim tail As String
   Dim mgrSecurityID  As String
   Dim posn As Integer
   Dim securityID As String

' Build the virtual security ID to include Manager and
   ' Product Category combination IDs.

' Loop to get a Manager security ID.
   tail = acl
      Do While tail <> ""
      posn = InStr( tail, "," )
      If posn = 0 Then
         SecurityID = Trim$( tail )
            tail = ""
      Else
         SecurityID = Trim$( Left$( tail, posn - 1 ) )
         tail = Trim$( Mid$( tail, posn + 1 ) )
      End If

   ' Check whether the security ID is a manager ID.

   If InStr( SecurityID, "Manager" ) > 0 Then
      mgrSecurityID = SecurityID
   End If
Loop

   ' If user is a manager, loop to get all Product security IDs.

   If mgrSecurityID <> "" Then
      tail = acl
      Do While tail <> ""
      posn = InStr( tail, "," )
         If posn = 0 Then
            SecurityID = Trim$( tail )
            tail = ""
         Else
            SecurityID = Trim$( Left$( tail, posn - 1 ) )
            tail = Trim$( Mid$( tail, posn + 1 ) )
         End If
' Check whether the security ID is a product ID.
   If InStr( SecurityID, "Product Category" ) > 0 Then
      ' Build the virtual security ID and add it to the list.
      ' Ensure that the comma in the virtual security ID
      ' reads as an AND operator. Typically, the comma in
      ' GrantExp list reads as an OR operator.
```

```
   acl = acl & ", " & mgrSecurityID & " " & SecurityID
   End If
   Loop
   End If
   GetUserACL = acl
End Function
```

Customizing an access control list

You can customize the access control list of a section to accomplish the following tasks:

- Control access to subsections within a container section.
- Generate a custom ACL.

Controlling access to subsections

By default, a subsection inherits security IDs from the section that contains it. Access to a container section grants access to all its subsections. This behavior is called cascading security. To grant access to the container section only, use one of the following techniques:

- Set the Security➛CascadeSecurity property to False. CascadeSecurity determines whether a subsection inherits security IDS from its container section.
- Override GetFullACL( ) for the subsection. The default behavior for GetFullACL( ) is to generate the complete ACL for a subsection, including the ACLs for any of its containers.

Setting CascadeSecurity

The default setting for the Security➛CascadeSecurity property of a section is True, as shown in Figure 18-18. This setting ensures that a subsection inherits security IDs from its container section. A report user who can access the pages in a container section can also access the pages in any subsections within that section.

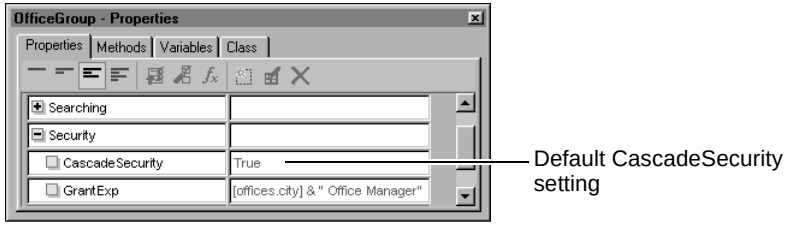


Figure 18-18 Setting cascading security

When CascadeSecurity is False, the access control list for a subsection contains only the security IDs in the subsection's GrantExp property.

Overriding GetFullACL( )

When the Security➛CascadeSecurity property of a section is True, a subsection calls GetFullACL( ) for its container section. Then, the subsection appends its ACL to the ACL for the container section. In this manner, a report user who can access pages in the container section also can access pages in the subsection.

To disable cascading security, you can override GetFullACL( ) for the subsection. If you override GetFullACL( ) as shown in the following example, the access control list for the subsection contains only the security IDs the subsection's GrantExp property specifies:

```
Function GetFullACL( ) As String
   GetFullACL = GetComponentACL( )
End Function
```

Generating a custom ACL

Typically, you set the access control list for a page by using a section's Security➛GrantExp property to link a security role in the Encyclopedia volume to a database value. For example, the GrantExp expression:

```
[offices.city] & " Office Manager"
```

links the value of [offices.city] to the Office Manager security role. This expression grants users in the Office Manager role access to data about only the offices they manage.

It is also possible to create a custom ACL using either of the following methods:

- Overriding the SetSecurity( ) method for a section and ignoring the value of GrantExp
- Creating a custom function and setting Security➛GrantExp to call the function

Generating a custom ACL using SetSecurity( )

The Factory calls SetSecurity( ) to generate an access control list for a section. Typically, SetSecurity( ) returns the security IDs that Security➛GrantExp defines for the section. You can build a custom access control list by overriding SetSecurity( ) on a section. Then, set the ACL variable to a single security ID or a list of security IDs separated by commas. When you override SetSecurity( ), the Factory ignores the value of GrantExp for the section.

The following example uses the AccountType property of the current data row to define the data values and the ACL variable to define the security role. If no security restrictions exist, set an empty ACL variable.

```
Sub SetSecurity( row As AcDataRow )
   Dim myRow As MyDataRow
   Set myRow = row

   If myRow.AccountType = "Commercial" Then
      ACL = "MajorAccts"
   ElseIf myRow.AccountType = "Private" Then
      ACL = "PrivateBanking"
   Else
      ACL = "Accounting"
   End If
End Sub
```

Generating a custom ACL using an Actuate Basic function

You can create an Actuate Basic function to map database values to security roles when the security roles do not match values in the database. Then, set Security➤GrantExp for the section to call this method.

For example, you can create a MapAcctType( ) function to map account type values from the current data row to security roles, as shown in the following example:

```
Function MapAcctType( row As AcDataRow ) As String
   Dim myRow As MyDataRow
   Set myRow = row

   If myRow.AccountType = "Commercial" Then
      MapAcctType = "MajorAccts"
   ElseIf myRow.AccountType = "Private" Then
      MapAcctType = "PrivateBanking"
   Else
      MapAcctType = "Accounting"
   End If

End Function
```

Then, set the section's GrantExp property to call this function, as shown in Figure 18-19.

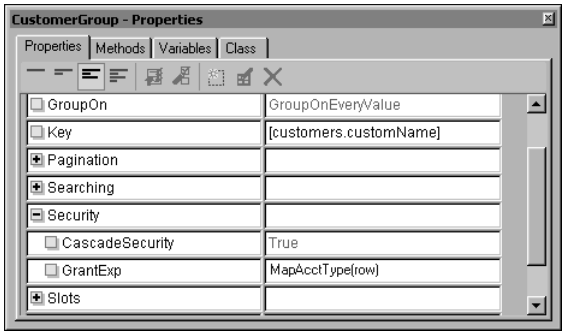


Figure 18-19 Generating a custom ACL using Actuate Basic

Chapter

19

Designing a master report to generate multiple reports

This chapter contains the following topics:

- About report bursting
- Understanding report bursting techniques
- Understanding report bursting tasks

About report bursting

Report bursting is the process of creating a single report object executable (.rox) file that generates multiple smaller report object instance (.roi) files. Generating many small reports has the following advantages:

- You can set privileges for each report by placing the report in a folder with the appropriate privileges.
- A report user can download and view several small files offline more easily than one large file. The size of a report can affect the overall performance of Actuate iServer. A report that contains more than 10,000 pages requires a longer report generation time than a short report.
- It is often easier for a report user to find data by locating a short report in an Encyclopedia volume than by navigating a large report.
- If all reports in a folder use the same data source query, the query executes only once.

Actuate BIRT iServer supports report bursting only if the report runs as a scheduled background job, or asynchronously. If you run a burst report synchronously, Actuate BIRT iServer displays an error message because it cannot determine which report to display.

Understanding report bursting techniques

To use the report bursting techniques described in this section, your report design must include a master report. The master report controls the report generation process. Override the RoiIsTemporary( ) method to delete the master report after generation.

Use one of the following techniques to create the detail reports. A report user views each detail report in a different report object instance (.roi) file.

- Place a sequential section in the master report. Then, place the detail reports in the sequential section.
- Use a report and subreport structure.
- Burst the report on group boundaries.

Understanding report bursting tasks

To create a report that uses bursting, you must perform the following tasks:

- Create the detail report by adding a second subclass of AcReport to a report design. The detail report appears in a separate report object instance (.roi) file.
- Add a page list and a page to the detail report.
- Provide a name for the detail report by overriding the detail report's SuggestRoiName( ) method.

The following additional tasks are optional:

- Create a hyperlink from the master report to the detail report.
- Pass data from the master report to the detail report.
- Place the detail report in a different folder.

How to add a second subclass of AcReport

To add a second subclass of AcReport to a report design, drag a report component from Toolbox—Structure and drop it in the appropriate Content slot in Report Structure.

Examining report bursting examples

The Examples directory that ships with e.Report Designer Professional contains two report designs that use report bursting, BurstSubreports.rod and BurstGrouping.rod. Both designs use the Sfdata database and both generate the same master report and detail reports. BurstSubreports.rod uses subreport bursting, and BurstGrouping.rod uses group bursting.

The master report lists the state locations for MultiChip Technologies' customers. Each state is a hyperlink to a separate detail report that lists the customers in that state. The master report serves as an index to the detail reports.

The report user can navigate from the master report to the detail reports using the hyperlinks, as illustrated in Figure 19-1.

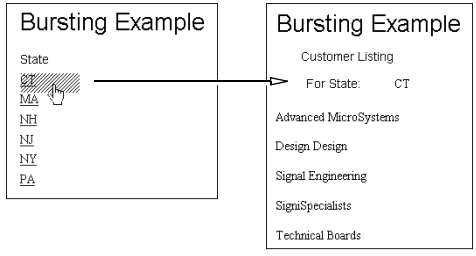


Figure 19-1 Example of report bursting

Examining a subreport bursting report

To use subreport bursting, you place each subreport in a different report object instance (.roi) file.

The report design in Figure 19-2 uses a report and subreport structure. The subreport, or detail report, is in a sequential section.

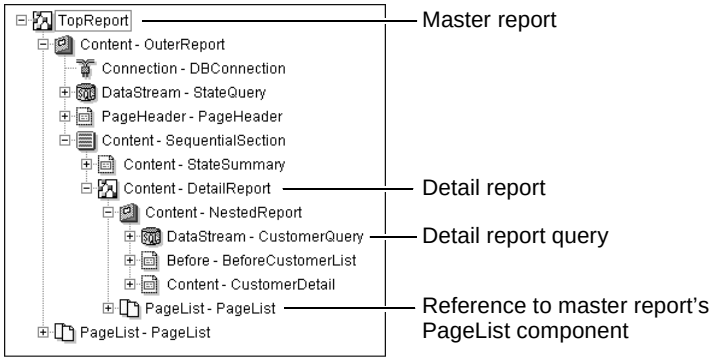


Figure 19-2 Report structure for subreport bursting

The detail report's PageList component is a reference to the master report's PageList component.

The detail report's query, CustomerQuery, has a parameter using the value of the summary entry, state.

For CustomerQuery, the Conditions tab of Query Editor looks like the one shown in Figure 19-3.

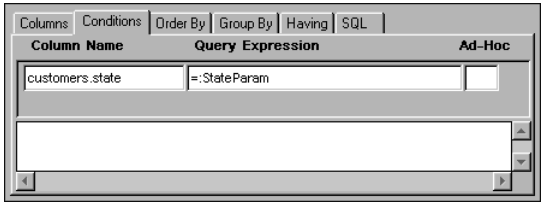


Figure 19-3 Query Editor—Conditions

Query Editor—SQL looks like the one in Figure 19-4.

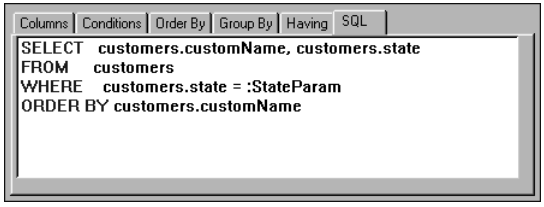


Figure 19-4 Query Editor—SQL

Examining the BuildFromRow( ) method for a subreport bursting report

The BuildFromRow( ) method of the detail report's section, NestedReport, is overridden to set the detail query's parameter, StateParam, to the value of the summary entry, state in the following code:

```
Function BuildFromRow( row As AcDataRow ) As AcBuildStatus
   If Not row Is Nothing Then
      CustomerQuery::StateParam = row.GetValue( "state" )
   End If
   BuildFromRow = Super::BuildFromRow( row )
EndFunction
```

Examining the SuggestRoiName( ) method for a subreport bursting report

The detail report's SuggestRoiName( ) method is overridden to give each detail report a unique name. The value of the summary entry, state, creates the name.

Note that there are two versions of the SuggestRoiName( ) method. One version takes no arguments. The other version takes a data row. The following example uses the version that takes a data row:

```
Function SuggestRoiName( row As AcDataRow ) As String
   SuggestRoiName = "State_" & row.GetValue( "state" ) & " .roi"
EndFunction
```

If the report design generates a large number of reports, place the reports in different folders. To place the reports in different folders, include a folder in the name returned from SuggestRoiName( ). For example, if the report design generates reports for counties in different states and you want to place the reports for each state in a different folder, override SuggestRoiName( ) as follows:

```
Function SuggestRoiName( row As AcDataRow ) As String
   SuggestRoiName =   row.GetValue( "state" ) & "/" &
                      row.GetValue( "county" ) & ".roi"
End Function
```

Examining the summary entry's LinkExp property for a subreport bursting report

The summary entry's Linking➤LinkExp property is set to create hyperlinks from the master report to the detail reports, as shown in Figure 19-5.

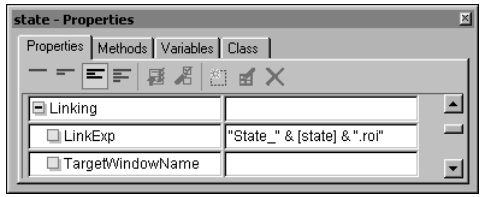


Figure 19-5 Defining linking properties for a subreport bursting report

Examining a group bursting report

Group bursting divides the report on group boundaries and places each group in a different report object instance (.roi) file, as shown in Figure 19-6.

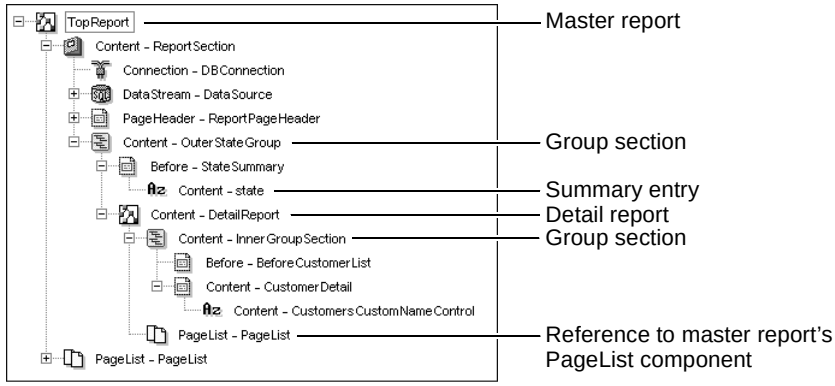


Figure 19-6 Report structure for group bursting

This report design contains two group sections. The first group section, OuterStateGroup, is in the master report. OuterStateGroup's Before slot contains the summary entry for the group, state. The second group section, InnerGroupSection, is in the detail report. Both group sections have the same key, [customers.state].

The detail report's PageList component is a reference to the master report's PageList component.

Examining the SuggestRoiName( ) method for a group bursting report

Override the detail report's SuggestRoiName( ) method to give each detail report a unique name. The value of the summary entry, state, creates the name.

Note that there are two versions of the SuggestRoiName( ) method. One version takes no arguments. The other version takes a data row. The following example uses the version that takes a data row:

```
Function SuggestRoiName( row As AcDataRow ) As String
   SuggestRoiName = "State_" & row.GetValue( "state" ) & ".roi"
End Function
```

If a report design generates a large number of reports, place the reports in different folders. To place the reports in different folders, include a folder in the name returned from SuggestRoiName( ). For example, if your design generates reports for counties in different states and you want to place the reports for each state in a different folder, override SuggestRoiName( ), as shown in the following example:

```
Function SuggestRoiName( row As AcDataRow ) As String
   SuggestRoiName =   row.GetValue( "state" ) & "/" &
                    row.GetValue( "county" ) & ".roi"
End Function
```

Examining the summary entry's LinkExp property for a group bursting report

The summary entry's Linking➤LinkExp property is set to create hyperlinks from the master report to the detail reports, as shown in Figure 19-7.

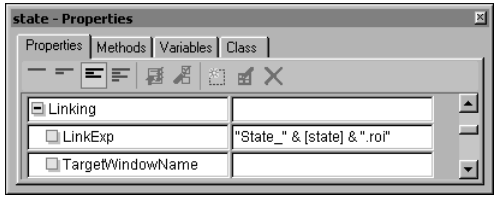


Figure 19-7 Defining properties for a group bursting report

Part Four

Optimizing report development tasks for the enterprise environment

Chapter

20

Designing reusable components

This chapter contains the following topics:

- About designing reusable components
- Working with custom properties
- Working with structures
- Working with enums

About designing reusable components

You can create custom components for use in report designs. Typically, you create a custom component in a report design that you use as a test framework. After you have developed the component, you can publish it to a component library to support its use in other report designs.

As well as the functionality described in earlier chapters, Actuate e.Report Designer Professional, the Actuate Foundation Classes, and Actuate Basic support the following capabilities:

- Creating a custom property
- Creating a custom structure data type using the Types tool in e.Report Designer Professional
- Creating and using an enum in Actuate Basic and e.Report Designer Professional

Working with custom properties

In e.Report Designer Professional, you can create, edit, and delete custom properties for a class. A custom property that you create can be of any data type, including an Actuate Basic data type, an AFC enum (enumeration) or structure, or a custom enum or structure. When you set the data type of a custom property to a structure, e.Report Designer Professional displays the property as an expandable group of values. When you set the data type of a custom property to an enum, e.Report Designer Professional displays the property as a drop-down list of values.

You can also set a custom property as a value expression, similar to the ValueExp expression for a data control. The data type of a custom property that you set as a value expression must be a numeric or string data type, an enum, or other data type that is equivalent to a numeric or string data type. For example, you can set a property to be a value expression if its data type is Integer or AcTwips, but not if its data type is AcFont or Date.

When you create or edit a custom property, you can specify filtering the property at any class level. You can also change the filtering level of an existing property. For example, for a data source, you can change the filtering level of DisplayName from Expert Properties to Commonly Used.

Creating, editing, and deleting a custom property

To create, edit, or delete a custom property, use the buttons on the toolbar of the Properties pages, as shown in Figure 20-1.

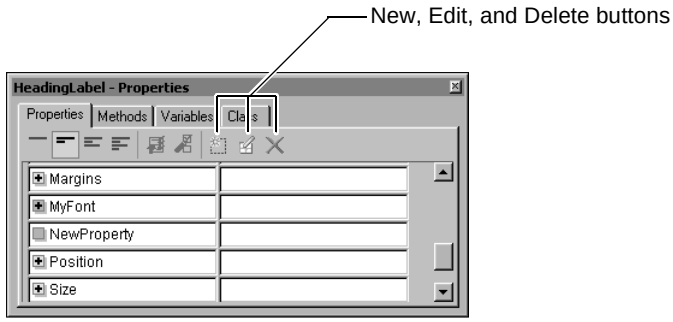


Figure 20-1 Buttons that support working with custom properties

The value for a custom property is stored in a variable that has the same name as the custom property, as shown in Figure 20-2. The icon for a custom property appears in a different color from inherited properties of a class.

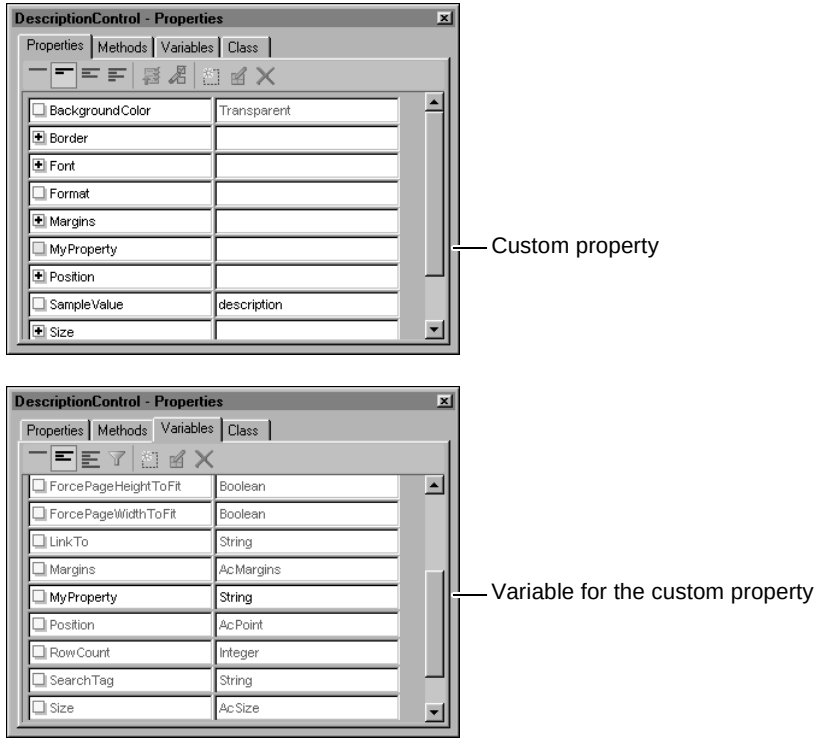


Figure 20-2 A custom property and its corresponding variable

When you create a property for a class, such as MyImageControl, then you create a subclass of MyImageControl, the subclass inherits the custom property. You can save custom classes in libraries for use throughout the enterprise. For example, Figure 20-3 shows the custom properties of a line control in a library. All custom

properties of the line control appear on the Properties page, on which a blue icon appears beside each custom property's name.

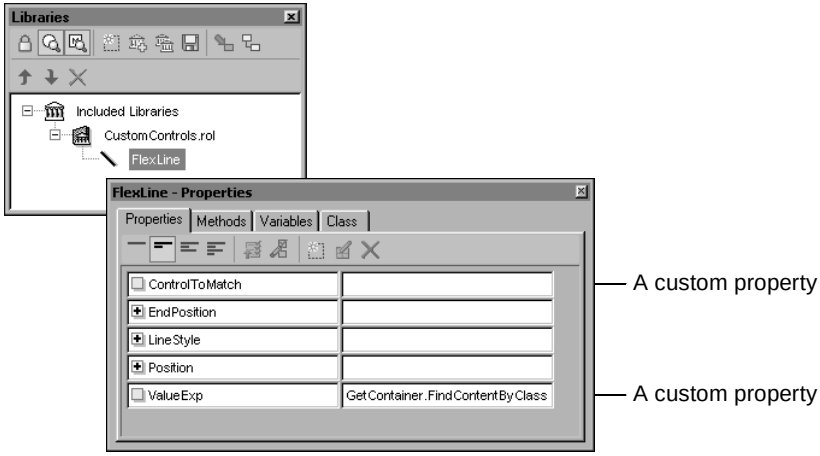


Figure 20-3 Custom properties of a line control in a library

When you add this line control from the library to a report design or subclass the control, the appearance of the custom property changes to indicate that the property definition is in a different module. The custom property appears and behaves like any other inherited property. The filtering settings that you apply in the custom property's definition or in a library apply in the reference or subclass. As shown in Figure 20-4, on the Properties page for this control, the custom property, ControlToMatch, appears with a yellow icon beside it. Also, because the report developer who created the custom property, ValueExp, for this line control set its visibility to Hidden, the ValueExp property does not appear on the Properties page.

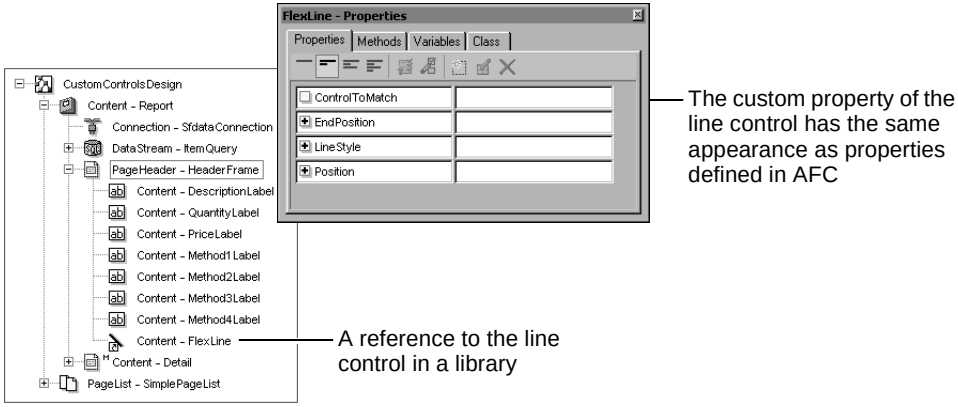


Figure 20-4 Properties of a reference to a line control in a library

Creating a property

When you create a property, you assign it the following settings:

- Name
  Use Actuate Basic conventions for a name.
- Property group
  You can assign a new property to an existing property group. This assignment places the property in a group on the Properties page. There is no default property group for a new property, and you do not need to assign a property to a group.
  You also can create a new group for the property. The new group appears on the Properties page with the new property.
- Value expression
  When you create a property for a control, you also indicate whether the property is a value expression. For more information about value expressions, see "Setting a property as a value expression," later in this chapter.
- Type
  The data type for a new property can be an Actuate Basic type, an AFC class, or a custom enum or structure.
- Visibility
  The visibility of a property determines the filtering level for a property or a structure. The visibility options are:
  - Essential
    e.Report Designer Professional prompts for essential properties when you place a component in a report design.
  - Commonly used
    Specify this setting to display the new property in the list of properties that report designers typically set, such as the background color and borders properties of a control.
  - Advanced
    Specify this setting to display the new property in a list of properties that includes more specialized formatting or modifications, such as dynamic size and position properties.
  - Expert
    Specify this setting to display the new property in a list of properties that includes properties that only experienced developers should use, such as the ObjectVariable property of a control or frame.

- Hidden

  Hidden properties do not appear on the Properties page of a subclass. Use this setting for a property when you do not want a developer using the class in a report design to change the functionality that the property provides.

How to create a custom property

1 In Report Structure, select the class to which the property belongs.

2 In the Properties page for the class, choose New. New Property appears, as shown in Figure 20-5.

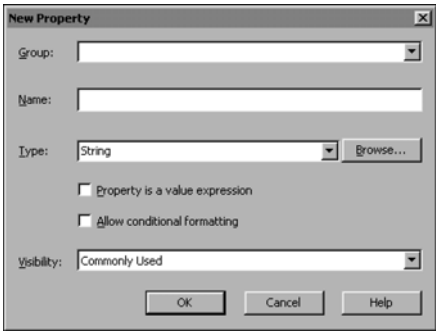


Figure 20-5 The New Property dialog box

3 Set the following attributes of the property:

- Optionally, to assign this property to an existing property group, select a group from the Group drop-down list. To assign this property to a new property group, type a group name in this field.
- In Name, type a property name. Use Actuate Basic conventions for a name.
- Select the data type for the property. Available Actuate Basic types are Boolean, Double, Integer, and String. You also can type the name of a class, enum, or structure, such as AcFont.

  Alternatively, choose Browse to display the Types dialog box. In Types, you can select a custom enum or structure.
- Indicate whether the property is a value expression, which is available for particular types.
- Indicate whether to allow conditional formatting for this property.
- Select a Visibility setting.

Choose OK.

The custom property appears in Properties. A custom property has a blue icon beside it, as shown in Figure 20-6.

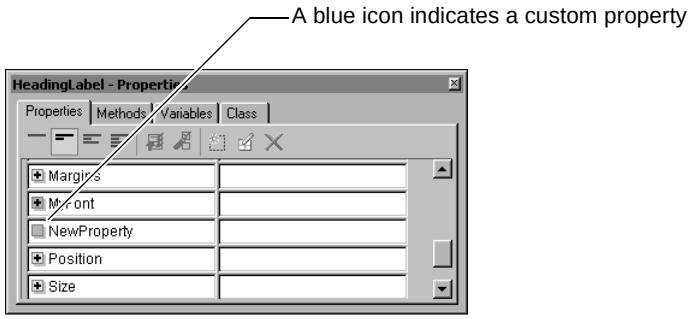


Figure 20-6 A custom property

To use the custom property, set a value for it in the class in the same way as you set a value for any other property.

Editing a custom property

You can edit the attributes of a custom property, including the property name, group, type, and visibility. There are inherent risks to editing a property. For example, if you define a property as an Integer and change it to a Date, any custom code in a method that uses the property can fail to compile. A name change creates similar risks.

How to edit a custom property

1 In the Properties page, choose Edit. Edit Property appears, as shown in Figure 20-7.

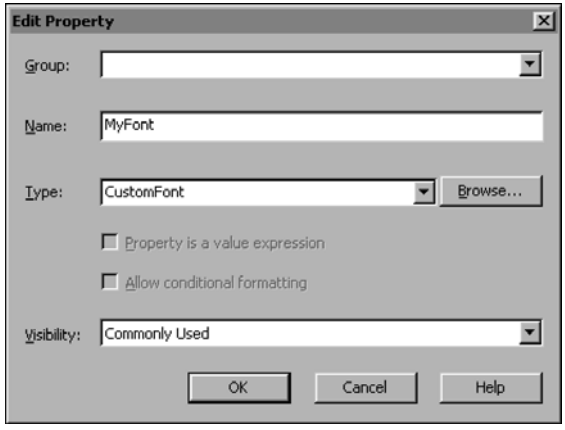


Figure 20-7 The Edit Property dialog box

2 Change the attributes. Choose OK.

Deleting a property

You can delete only a custom property. In the Properties page, select the property and choose Delete. If you delete the only property in a property group, e.Report Designer Professional also deletes the property group.

Setting a property as a value expression

A value expression is a set of values and operators that evaluates to one of the Actuate Basic data types. The AFC properties ValueExp and LinkExp are examples of value expressions. In LinkExp, for example, you use an expression to define a hyperlink's target.

When you create a new property for a control, you can set the property as a value expression when you set the property's other attributes.

You can set only a custom property of a scalar type as a value expression. If you set a member of a custom structure property as a value expression, the setting affects only the member. The root of the structure cannot become a value expression.

To set an existing custom property as a value expression, use the Value Expression icon, as shown in Figure 20-8. When you select a property that is set as a value expression, the Value Expression button appears beside the property field.

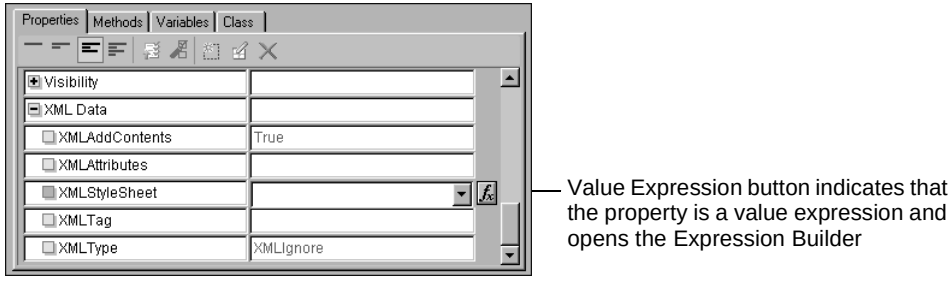


Figure 20-8 The Value Expression icon and button

When you set a property as a value expression, e.Report Designer Professional determines whether the control uses a single data row or multiple rows. When the expression in the expression property is an aggregate expression, such as Sum([Extprice]), e.Report Designer Professional processes multiple rows to calculate the property's value. When the expression in the expression property is a non-aggregate expression, such as [Extprice], e.Report Designer Professional uses a single row to calculate the value. If you need to change a control's behavior from this default so that it processes multiple rows, change the value of the ValueType property from AutoValueControl to SummaryControl.

How to modify a custom property to set it as a value expression

1 In the Properties page, select the custom property that you want to modify. The Value Expression button appears beside the field for the property.

2 Choose the Value Expression button.

3 In Expression Builder, create the expression.

This expression uses the same syntax as a ValueExp expression. It can be a simple expression or an aggregate.

Choose OK. The expression appears in the field beside the property name.

How to remove the value expression setting from a custom property

1 To remove the value expression setting from a custom property, select the property on Properties. The Value Expression button appears beside the field for the property.

2 In Expression Builder, delete the expression. Choose OK. The expression does not appear on the Properties page.

3 Choose the Value Expression button.

The property remains on the Properties page but the expression no longer applies to the custom property.

Filtering a property

You can set the visibility of any property, so that it appears at a particular filtering level on the Properties page. You set this attribute when you create a custom property. To change the visibility of any property, including properties defined by AFC, choose Edit. If you set the Visibility of a property to Hidden, as shown in Figure 20-9, the property does not appear on the Properties page of a subclass or reference. Use the Hidden setting for a custom property to set the value of the property on a component in a library. When a report developer subclasses the component in a report design, no changes to the hidden property are possible.

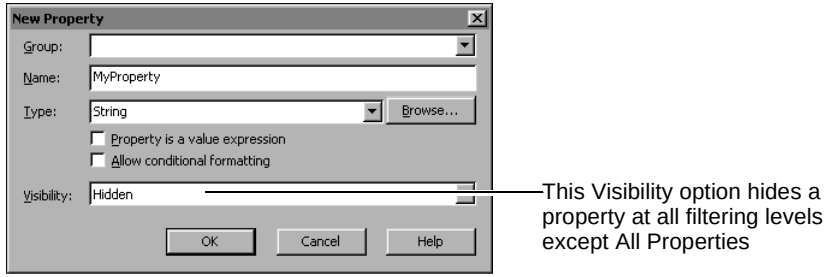


Figure 20-9 Setting a new property's visibility to Hidden

Working with structures

A structure is a group of member variables that describe a single item. Each structure member can be an Actuate Basic data type, such as an Integer or Boolean, or an AFC scalar type, such as AcColor.

A structure and its members appear on the Properties page if the property is defined as being of that structure type. For example, in Figure 20-10, Font is a structure that is composed of a number of members.

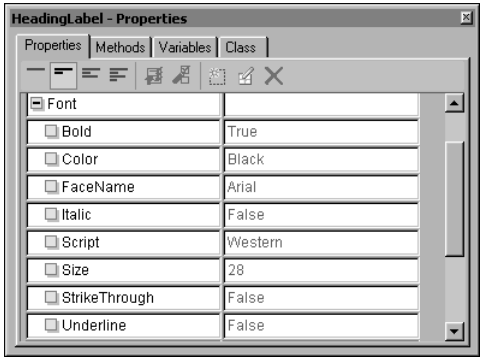


Figure 20-10 The Font structure and its members

You can create structures in e.Report Designer Professional and save them in the report design or a library file. The definition of a structure appears in the Project window when the Types filter includes structure aliases. Structures appear as Type components in the hierarchy, as shown in Figure 20-11.

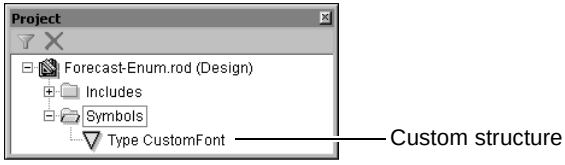


Figure 20-11 A structure as a Type symbol in Project

You can use custom structures in the same way as any AFC structure type. For example, you can use a custom structure as the data type for a custom variable or property on a report component. You can use a custom structure in the code of overridden methods.

How to create a structure

1 With a report design or library open, choose Tools➛Types. Types appears, as shown in Figure 20-12. Using this dialog box, you can create and delete enums (enumerations) and structures.

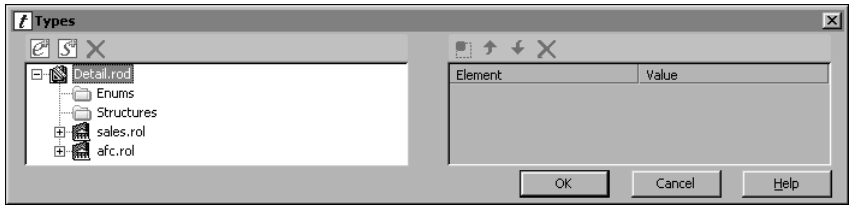


Figure 20-12 The Types dialog box

2 Choose Add Structure. A structure icon and field appear beneath the Structures folder, as shown in Figure 20-13.

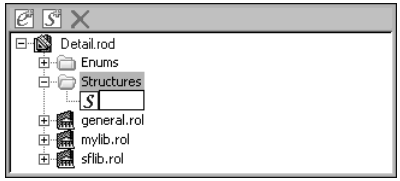


Figure 20-13 The structure icon and field

3 In the structure field, type a name for the structure. Click the icon beside the structure field.

The pane at the right of the dialog box becomes active, as shown in Figure 20-14. In this pane, you add members to the structure and assign types to them.

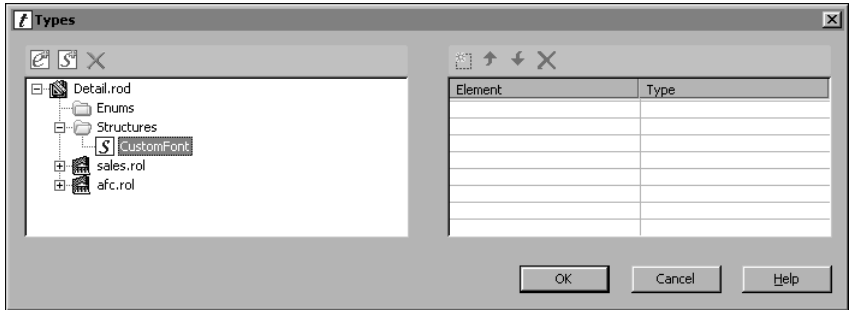


Figure 20-14 Use the pane at the right to create structure members

4 Choose New. The first Element field becomes active.

5 Type a name for the first structure member. Use Actuate Basic conventions for a name.

6 Place the cursor in the first Type field. Type or select a valid data type for the member.

7 Create as many member variables as the structure requires. When you finish, choose OK.

How to use a custom structure as a property

1 In the report design, select the class to which you want the structure to apply. For example, select a text control.

Choose New. New Property appears, as shown in Figure 20-15.

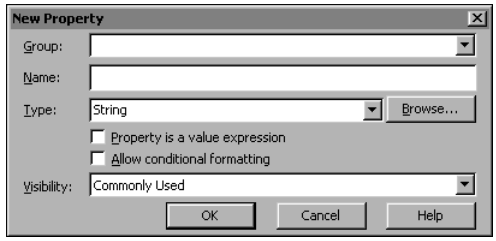


Figure 20-15 The New Property dialog box

2 Optionally, assign the property to a group.

3 In Name, type a name for the property. Use Actuate Basic conventions for a name.

4 In Type, choose Browse.

5 In Types, expand the Structures folder.

6 Select the structure that you want to use. Choose OK.

7 The structure appears in the Type field on New Property, as shown in Figure 20-16.

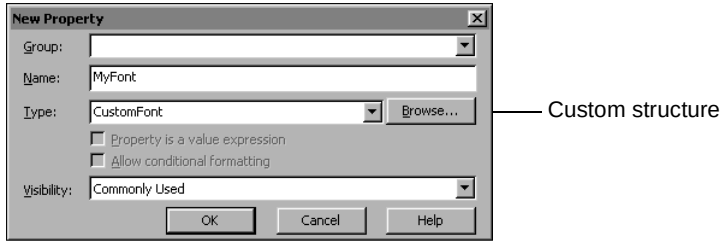


Figure 20-16 A custom structure on New Property

8 Select a visibility setting. Choose OK. The custom structure property appears on the Properties page for the specified class.

9 Expand the structure to display its member variables, as shown in Figure 20-17.

You can apply the structure to the class by setting values for one or more members.

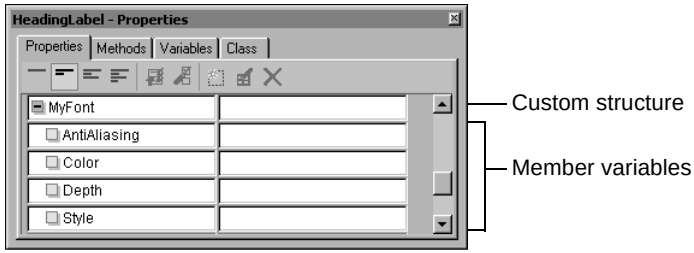


Figure 20-17 A custom structure and its members on Properties

Working with enums

An enum (enumeration) is a data type whose value is one of a set of named values. For example, the value of a TrafficLightColor enum can be only Red, Yellow, or Green.

e.Report Designer Professional displays an enum property as a drop-down list of values.

In addition to creating an enum in Actuate Basic, you can use the graphical user interface in e.Report Designer Professional for this functionality. The scope of an enum you create using this interface is global to the report design. The definition of an enum appears in the Project window when the Types filter includes type aliases. Enums appear as Typedef components in the hierarchy, as shown in Figure 20-18.

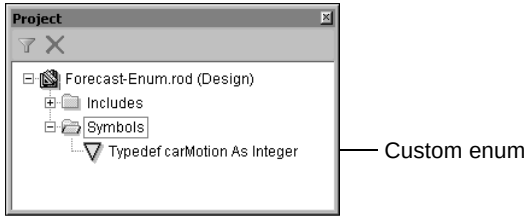


Figure 20-18 An enum as a Typedef component in Project

You can use custom enums in the same way as any built-in enum type. For example, you can use a custom enum as the data type for a custom variable or property on a report component. You can use a custom enum in the code of overridden methods.

How to create an enum in e.Report Designer Professional

1 With a report design open, choose Tools➛Types. Types appears, as shown in Figure 20-19. Using this dialog box, you can create and delete enums and structures.

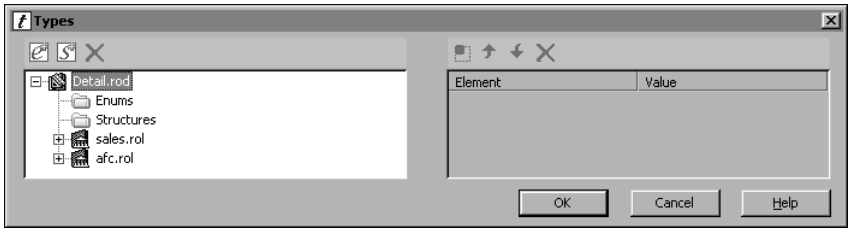


Figure 20-19 The Types dialog box

2 Choose Add Enum. An enum icon and field appear beneath the Enums folder, as shown in Figure 20-20.

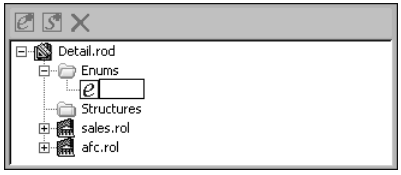


Figure 20-20 The enum icon and field

3 In the enum field, type a name for the enum. Click the icon beside the enum field.

The pane at the right of the dialog box becomes active, as shown in Figure 20-21. In this pane, you add constants to the enum and assign types to them.

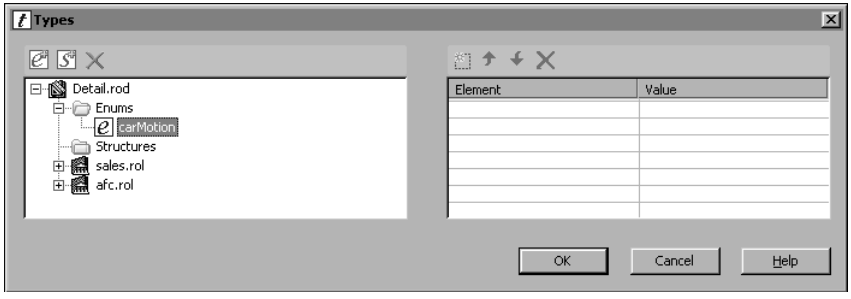


Figure 20-21 Use the pane at the right to create enum constants

4 Choose New. The first Element field becomes active.

5 Type a name for the first constant. Use Actuate Basic conventions for a name.

6 Place the cursor in the first Value field. Type an integer value for the constant.

7 When you finish adding constants, choose OK.

Chapter

21

Publishing reusable components in a library

This chapter contains the following topics:

- Understanding component and function libraries
- Planning a component library
- Creating a component library
- Creating a function library
- Including a component or function library in a report design
- Placing a library component in a report design
- Publishing a component to a library
- Providing access to a library using a configuration file
- Protecting a component or function library from modifications

Understanding component and function libraries

A library contains report components or an Actuate Basic function for reuse in multiple report designs. A library developer creates a library, then places it in a shared location for others to use. The shared location can be in a client file system or on a web server. Actuate e.Report Designer Professional supports working with two types of libraries, component libraries and function libraries.

About component libraries

A component library is a report object library (.rol) file that groups reusable report components. The components can be connections, data streams, images, frames, sections, page lists, or any other component a report object design (.rod) file can use. Figure 21-1 shows the contents of a component library.

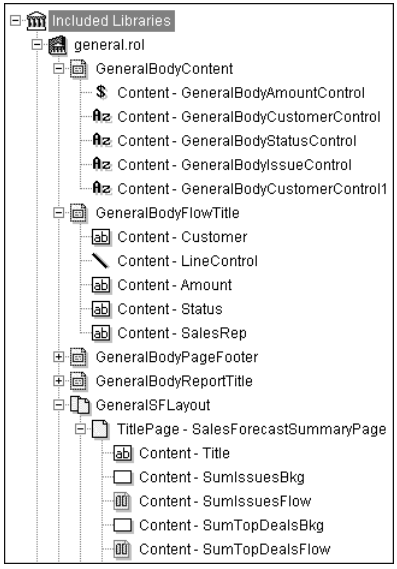


Figure 21-1 Component library

To use a component library, a report developer includes the library in a report design, then selects components from it for a report.

About function libraries

A function library is a Basic source (.bas) file that defines a function, such as a type declaration or a calculation. A function that a BAS contains is visible in Expression Builder as an external function, as shown in Figure 21-2.

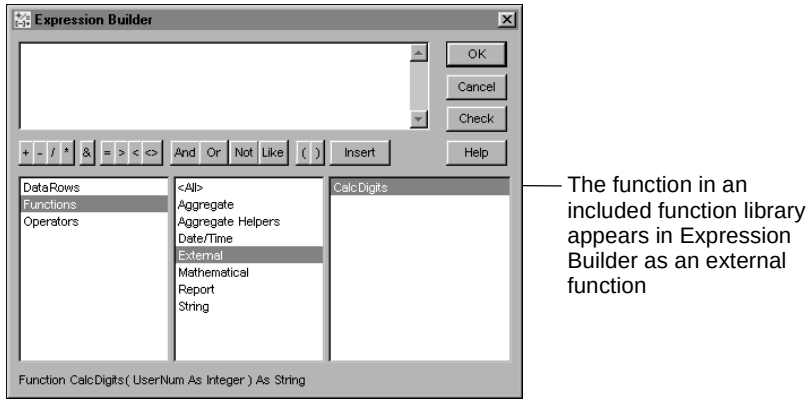


Figure 21-2 Function library

To use a function library, a report developer must include it in the report design.

About included and available libraries

To use a library, you include it in a report design. You can include a library only if the library is available to the design. An available library is one that e.Report Designer Professional can detect, using either the configuration file to which the design refers, the global search path, or the design search path. An available library does not appear in the design perspective until you include it in the report design.

When you include a library, you place the library in the design perspective so that you can use its contents. An included component library appears in Libraries in the design perspective. An included function library appears in the Layout window. Figure 21-3 shows each type of library in the design perspective.

You can include a library in a report design using either a configuration file or Library Organizer in e.Report Designer Professional. This chapter discusses both techniques for including libraries.

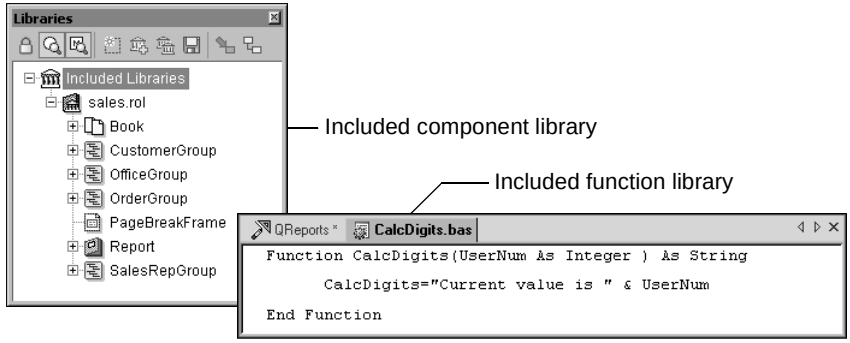


Figure 21-3 Included libraries in the design perspective

About Library Organizer

A report developer uses Library Organizer to work with component and function libraries. Library Organizer supports the following tasks:

- Viewing the libraries included in or available to a report design
- Including an available library in a report design
- Creating a new library and including it in a design
- Navigating to a library and including it in a report design

To view Library Organizer, choose Tools➛Library Organizer with a report design open. The Library Organizer appears, as shown in Figure 21-4.

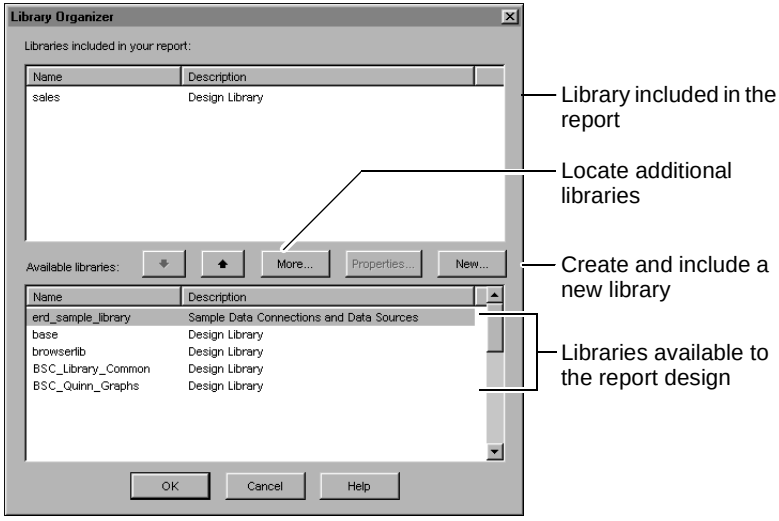


Figure 21-4 Library Organizer

The upper pane, Libraries included in your report, displays the libraries the report design includes. The lower pane, Available libraries, displays additional libraries available to the report. To use an available library, you must include it in the report design.

Planning a component library

When you develop a component library, keep the following concepts in mind:

- Understand the report developer's reporting task and provide only the components the report developer needs to complete the task. Avoid the practice of creating a large library that contains only frames or only images, for example, because few reporting tasks require only those components.

- Optimize report performance by keeping the size of the library small.
- Remember that library components are reusable and therefore must be stable. Place components in a library only when they are unlikely to change frequently.

Tailoring the library to the task

Ideally, a component library contains the basic components a report developer needs to accomplish a specific reporting task. For example, for a human resources department, a library developer creates a library that contains the following components of a standard report:

- Data streams and connections to two data sources, one listing personnel information, the other listing health insurance providers and premium information
- A report section that contains frames and controls to display data for a quarterly report about premium costs
- A page layout that contains flows, a corporate image, a page header, and a page footer

Using this library, the quarterly report has a standard appearance and always gets data from the same sources. Components such as the corporate image, page list, and data streams stay the same from quarter to quarter. When a component must change, the library developer can change it in the library. Any report design that refers to the component reflects the change.

Optimizing report performance

Including a large library in a design increases the size of the associated executable file and therefore affects report generation performance. For example, creating a library that contains dozens of frames and another library that contains numerous images forces the developer to include two large libraries in a report design.

Creating a component library

You can create a component library during the process of designing a report or you can create a library without reference to a report design. As you design a report, you can use Library Organizer to create a new library and include it in the report design. Then, in the design environment, you can add components to the library file.

To create a component library that is not associated with a report design, e.Report Designer Professional provides a design perspective for creating libraries. The following procedures explain both techniques for creating a component library.

How to create a component library without reference to a report design

1 In e.Report Designer Professional, choose File➛New. Create New Report appears.

2 Select Component Library.

Choose OK. The design perspective for a library appears, as shown in Figure 21-5.

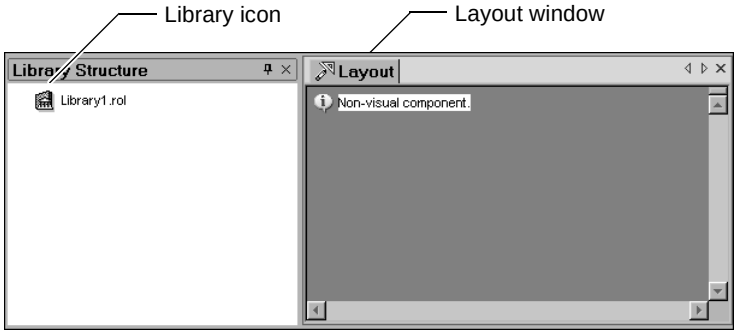


Figure 21-5 Component library in the design perspective

This perspective includes Library Structure and a layout window. Library Structure displays the hierarchy of library components when the library is complete. A library icon appears in Library Structure. In Layout, you can work with visual components. For example, you can move, resize, and align controls in the layout window.

3 Choose File➛Save As. Save As appears.

4 In Save As:

1 In Save in, select the folder in which to save the library.

2 In File name, type a name for the library.

3 Choose Save. The new name appears in Library Structure.

5 Drag the components that you want from Toolbox and drop them on the library icon.

You also can drop components within other components to create a hierarchy. For example, you can drop a report section on the library icon, then drop a series of frames and controls in the report section. As you add components to the library, you can rename them for ease of use.

At this point, you can either place the library in a shared location for others to use or you can use it in your own report designs.

How to create a component library and include it in a report design

1 With a report design open, choose Tools➛Library Organizer. Library Organizer appears, as shown in Figure 21-6.

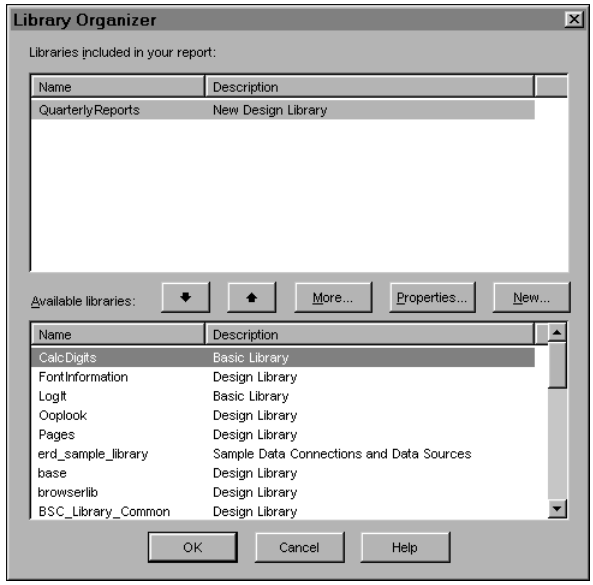


Figure 21-6 Including a new component library

2 Choose New.

3 In New Library:

1. In Save in, select the folder in which to save the component library.
2. In File name, type a name for the library.
3. In Save as type, select the file type Library (*.rol).
4. Choose Save. Library Organizer appears. The new library name appears in the upper pane, as shown in Figure 21-7.

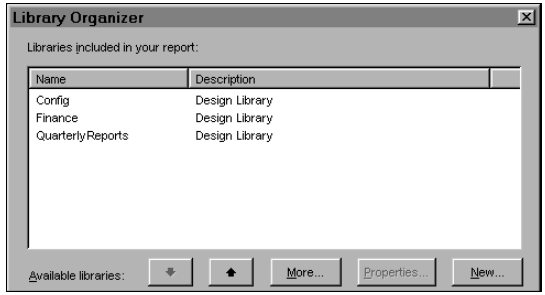


Figure 21-7 Viewing the new component library

4 In Library Organizer, choose OK. An icon that represents the library file appears in Libraries, as shown in Figure 21-8. The absence of + beside the icon indicates that this library does not contain components.

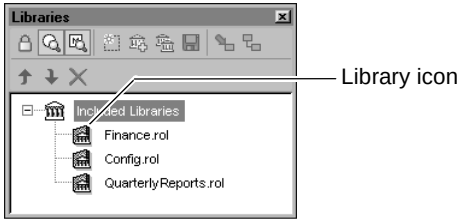


Figure 21-8 Icons for included libraries

5 Drag components from a component palette and drop them on the library's icon.

You can drop components within other components to create a hierarchy. For example, you can drop a report section on the library icon, then drop a series of frames on the report section icon, and add controls to each frame. As you add components to the library, you can rename them for ease of use.

Creating a function library

A function library is a Basic source (.bas) file that contains a reusable expression, such as a type declaration or a calculation. You create a function library using a text editor, then place the file in a shared directory for use by other library developers. You also can create a function library using Library Organizer. This topic explains how to use Library Organizer to create a function library. For more information about creating a BAS using a text editor, see *Programming with Actuate Basic.*

An expression that you place in a function library should be sufficient to perform a discrete task and succinct enough to be applicable in more than one context. The procedure in this section shows how to create a function library that converts an integer to a string value and concatenates the value to a text string.

How to create a function library and include it in a report design

1 With a report design open, choose Tools➤Library Organizer. Library Organizer appears.

2 Choose New.

3 In New Library:

1 In Save in, navigate to the directory in which to save the library.

2 In File name, type a name for the file.

3 In Save as type, select Source File (*.bas).

4 Choose Save. The new file name appears in the upper pane of Library Organizer, as shown in Figure 21-9.

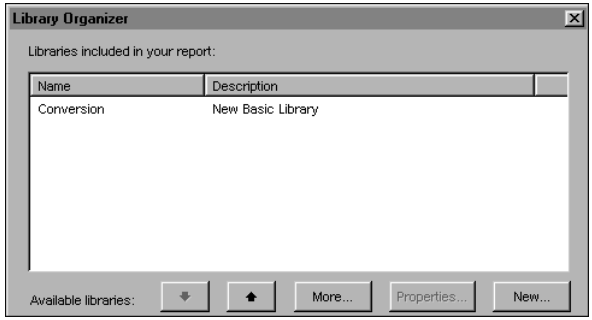


Figure 21-9 The new function library

4 Choose OK. An icon for the library appears in Libraries. A blank text editor page appears in the layout window, as shown in Figure 21-10.

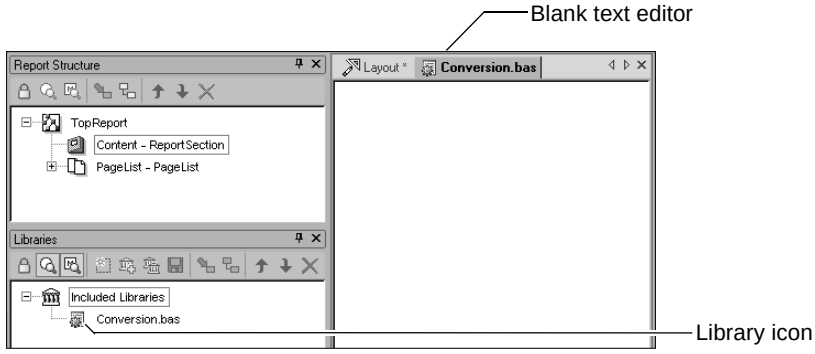


Figure 21-10 Text editor page for a new function library

5 Use the text editor page to write the function. For more information about creating a function, see *Programming with Actuate Basic.*

Including a component or function library in a report design

To use a component or function library, the report developer must first include the library in a report object design (.rod) file. You include a component or function library using Library Organizer or a configuration file. This section discusses using Library Organizer.

Using Library Organizer, a report developer includes a library in either of the following ways:

- Select an available component or function library from the list in the lower pane.
- Navigate to a component or function library in a local file system or on a web server.

You can also use e.Report Designer Professional to specify default libraries to include in every report design.

How to include an available library in a report design

1. With a report design open, choose Tools➛Library Organizer. Library Organizer appears.
2. In Available Libraries, select a component or function library. Choose the up arrow. The library appears in the upper pane, as shown in Figure 21-11.

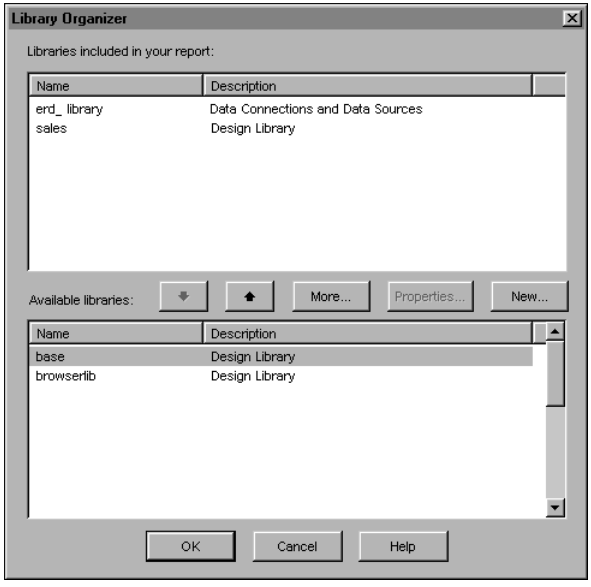


Figure 21-11 Selecting an available library

3. Choose OK. Library Organizer closes. The library that you included appears in the design perspective.

   If you include a component library, you can place the library's components in the design. If you include a function library, the function appears in the layout window.

How to include a library from a local file system

If a component or function library is not available to a report, you can use Library Organizer to navigate to the library in a local file system.

1 With a report design open, choose Tools➛Library Organizer. Library Organizer appears.

2 Choose More. Include Library appears, as shown in Figure 21-12.

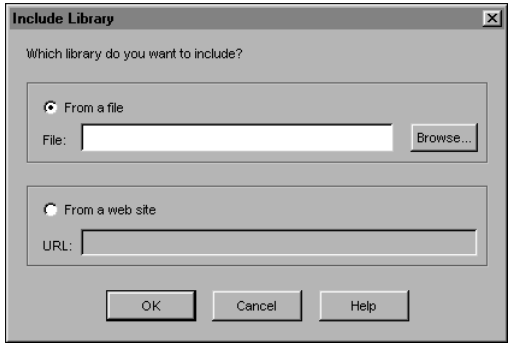


Figure 21-12 Include Library window

3 Select From a File.

In the File field, you can type a file name or relative path to navigate to a library. You also can choose Browse. If you choose Browse, Include Module appears, as shown in Figure 21-13. Files of type displays the types of file that you can include.

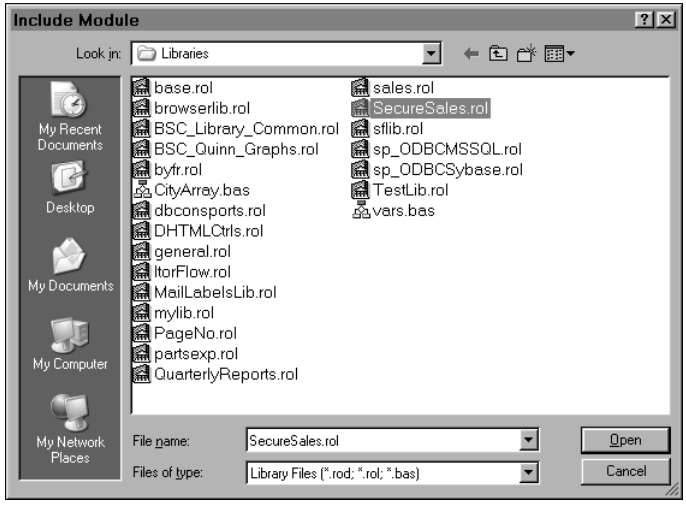


Figure 21-13 Include Module window

4 Select the library to include. Choose Open. Include Library appears, displaying a path to the file, as shown in Figure 21-14.

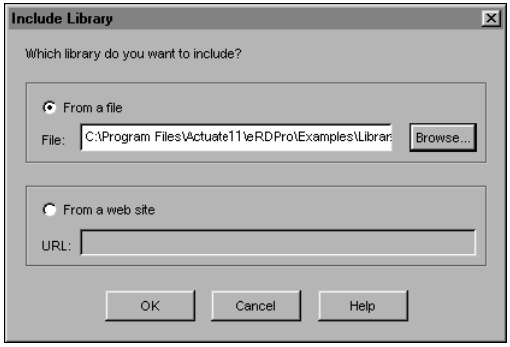


Figure 21-14 Library path displayed

5 Choose OK. Library Organizer appears. The library is included in the report design.

6 Choose OK to return to the report design. If you include a component library, the library appears in Libraries. If you include a function library, the Basic source (.bas) file appears in the layout window.

How to include a library from a web server

If a component or function library resides on a web server and is not available to your report, you can navigate to the library to include it. The library must be in the root directory of the web server or in a subdirectory of the root directory.

1 With a report design open, choose Tools➛Library Organizer. Library Organizer appears.

2 Choose More. Include Library appears.

3 Select From a web site. Then, type a URL similar to the following example:

```
http://localhost:8900/Libraries/ForecastDetails.rol
```

where

- localhost:8900 is the name and port of the web server.
- Libraries is the directory that contains the library. This directory is in the web server's document root directory.
- ForecastDetails.rol is the name of the library file.

Choose OK. The library name appears in the upper pane of Library Organizer.

4 In Library Organizer, choose OK. The report design appears.

If the library that you include is a component library, an icon for the library appears in Libraries. If the library is a function library, the function appears in a text editor page in the layout window.

How to add a default component library for report designs

The e.Report Designer Professional configuration file specifies default libraries to include in all your report designs. You can include additional default libraries if you know the libraries contain stable components and you are likely to use those components in many of your reports.

1. With a report design open, choose Tools➤Options. Options—Design Editor appears.
2. Choose Libraries. Options—Libraries appears, as shown in Figure 21-15.

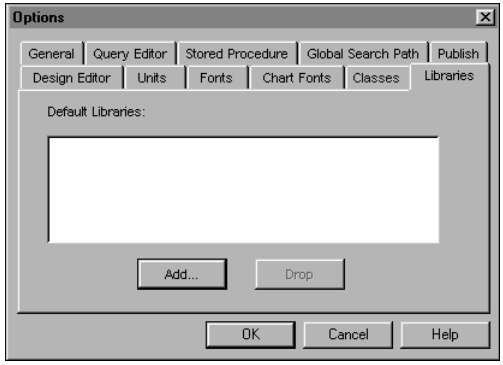


Figure 21-15 Options—Libraries

3. Choose Add. Locate Default Library appears, as shown in Figure 21-16.

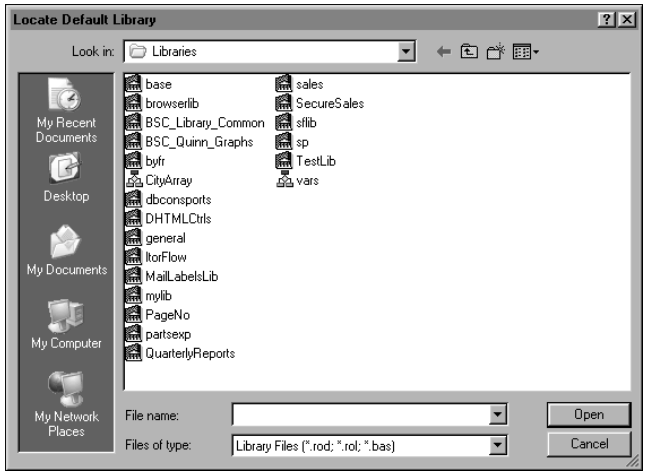


Figure 21-16 Locate Default Library window

4 Navigate to and select the file that you want to add. Choose Open. The library path appears in Options—Libraries.

5 Choose OK. The library that you selected is included in every new report design you develop.

Placing a library component in a report design

After you include a component library in a report design, you can place components from the library in the design. When you place a library component in a design, e.Report Designer Professional creates a reference, or link, to the library component. If you then modify the reference, e.Report Designer Professional creates a local subclass of the component.

Creating a component reference

When you drag a component from a library and drop it in a report design, you create a reference to the component. Referring to a component means creating a link between the component and the design. If another user changes the component in the library, your reference inherits the change. Use a reference when an existing component meets your needs, you want to use the component in its current form, and the report design can accept future changes to the component.

In Report Structure, the icon for a component reference contains an arrow on top of the standard icon for the component type, as shown in Figure 21-17.

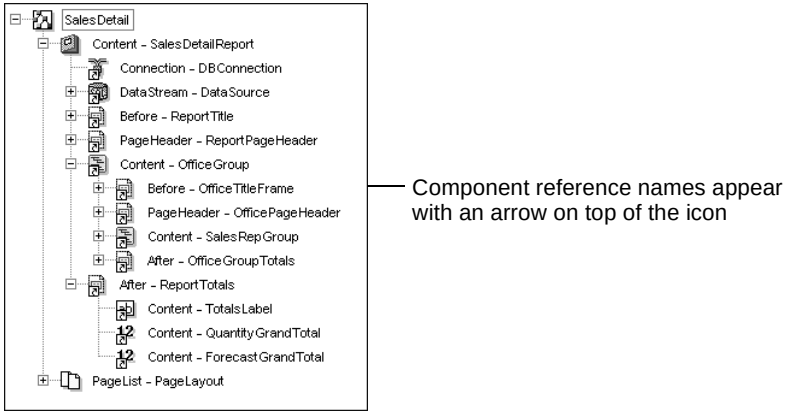


Figure 21-17 Component reference name in Report Structure

A component reference has the same class name and superclass as the original library component. The reference class name and superclass are read-only fields in Properties, as shown in Figure 21-18.

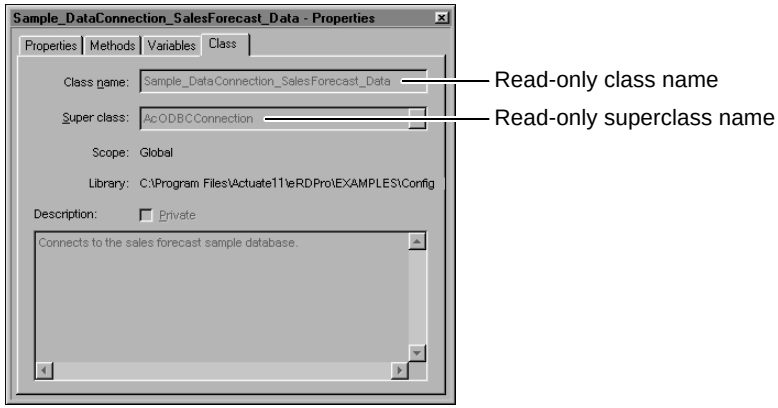


Figure 21-18 Read-only fields

Creating a subclass of a library component

When a component library is available to multiple users, changing the library or removing the library from circulation can introduce unexpected changes to any report design that uses the library. It is possible for a developer to open a report design and find a Report Structure that looks like the one in Figure 21-19.

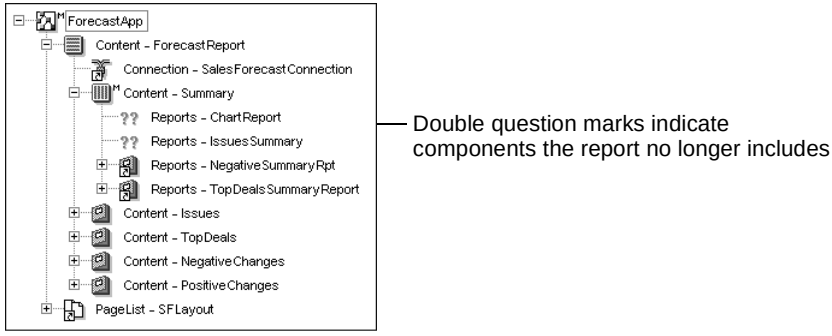


Figure 21-19 Missing components

The double question marks beside a component name indicate that the component no longer exists in the report. Someone either deleted, renamed, or moved the component library that contains these components. To use the report design, you must determine what happened to the library. If someone renames or moves a library, you can restore it by including it again using its new name or path. If someone deletes a library, you must either reconstruct it or change your report design.

You can avoid having to restore a library by converting references to local subclasses. Converting to subclasses removes the relationship between the component and the library without removing the component from the library.

To convert a referenced component to a local subclass, use one of the following techniques:

- Right-click the component, then choose Local Subclass.
- Modify the component in the report design. For example, if you change a property value, the referenced component becomes a local subclass.

When you change a reference to a local subclass, the icon for the component name in Report Structure no longer has an arrow on it. The contents of the frame remain references until you create subclasses for them.

When you create a subclass of a library component, you change both its class name and its superclass. The new class name reflects the name of the component in the report. The new superclass name reflects the component's relation to the report structure. For example, Figure 21-20 shows the superclass of a modified label control, TotalsLabel, in a frame, ReportTotals.

The superclass name refers to the frame that contains the control and to the control itself.

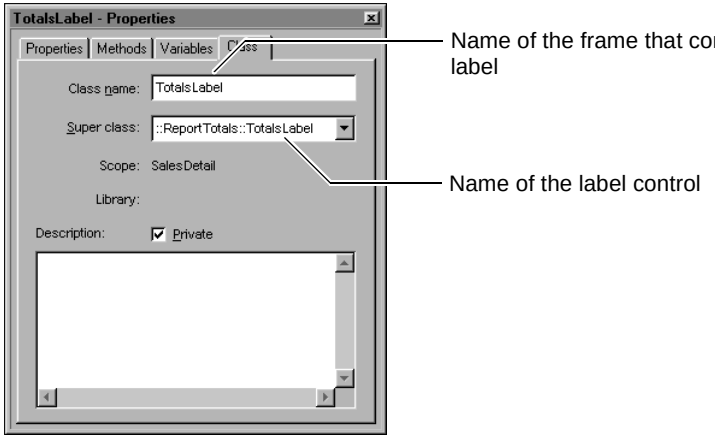


Figure 21-20 Superclass name components

Publishing a component to a library

When you publish a component to a library, you can make the library available to other report developers or reserve it for your own reports. You can publish a report component that you create in a report design. You also can publish a component that is a subclass of a library component. You cannot publish a reference.

You can publish a component either to an existing library that is included in your report design or to a new library you create.

After you publish an original or subclassed component to a library, the design contains a reference to the component. If you change the component in the library after you publish it, the component changes in the design and in any other file that refers to the library component.

When you publish a data stream in a library, publish the connection component with the data stream. Grouping the connection and the data stream makes it easier for others to reuse or modify the component.

How to publish a report component to a library

1. Open the report object design (.rod) file or report object library (.rol) file that contains the component you want to publish.
2. In Libraries, verify that the library to which you want to publish is included in the report design. If the library is not included, you must include it.
3. Drag the component from Report Structure and drop it where you want to place it in Libraries, as shown in Figure 21-21. You can drop it on the library icon or you can drop it in another component, such as a frame or a report section.

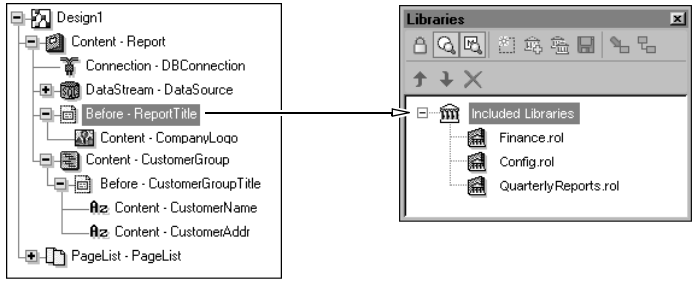


Figure 21-21 Publishing a report component to a library

Component Drop appears.

4. Select Publish the component, as shown in Figure 21-22.

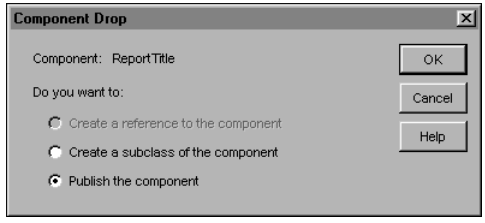


Figure 21-22 Component Drop window

Choose OK. The component and any components it contains appear in the library. A reference to the component appears in the report design, as shown in Figure 21-23.

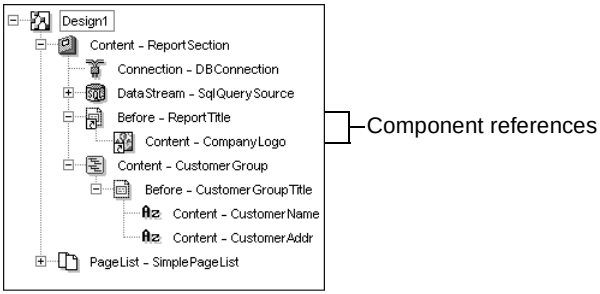


Figure 21-23 Component references

Providing access to a library using a configuration file

A configuration file is an XML document that provides paths to component and function libraries, data streams, report templates, and other configuration files for use in a report design. The Design element of a configuration file specifies which library (.rol) files to include in every report design that refers to the configuration file and which libraries are available but not included. It is not necessary to use a configuration file to place a library in a report design but doing so provides enterprise-wide access to structural, data, and visual components for report building. The configuration file also ensures centralized control of the structure and format of report components.

Configuration files are especially useful when you want to include a different set of components for different departments or projects. For example, you can specify that all financial report designs use one library and all customer service report designs use another library. By specifying the appropriate configuration file for each report design, you provide access to required components without explicitly including each library.

Listing 21-1 shows a configuration file in its entirety. The file can contain any or all of the elements shown here, in any order. For the purposes of designating libraries, you use the Library element in the Design element.

Listing 21-1 Example of a complete configuration file

```
<Config>
   <SearchPath>
      <Location>\DesignResources</Location>
   </SearchPath>

   <Include>
      C:\Market\StocksConfig.xml
   </Include>
```

```
   <Design>
      <Library
         Autoinclude="true"
         Alias="Financial Reports">
            C:\LibraryFiles\Finance.rol
      </Library>

      <Connection Type="Trades"
         Alias="Financial reporting resources"
         DefinedIn="C:\LibraryFiles\Stocks.rol"
         Description="For quarterly and weekly financial
            reports."
         IsDefault="True">
         <Property
            PropName="DataSource">
            ChartExamples
         </Property>
       </Connection>

      <Template
         Alias="Finance Reports Template"
         Description="Template for Finance Department use.">
            C:\Market\FinanceTemplate.rod
      </Template>
   </Design>

   <Runtime>
      <ConnectOptions
         Type="ODBCConnection">
         <Property
         PropName="DataSource">
            sfdata
         </Property>
      </ConnectOptions>
   </Runtime>
</Config>
```

This section describes how to use the elements of the configuration file that make libraries available to a report design or included in the report design. For general information about selecting a configuration file for a report design or creating a configuration file, see *Accessing Data using e.Report Designer Professional.*

Including or making available a single library

Using the Library element within the configuration file's Design element, you can include one or more libraries in a report design that uses the configuration file. You also can specify that a library be available to the design. Create a separate Library element for each library you want to include or make available.

The following Library element includes the Finance.rol library into report designs using this configuration file. It specifies the path to the library file and provides an alias for the library.

```
<Design>
   <Library
      Autoinclude="true"
      Alias="Financial Reports">
         C:\LibraryFiles\Finance.rol
   </Library>
</Design>
```

The Library element defines a path to the library. You use the attributes of this element, shown in Table 21-1, to provide additional information about the library.

Table 21-1 Library element attributes

Attribute	Description
Alias	Optional element that provides text for the Description column in Library Organizer. If you do not set Alias, Library Organizer displays a default description for the library. The default description for a component library is Design Library. The default description for a function library is Basic Library.
Autoinclude	Specifies whether to include a library in a new report design. If you set Autoinclude to true, the library appears in the Libraries page of any new report design that uses this configuration file. If you set Autoinclude to false or omit it, the library is available for use in report designs and appears in the lower pane of Library Organizer.

Making all libraries in a directory available

You can use the configuration file's SearchPath element to provide the location of function and component libraries that you want to make available to report designs. Using Library Organizer, a report developer can view the libraries that are made available through SearchPath and choose to include one or more of the libraries in the report design. The lower pane of Library Organizer displays the libraries listed in the SearchPath element, as shown in Figure 21-24.

To specify a search path, provide an absolute path or a path relative to the directory that contains the configuration file. You also can specify an absolute or relative URL. Place each path within a separate Location element, as shown in the following example:

```
<SearchPath>
   <Location>\Libraries</Location>
   <Location>C:\Includes</Location>
</SearchPath>
```

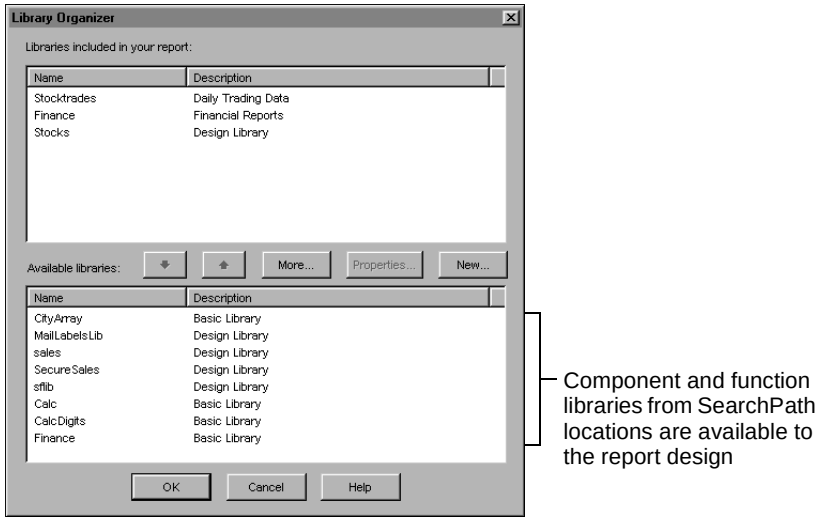


Figure 21-24 Making libraries available through SearchPath

Protecting a component or function library from modifications

In an environment where developers share and reuse libraries, you can make a library read-only to prevent unintentional modifications. Other developers can include a read-only library in their designs. They can create references and subclasses using the library but they cannot change the library. You can make a library read-only using the capabilities of your operating system. For example, in Windows Explorer, you can navigate to a library file. Right-click the file and choose Properties to view the file properties. Then, select Read-only, as shown in Figure 21-25.

When you select a component of a read-only library, the words Read Only appear in the title bar of Properties, as shown in Figure 21-26.

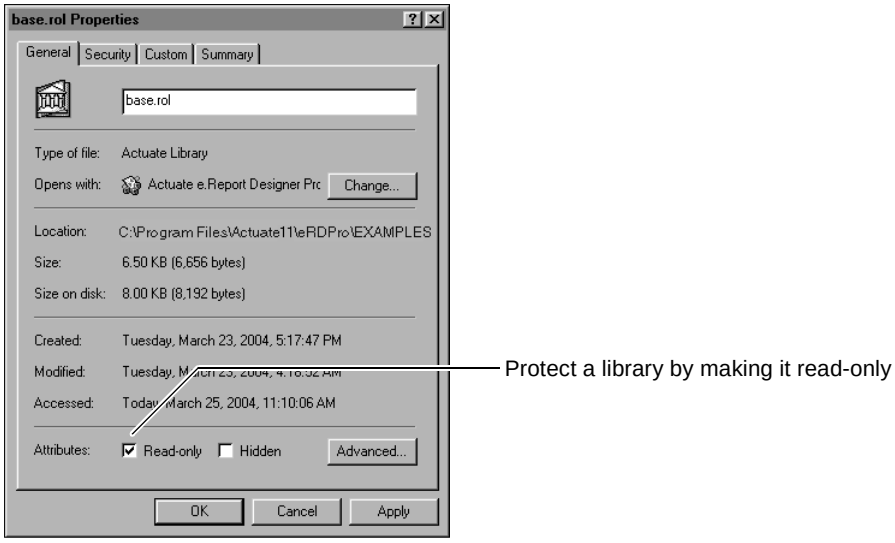


Figure 21-25 Applying the read-only attribute to a library

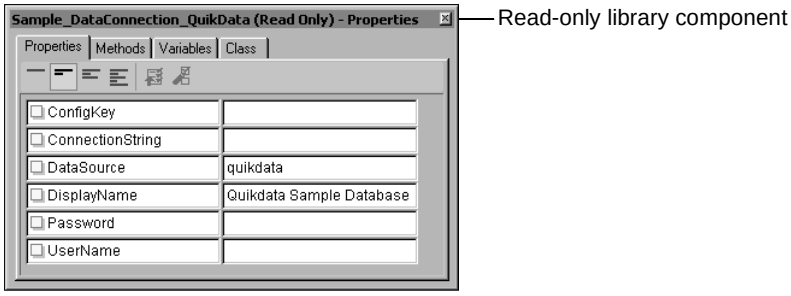


Figure 21-26 A read-only component

Chapter

22

Publishing a report

This chapter contains the following topics:

- About publishing a report
- Creating an Actuate iServer profile
- Publishing a report to an Encyclopedia volume

About publishing a report

When you publish a report to an Actuate iServer Encyclopedia volume, you make that report available to other users. To publish a report, you use a web browser, a web server, an Encyclopedia volume, and Actuate Management Console. To locate and view a report, a report user uses Actuate Information Console.

An Encyclopedia volume is a repository for report items, such as executable files, report instances, and parameter values files that you create using e.Report Designer Professional. In order to publish a file, you must have a user account for the Encyclopedia volume and an iServer profile. This chapter explains how to create an iServer profile and publish an item to an Encyclopedia volume.

After you publish an item, you can use Actuate Management Console to manage and deliver the item. For example, you can control user access to a report, run a report, schedule a job, distribute a report over the web, archive an Encyclopedia volume item, and manage version control. For more information about Actuate iServer and Encyclopedia volumes, see *Managing an Encyclopedia Volume.*

Actuate Information Console supports accessing and working with reports using a web browser. A report that you publish to an Encyclopedia volume is visible to a user in Actuate Information Console, based on the user's access privileges. For more information about Actuate Information Console, see *Using Information Console.*

Creating an Actuate iServer profile

To publish an executable file to an Encyclopedia volume, you must have at least one iServer profile. The profile indicates the Actuate iServer, Encyclopedia volume, and destination folder to which you can publish a file. When the file that you publish has the same name as an existing file in the Encyclopedia volume, the profile specifies whether to replace the existing file or create a new version of it. The profile also indicates whether to copy autoarchive rules, privileges, and descriptions from the latest version of a file to a new file that has the same name. Each profile can publish to a different destination and can use different publication settings.

Table 22-1 describes the information that you need to create an iServer profile.

Table 22-1 Actuate iServer profile information

Profile item	Description
Profile name	Descriptive name for the iServer profile.

Table 22-1 Actuate iServer profile information

Profile item	Description
Application server and Port	Machine name and port number for the application server. The application server is the machine that hosts the Actuate Management Console executable file. The default port is 8900. Actuate HTTP service listens on this port.
Context path	Part of a URL that specifies the volume administration path. Use /acadmin for Actuate Management Console.
iServer and Port	Machine name and port number that e.Report Designer Professional uses to contact Actuate iServer. The default iServer port number is 8000.
Volume name	Name of the Encyclopedia volume to which you want to publish files using this profile.
User name	User name with which to log in to the Encyclopedia volume.
Password	Password with which to log in to the Encyclopedia volume.
Remember Password	If you select Remember Password, e.Report Designer Professional stores the password in encrypted format in acserverprofile.xml. You are not required to provide the password again when you next launch e.Report Designer Professional. Storing the password in this way may pose a security risk. If you deselect Remember Password, e.Report Designer Professional stores the password in memory for the duration of the session. When you next launch e.Report Designer Professional, you must provide the password again in order to connect to the iServer. Remember Password is selected by default.
Destination folder	Folder into which you want to upload the file, such as: ■ ~ for the current user's home folder ■ /Sales/Forecasts for the Forecasts folder in the Sales folder in the Encyclopedia volume root folder

How to create and test an Actuate iServer profile

1. With a report design open in the design perspective, choose File➛Publish to Server. Publish and Preview Options appears.
2. Choose Add.

3 In iServer Profile, provide profile details, as shown in Figure 22-1. In User name, type your user name in place of MyUser.

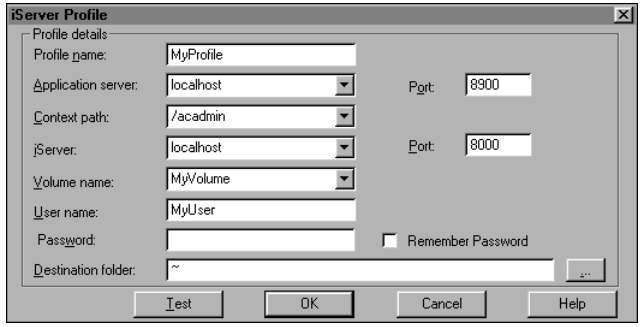


Figure 22-1 Setting up profile information

4 Choose Test. e.Report Designer Professional tests the Actuate iServer connection using the values that you provide. If the connection succeeds, a confirmation message appears. If the connection fails, an error message appears. Choosing Test does not verify the presence of the destination folder.

5 Choose OK to close the confirmation message.

6 Choose OK to close iServer Profile. Publish and Preview Options appears. Figure 22-2 describes the settings on Publish and Preview Options.

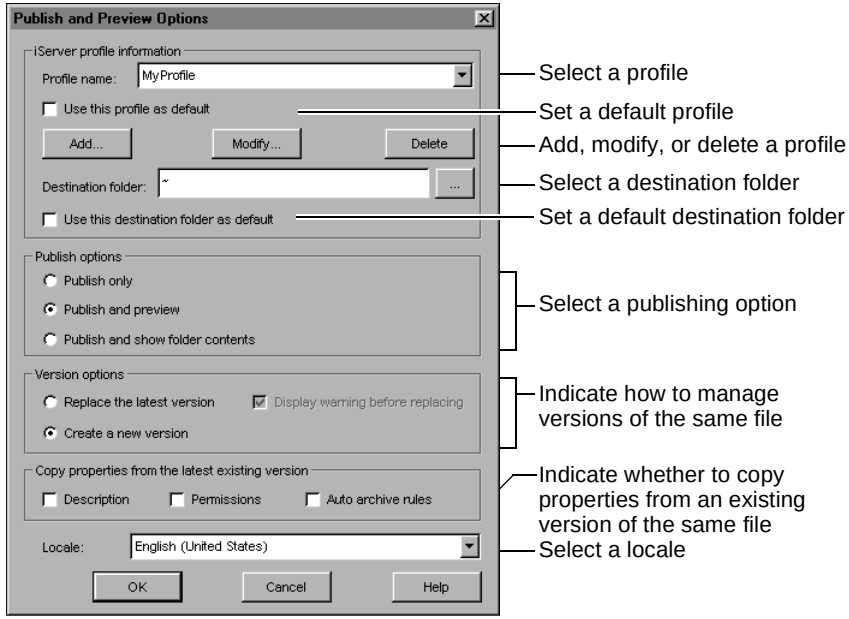


Figure 22-2 Publish and Preview Options

Setting publication options

You can set publication options for an iServer profile using Publish and Preview Options. You can also change the settings for an existing profile or delete a profile.

Table 22-2 describes the publication options that you can set in the Publish options section of Publish and Preview Options.

Table 22-2 Publication options

Option	Description
Publish only	This option uploads the file to the Encyclopedia volume for the profile and returns a confirmation message.
Publish and preview	This option uploads the file to the destination folder for the profile, runs the file, and displays the output in DHTML.
Publish and show folder contents	This option uploads the file to the destination folder for the profile, opens the folder, and displays the file in a list of folder contents.

Managing file versions

When a file that you publish has the same name as an existing file in the destination folder in the Encyclopedia volume, you can use Version options to replace the existing file or create a new version of it. If you replace the existing file, you can display a warning message before overwriting the file, as shown in Figure 22-3.

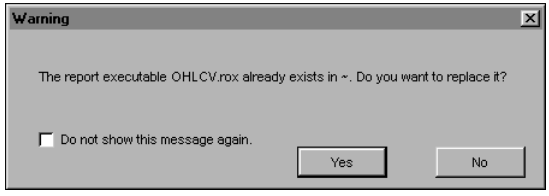


Figure 22-3 Warning message

Copying file properties

When a file that you publish has the same name as an existing file in the destination folder in the Encyclopedia volume, you can copy properties of the existing file to the file that you publish. The properties that you copy come from the latest version of the existing file. You can copy properties whether you replace

the existing file or create a new version of it. You can copy the following properties when you publish a file:

- Description
- Permissions
- Autoarchive rules

Publishing a report to an Encyclopedia volume

When you publish a report executable file, e.Report Designer Professional:

- Generates an executable file from the report design.
- Logs in to the Encyclopedia volume that your iServer profile uses.
- Uploads the executable file to the destination folder your profile specifies.
- Performs other tasks your publication settings require, such as opening your destination folder, displaying the output, or copying properties from an existing version of the file.
- Displays the report in DHTML format in the layout window.

  If the report that you publish uses parameters, the layout window first displays an Actuate Management Console page that supports providing parameter values, then running the report.

How to publish an executable file to an Encyclopedia volume

1. With a report design open in the design perspective, choose File➛Publish to Server. Publish and Preview Options appears.
2. In Publish and Preview Options:
   1. In iServer profile information:
      - Select a profile from the drop-down list.
      - Select a destination folder.
   2. In Publish options, select Publish and preview.
   3. In Version options, select Create a new version.
3. Choose OK. e.Report Designer Professional generates the report object executable (.rox) file.

   If the report that you publish uses parameters, the Parameters page of Actuate Management Console appears in DHTML Viewer, as shown in Figure 22-4.

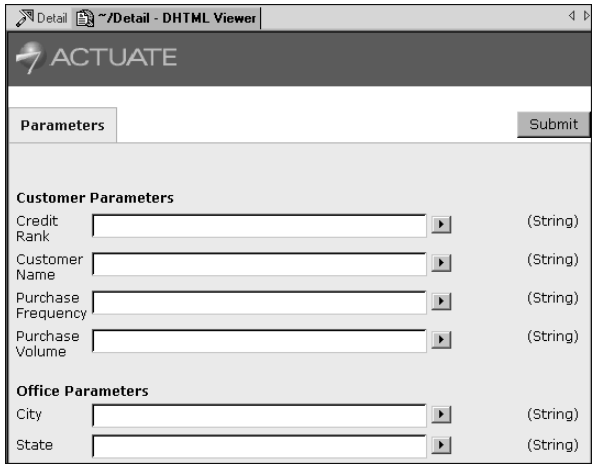


Figure 22-4 DHTML Viewer

Select the parameters to use and choose Submit. The report appears in DHTML Viewer as shown in Figure 22-5.

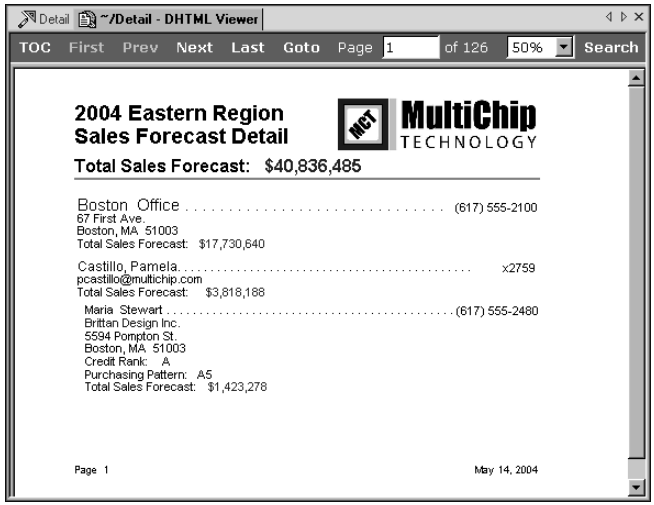


Figure 22-5 Published report in DHTML Viewer

If the report that you publish does not use parameters, the first page of the report appears in DHTML Viewer.

Chapter

23

Generating information objects from Actuate Basic report designs

This chapter covers the topic "Generating an information object from an Actuate Basic report design."

Generating an information object from an Actuate Basic report design

An Actuate Basic report can be used as a data source for an Actuate BIRT Studio report. BIRT Studio is a web-based report design tool. In BIRT Studio, information objects and report designs are stored on Actuate iServer. An information object is a named SQL query.

To generate an information object from an Actuate Basic report design, you must specify a default iServer profile. In addition, the connection component in the report design must be of a supported type. Supported connection types are Oracle, SQL Server, ODBC, or DB2.

In e.Report Designer Professional, you open a report design based on a supported connection type. Then, you export the query to Actuate Information Object Designer to create an information object and store it on Actuate iServer. Information Object Designer uses the query defined for the data source component in the report design to create the following files, using information from the report design:

- Information object project

  Information Object Designer creates a project using the default iServer profile defined in e.Report Designer Professional.

- Connection definition

  Information Object Designer creates a connection definition using the properties of the connection component in the report design.

- Map file

  A map (.sma) file is a representation of the data query. A BIRT Studio user uses the map as an information object data source.

  A map consists of data columns. The properties of a map column derive from the properties of the first control whose value expression is set to the corresponding database column. For example, if the CustID control's TextPlacement.Horizontal property is set to TextAlignRight, the CustID column's Horizontal Alignment property is set to right. If no such control exists, Information Object Designer uses the default values for map column properties.

How to generate an information object from an Actuate Basic report design

1. In e.Report Designer Professional, open an Actuate Basic report design that has a SQL query source, using either a graphical or a textual query.

   The query must use fully qualified table names. Figure 23-1 shows the Textual Query Editor.

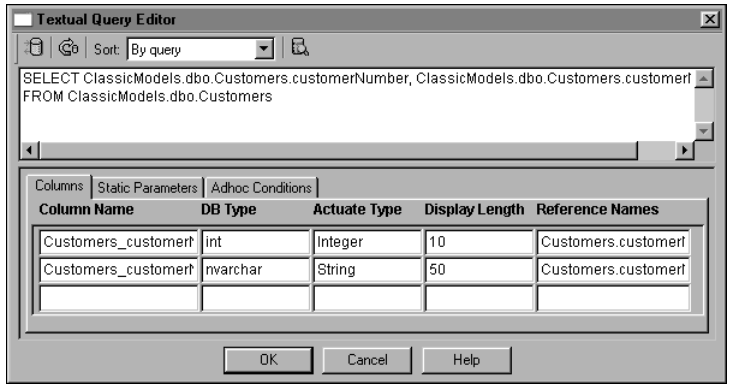


Figure 23-1 Textual Query Editor

2 In Report Structure, select the data source component, as shown in Figure 23-2.

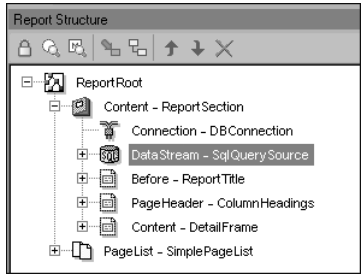


Figure 23-2 Report Structure pane with SqlQuerySource component selected

3 On the main toolbar, choose Export to Information Object.

4 In Select Server Profile, complete one of the following tasks to specify a default iServer profile:

- Select an iServer profile from the list, as shown in Figure 23-3.

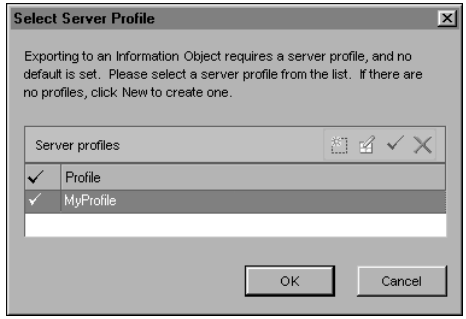


Figure 23-3 Selecting an iServer profile

- Choose New to create a new iServer profile. In iServer Profile, provide the iServer profile details then choose OK.
  Figure 23-4 shows the settings for an iServer profile called MyServerProfile.

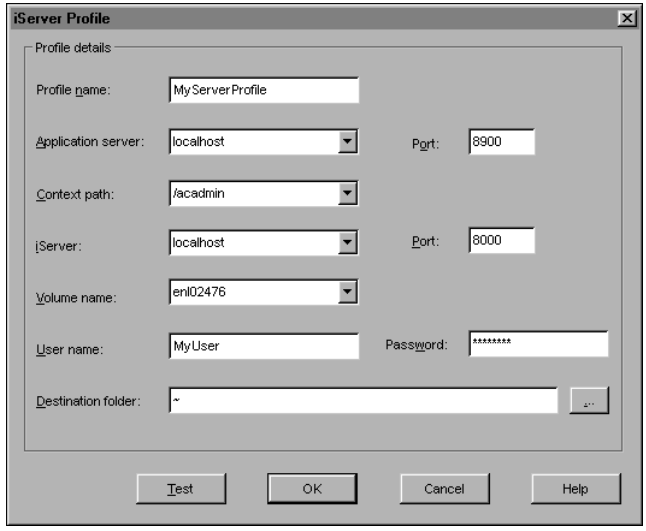


Figure 23-4 Settings for an iServer profile called MyServerProfile

In Select Server Profile, choose OK.

A message appears. Choose OK.

Information Object Designer opens and creates a project, a connection definition (.dcd) file, and a map (.sma) file, as shown in Figure 23-5. The project appears at the left under the Navigator tab. The map file containing the query appears next to it, and, to the right, a properties window displays information about the selected item.

The Servers tab shows that as you create DCD files, maps, and information objects, Information Object Designer saves them to the designated server, using the profile that you created earlier.

5 If the connection does not use the default port, modify the port number in Data source connection properties. In Projects, double-click the connection definition (.dcd) file to display Data source connection properties, as shown in Figure 23-6.

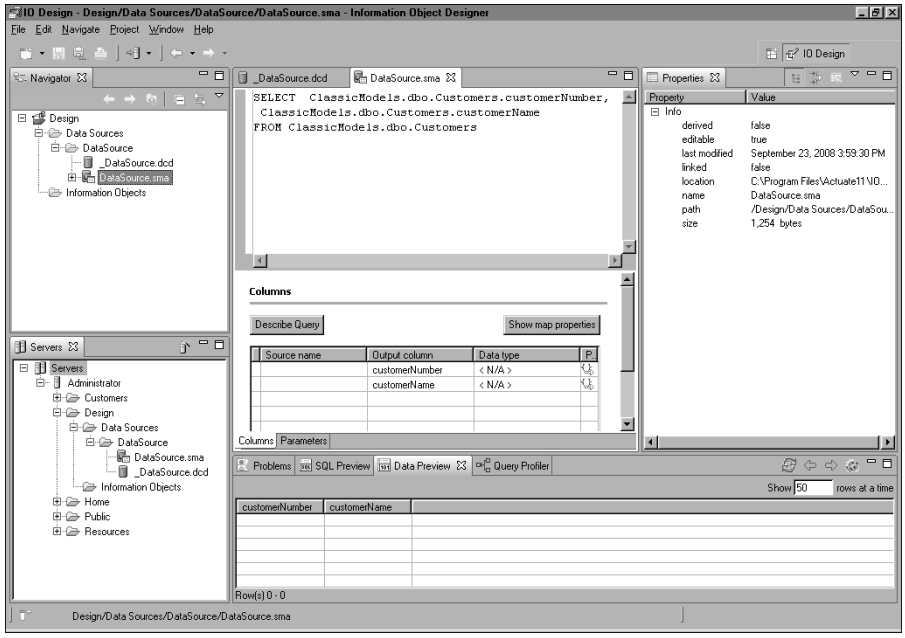


Figure 23-5 Information Object Designer

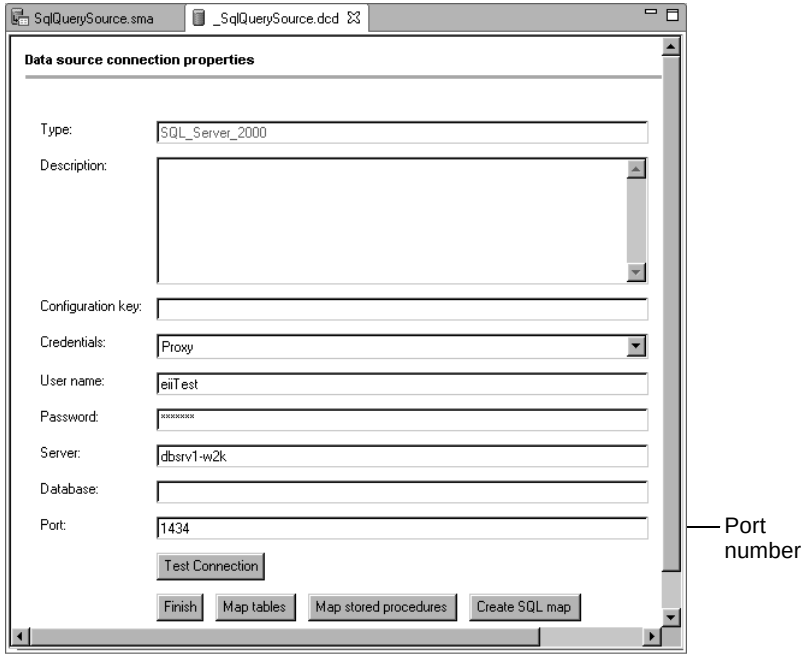


Figure 23-6 Data source connection properties in Information Object Designer

6 Choose the Data Preview tab.

7 In Data Preview, choose Refresh to display information object output, as shown in Figure 23-7. The first rows of data appear in Data Preview.

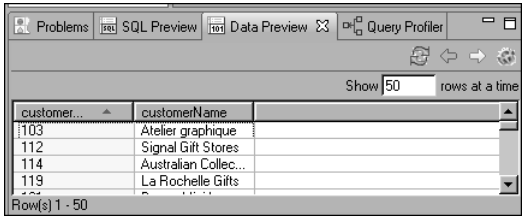


Figure 23-7 Data Preview view in Information Object Designer

8 Close Information Object Designer.

Chapter

24

Customizing the design environment

This chapter contains the following topics:

- About design environment customization
- Setting design environment options
- Setting a font in the designer
- Setting units of measurement
- Setting design search paths
- Setting general options in the designer
- Customizing the classes of the toolbox
- Changing the item list for Create New Report

About design environment customization

Actuate e.Report Designer Professional is a highly customizable interface. You can change display characteristics, such as default fonts, units of measurement, and tool placement. You can choose the configuration file that provides libraries and data source connections for a report design. You also can customize search paths, the component toolbox, and other aspects of the report design environment.

The topics in this chapter describe how to use the Options dialog, the design perspective, and registry key settings to customize the environment.

Setting design environment options

In e.Report Designer Professional, if you right-click a feature, a menu from which you can set options appears. For example, if you right-click the white space in the Report Structure window, available options appear in a context menu, as shown in Figure 24-1.

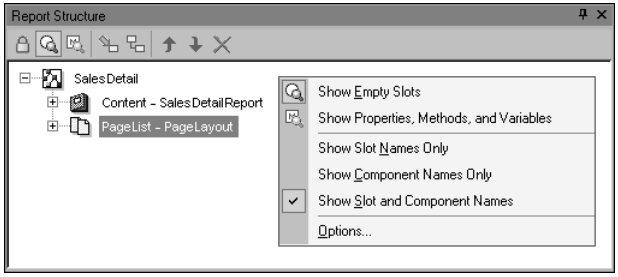


Figure 24-1 Context menu for setting options

When you choose Options from the context menu, Options—Design Editor appears. You can set options on this page or choose a tab to access more options.

Figure 24-2 shows the available options on Options—Design Editor.

In Options—Design Editor, you can customize the appearance of information and the design perspective appearance in the following ways:

- Show or hide group sections in Report Structure.
- Display sample data in controls, such as 1234 in an Integer control. In the design layout, the table.column name appears in the control if you deselect Display sample data.
- Display slot names, component names, or both in Report Structure. For example, if you select Both Slot and Component Names,

Connection—ODBCConnection appears in Report Structure for a connection slot that contains a component named ODBC Connection.

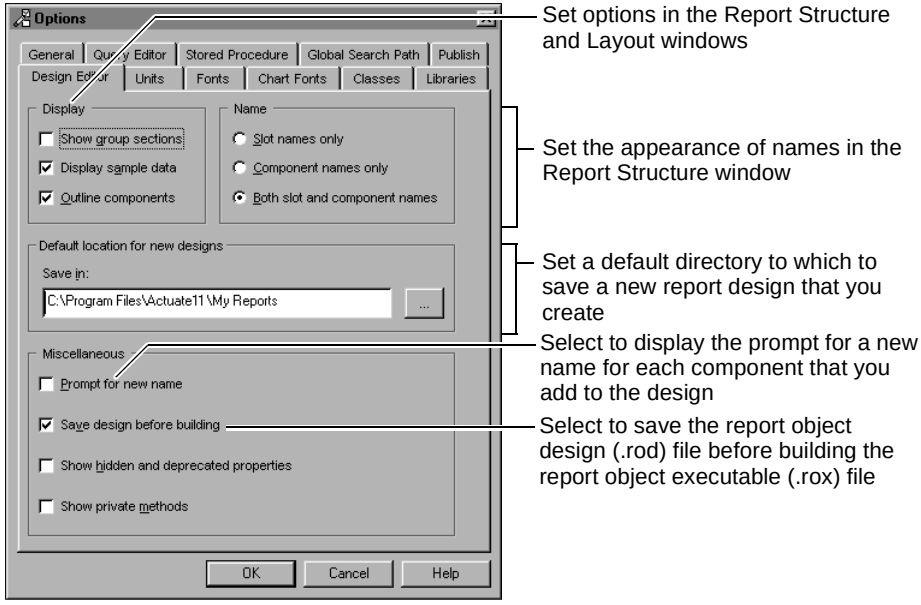


Figure 24-2 Options—Design Editor

You can also modify the following e.Report Designer Professional features:

- Specify a default directory to which to save a new report design that you create.
- Specify that a prompt appears for you to provide a component name for each component that you add to a design.
- Specify saving a report object design (.rod) file before e.Report Designer Professional builds the report object executable (.rox) file.

Setting a font in the designer

To customize fonts, e.Report Designer Professional supports the following options:

- Setting the default font for labels, data, and source editor
- Setting the default font for a chart
- Setting the font for Report Structure

Setting the default font for labels, data, and source editor

When you design a report, e.Report Designer Professional uses default fonts for label, data, and source editor text. You can modify a default font attribute, such as font style or size. You use Options—Fonts to modify default fonts. The default font for labels and data controls appears only for controls that are direct subclasses of the Actuate Foundation Classes. The default font does not appear for controls that are subclasses of a user-defined class.

Specifying a TrueType font embeds font width information in the report object executable (.rox) file. If you use a Type 1 font, a generated Adobe PDF file or a dynamic text control's HTML content can display inaccurate positioning for right-aligned and center-aligned text. The inaccurate positioning does not affect left-aligned text in generated PDF files or non-HTML text in dynamic text controls.

A font color can use different names in Options—Fonts, Properties, and the Format toolbar. For example, fully saturated green is called Green in Properties and Lime in Font. For more information about color, see AcColor in *Programming with Actuate Foundation Classes.*

If you change the source editor font and you have a source editor open, you must close and reopen the source editor for the font change to apply.

How to set the default font for labels, data, and source editor

1. Choose Tools➛Options.
2. In Options—Design Editor, choose Fonts. Options—Fonts appears, as shown in Figure 24-3.

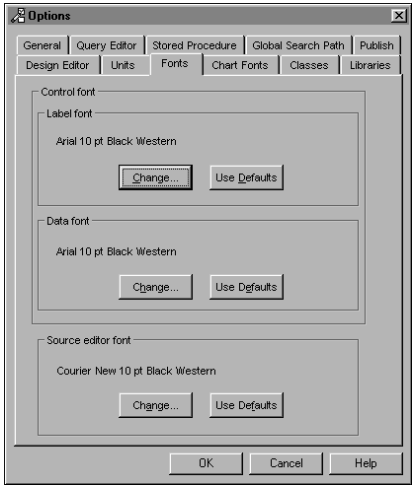


Figure 24-3 Setting default fonts

3 In Label Font, to change default font properties for labels:

1 Choose Change. Font appears, as shown in Figure 24-4.

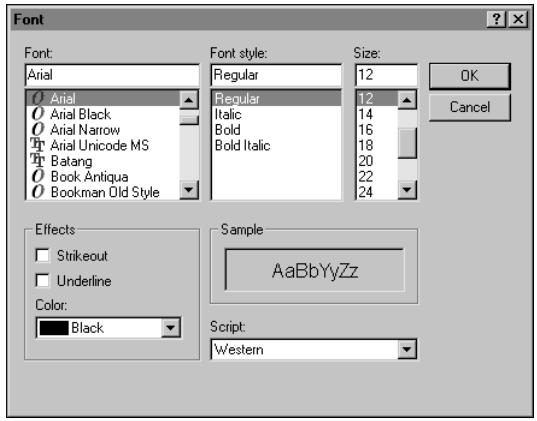


Figure 24-4 Setting default label fonts

2 Select a font, font style, size, font effect, or script type. Choose OK.

4 To revert to the original label font settings, in label Font, choose Use Defaults.

5 To change default font attributes for data, in Data Font, repeat step 3.

6 To change default font attributes for source editor text, in Source Editor Font, repeat step 3. Choose OK.

Setting the default font for a chart

When you build a chart, e.Report Designer Professional uses default fonts to create chart title, axis title, label, and legend text. You can modify a default font property, such as font style or size.

You use Options—Chart Fonts to modify default chart fonts. A font color can use different names in Options—Chart Fonts, Properties, and the Format toolbar. For example, fully saturated green is called Green in Properties and Lime in Font. For more information about color, see AcColor in *Programming with Actuate Foundation Classes.*

How to set a default font for a chart title, label, or legend

1 Choose Tools➛Options.

2 In Options—Design Editor, choose Chart Fonts. Options—Chart Fonts appears, as shown in Figure 24-5.

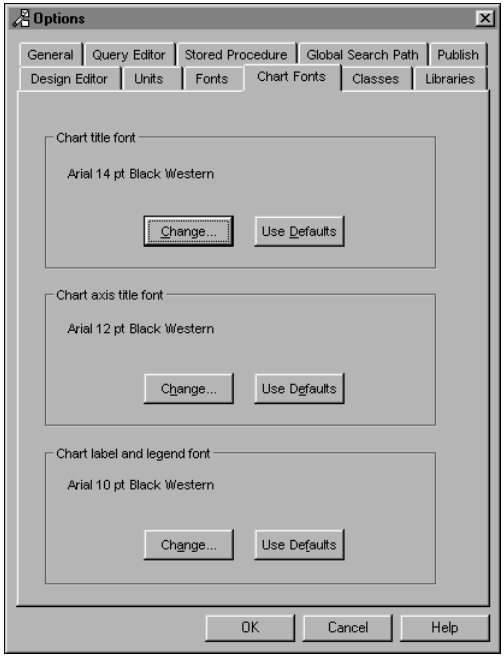


Figure 24-5 Setting default chart fonts

3 In Chart Title Font, to change default font attributes for chart titles:

1 Choose Change. Font appears, as shown in Figure 24-6.

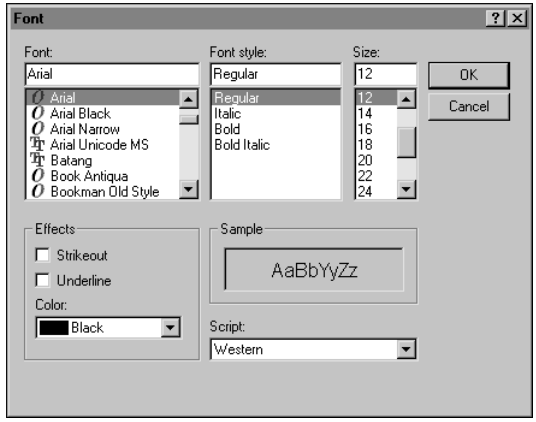


Figure 24-6 Setting default chart title fonts

2 Select a font, font style, size, font effect, or script type. Choose OK.

4 To revert to the original chart title font settings, in Chart Title Font, choose Use Defaults.

5 To change default font attributes for axis titles, in Chart Axis Title Font, repeat step 3.

6 To change default font attributes for label and legend text, in Chart Label and Legend Font, repeat step 3. Choose OK.

Setting the font for Report Structure

You can change the font face and font size that appear in Report Structure using Windows registry key settings. Use caution when you edit Windows registry keys. Changes to registry key entries and values can affect the Windows operating system. Before you begin, back up your system's registry file.

You can change the default font and size for Report Structure by setting the following Windows registry keys:

- HKEY_CURRENT_USER/Software/Actuate/e.Report Designer Professional 11.0/Settings/TreeViewFontName
- HKEY_CURRENT_USER/Software/Actuate/e.Report Designer Professional 11.0/Settings/TreeViewFontSize

If you do not set these registry keys, e.Report Designer Professional uses as the default value the font name and size shown in Table 24-1.

Table 24-1 Report Structure default fonts and sizes

Locale	Font name	Font size
Chinese (Simplified)	MS Song	12
Chinese (Traditional)	MingLiU	12
Japanese	MS Gothic	12
Korean	Gulim	12
Other languages	MS Sans Serif	8

If your machine uses Arial Unicode MS font, e.Report Designer Professional uses it with the font size 8 as the default settings for other languages.

Setting units of measurement

You can choose the unit of measurement to work with when resizing and positioning components. These units apply to properties such as size and position, and to the grid lines that you can display or hide on Layout. This option supports working with the units with which you are most familiar.

When specifying the units of measurement, you can also specify some of the display options. You can, for example, choose to add a unit suffix, such as cm,

after the numeric value. You can also specify the number of digits the application displays after the decimal point. This option is for display only and does not affect the mathematical precision that e.Report Designer Professional uses in its calculations.

Setting the default unit of measurement

Actuate e.Report Designer Professional uses and displays points as the default unit of measurement when you type values for properties such as size and position. You can, however, assign a different unit as the default unit of measurement for data entry and display. For internal use, e.Report Designer Professional converts the new default unit to twips.

You can assign one of the following units of measurement that e.Report Designer Professional recognizes:

- Twips, the unit Actuate Basic uses internally, where 1 twip equals 1/20 point
- Points, where 1 point equals 1/72 inch
- Inches
- Millimeters, where 1mm equals 0.1 cm
- Centimeters, where 1 inch equals 2.54 cm

How to set units of measurement

1. Choose Tools➛Options.
2. In Options—Design Editor, choose Units. Options—Units appears, as shown in Figure 24-7.

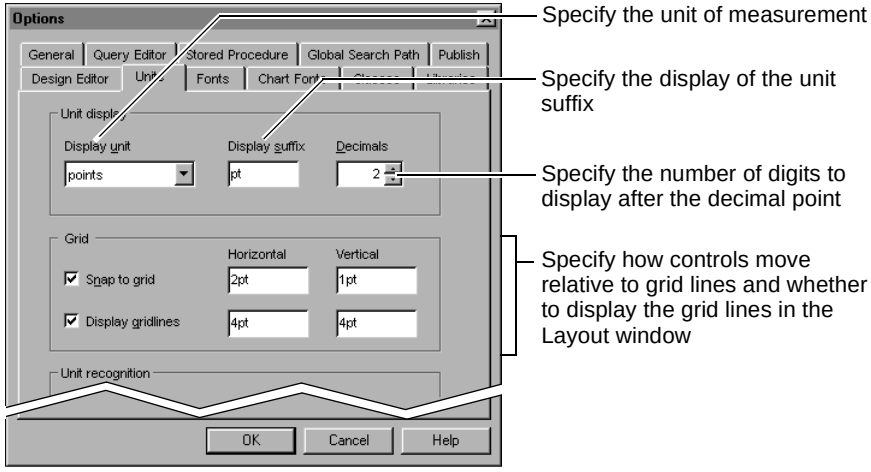


Figure 24-7 Setting units of measurement for data entry and display

3 In Unit Display, set the display options:

1 From Display Unit, select a unit of measurement.

2 To display a unit suffix after the value, such as in for inches or cm for centimeters, type the suffix in Display Suffix.

For example, if you select inches as the display unit and you want to display the abbreviation in for inches after the unit value, type:

```
in
```

e.Report Designer Professional displays 2.23in for the value 2.23. To display a space between the value and the suffix, type a space before the suffix.

3 Under Decimals, set the number of digits to display after the decimal point.

Customizing unit suffixes

The default unit of measurement that you set on Options—Units is the unit e.Report Designer Professional displays by default when you type numeric values for properties such as size and position. You can, however, explicitly choose a different unit to display when you type a value. For example, even if the default unit of measurement is inches, e.Report Designer Professional accepts data in centimeters if you add the suffix cm after the value.

By default, e.Report Designer Professional recognizes certain suffix strings, as Table 24-2 shows.

Table 24-2 Recognized suffix strings

Unit	Default recognition string
Twips	tw
Points	pt, pts
Inches	in or "
Millimeters	mm
Centimeters	cm

How to specify the custom unit suffixes

1 Choose Tools➤Options.

2 In Options—Design Editor, choose Units. Options—Units appears, as shown in Figure 24-8.

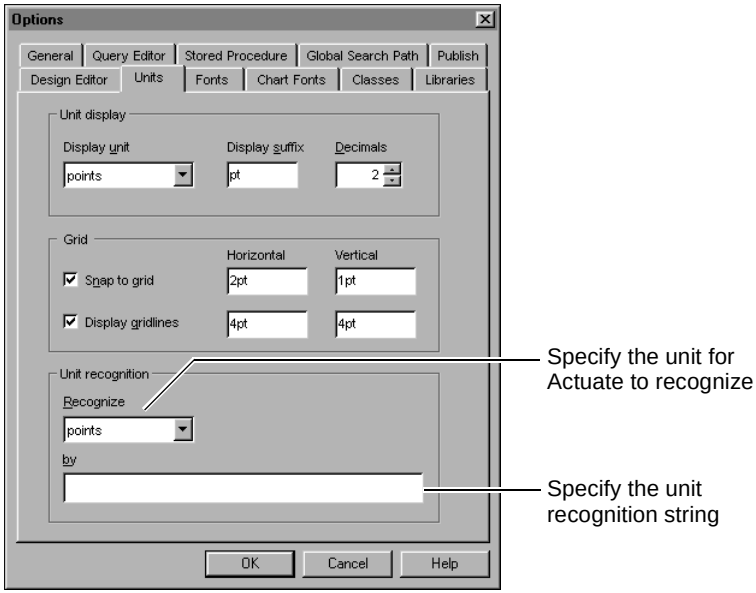


Figure 24-8 Specifying custom unit suffixes

3 In Unit Recognition:

1. In the Recognize list, select the unit that you want the application to recognize.
2. In by, type the string for e.Report Designer Professional to use to recognize the unit. Specify a recognition string only if it differs from e.Report Designer Professional's default strings.
3. Repeat substeps 1 and 2 until you specify unique strings for all units that you want e.Report Designer Professional to recognize.

4 Choose OK.

Setting design search paths

Actuate applications use relative file paths to locate included files. With relative file paths, you can specify a full path, a partial path, or just the file name when including files such as images, libraries, and Actuate Basic files. If the included file is not in the directory specified by the full path, e.Report Designer Professional searches other directories that you specify.

The order in which Actuate searches directories is always the same, but the type of files included determines which search paths e.Report Designer Professional

uses. Depending on the type of file included, e.Report Designer Professional searches some or all of the following directories and paths:

- Directory that contains the referring file
  For example, your design and included library files are in C:\Program Files\Actuate11. You move them to C:\My Reports. When the referring design file opens, e.Report Designer Professional searches C:\My Reports for the included library file.
- Directory specified in the ExtendSearchPath function
  You can use the ExtendSearchPath function to specify additional directories to search when locating images for report generation or viewing. For more information about the ExtendSearchPath function, see *Programming with Actuate Foundation Classes.*
- Design search path
  The design search path specifies the directories to search for the referring design file. e.Report Designer Professional does not use design search path for other design files.
- Global search path
  The global search path specifies the directories to search for any design file, including the referring file.
- Configuration file search path
  You can use configuration files to specify additional directories to search.
- Current working directory
  The current working directory is the directory in which you last saved a file or opened a file.

How to choose a design search path

1. In e.Report Designer Professional, open a report object design (.rod) file.
2. Choose Report➛Design Properties.
3. In Design Properties—Report Settings, choose Design Search Path.
4. In Design Properties—Design Search Path, choose New. A field appears in Directories.

5. Type the directory name, or choose Ellipsis to navigate to and select the directory, as shown in Figure 24-9.

   The directory that you provide appears at the bottom of the list. e.Report Designer Professional searches directories in the order they appear in the list.

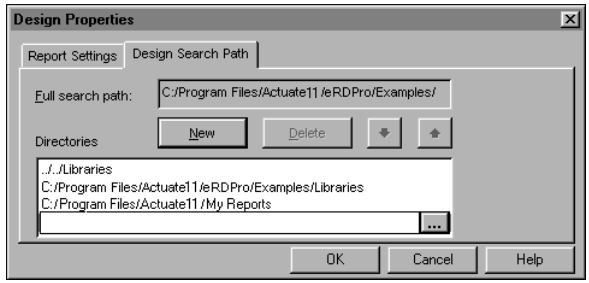


Figure 24-9 Selecting a design search path

6 To change the search order, select a directory and choose the up or down arrow.

7 To remove a directory from the design search path, select the directory and choose Delete.

8 Choose OK.

How to set a global search path

1 Choose Tools➛Options.

2 In Options—Design Editor, choose Global Search Path. Options—Global Search Path appears, as shown in Figure 24-10.

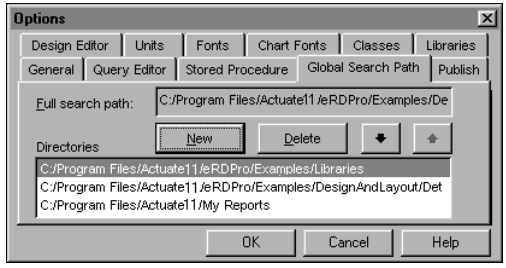


Figure 24-10 Selecting a global search path

3 Choose New. A field appears in Directories.

4 Type the directory name, or choose Ellipsis to navigate to and select the directory. The directory that you choose appears at the bottom of the list. e.Report Designer Professional searches the directories in the order they appear in the list.

5 To change the search order, select a directory and choose the up or down arrow.

6 To remove a directory from the global search path, select the directory and choose Delete.

7 Choose OK.

Setting general options in the designer

In Options—General in e.Report Designer Professional, you can:

- Set the number of recently opened files that appear in the File menu.
- Set the configuration file that the design uses.
- Set the default locale that the designer uses. For information about locales, see *Working in Multiple Locales using Actuate Basic Technology.*
- Set Actuate Management Console's URL that the designer uses to access an Encyclopedia volume.
- Set Basic source encoding.
- Reset the active perspective or all perspectives to the default settings that appear for a new e.Report Designer Professional installation. These settings affect the design, data, debug, and view perspectives.

How to change settings on Options—General

1. Choose Tools➛Options.
2. In Options—Design Editor, choose General. Options—General appears, as shown in Figure 24-11.

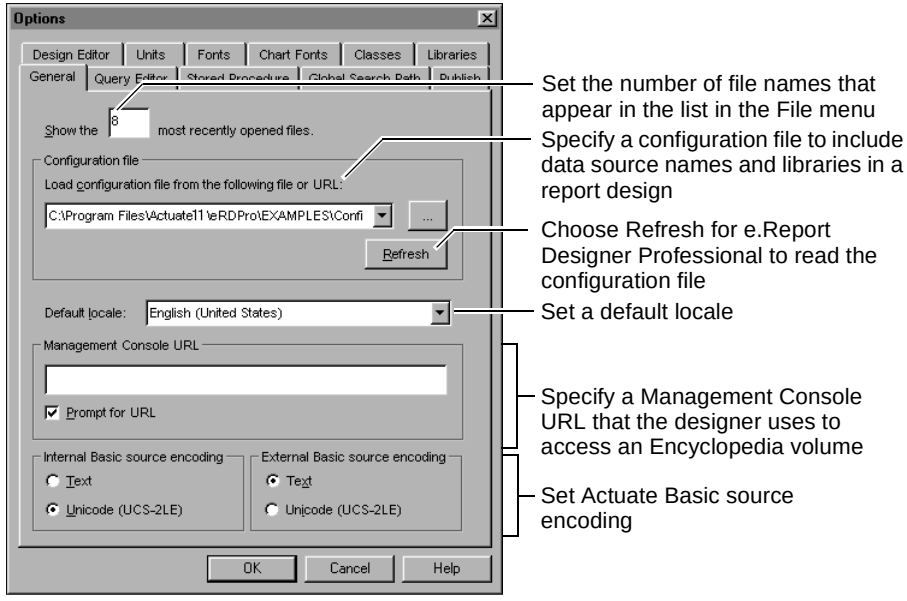


Figure 24-11 Changing settings on Options—General

3. Provide new information or change the existing information. Choose OK.

Customizing the length of the recently opened files list

You can specify the length of the recently opened files list on Options—General. The recently opened files list appears at the bottom of the File menu, as shown in Figure 24-12.

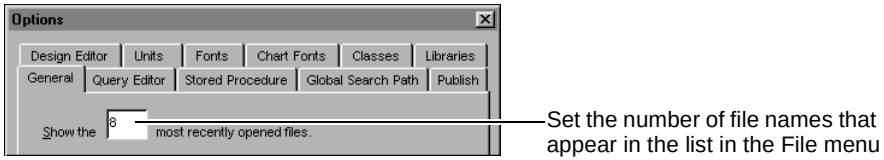


Figure 24-12 Recently opened files list

For example, in Show the <number> most recently opened files, if you type 3, the recently opened files list displays three files, as shown in Figure 24-13.

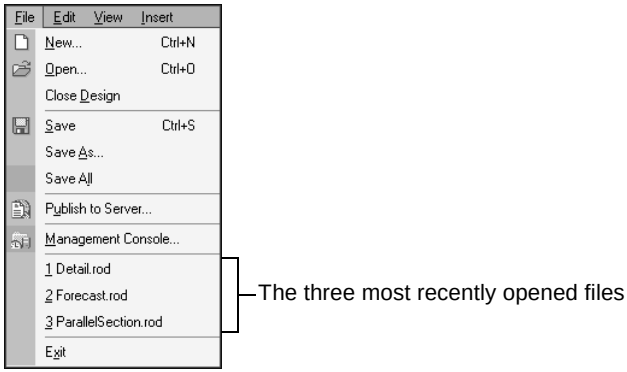


Figure 24-13 Customized recently opened files list

You can open any file on the recently used files list by choosing it from the File menu.

Setting the Management Console URL

You can set up e.Report Designer Professional to support accessing an Encyclopedia volume without providing a URL in Management Console URL on Options—General. To set up the Management Console URL, type the URL in Management Console URL and deselect Prompt for URL.

For example, if the web server and Actuate iServer reside on a machine named caligari using the default settings for the installations, the Management Console URL is:

```
http://caligari:8900/acadmin/login.jsp?serverURL=http:
   //caligari:8000&daemonURL=http://caligari:8100
```

For information about the URL to use to access your Encyclopedia volume, see your Actuate iServer System administrator.

Setting Basic source encoding

The setting for Basic source encoding specifies encoding in the Actuate Basic source (.bas) file used to create the report object executable (.rox) file. e.Report Designer Professional uses the selection in Internal Basic source encoding to create the Basic source (.bas) file from the report design file. In Options—General, for internal Basic source encoding, select one of the following options:

- Text
- Unicode (UCS-2LE)

If your data source contains Unicode, you must select Unicode (UCS-2LE). If you select Text, you can edit the BAS file using a text editor. If you use a custom external BAS file, you must specify the encoding used in that file.

Customizing the classes of the toolbox

An Actuate Foundation Class associates with each component in the toolbox. You can change the class associated with each component in the toolbox and use a custom class that suits your work environment. The custom class that you choose must be a subclass of the default class for the component.

In e.Report Designer Professional, you can modify the toolbox in the following ways:

- Add a component.
- Add a component group.
- Change the class with which a component associates.
- Save a class representation.
- Read a previously saved class representation.

You can, for example, combine the various components into one group that contains only the components that you frequently use or alternatively, you can replace some of the Actuate Foundation Classes with custom classes.

At any time after you modify settings, you can restore the toolbox and the classes associated with the components to their default values.

How to add a component to the toolbox

1. Right-click a component in the toolbox. Choose Add Component, as illustrated in Figure 24-14.

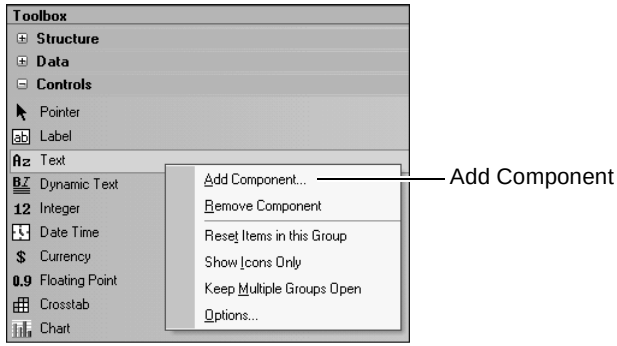


Figure 24-14 Modifying the toolbox

2 In Add Component, from the list of components, select a component. Choose OK. The component appears in the toolbox.

How to add a group to the toolbox

1 Right-click a toolbox group name. Choose Add Group, as shown in Figure 24-15.

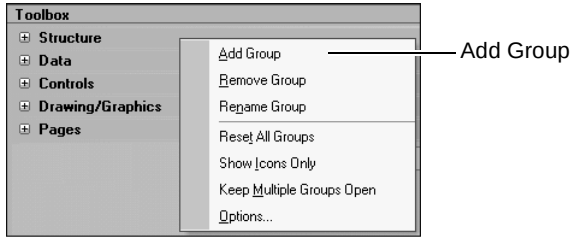


Figure 24-15 Adding a group to the toolbox

2 In Group1, replace Group1 with a name for the new group, as shown in Figure 24-16.

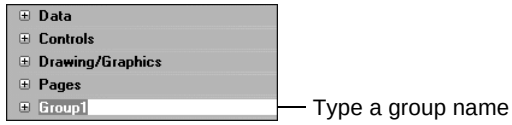


Figure 24-16 Changing the group name

Press Enter. The new group name appears in the toolbox.

How to restore the default toolbox layout

1 To restore the toolbox group names, right-click a toolbox group. Choose Reset All Groups. The default toolbox group names appear.

2 To restore the component names in a group, right-click in a toolbox group. Choose Reset Items in this Group. The default names appear in the group.

Customizing the class association of the toolbox

You can add custom classes to your toolbox by adding components and associating these newly created components with your own frequently used classes. Choosing custom classes directly from the toolbox saves time when designing reports because you do not have to choose these classes from a library.

For example, you can add your own currency control and set that control as the base control for currency columns in a database. When you create a currency control from the toolbox or drag a currency column from Fields, e.Report Designer Professional uses the custom currency control instead of the default Actuate Foundation Class currency control.

How to change the class associated with a component in the toolbox

1 Right-click in the toolbox. Choose Options, as shown in Figure 24-17.

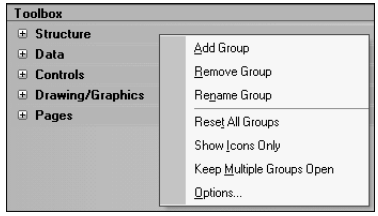


Figure 24-17 Changing class associations in the toolbox

Options—Classes appears, as shown in Figure 24-18.

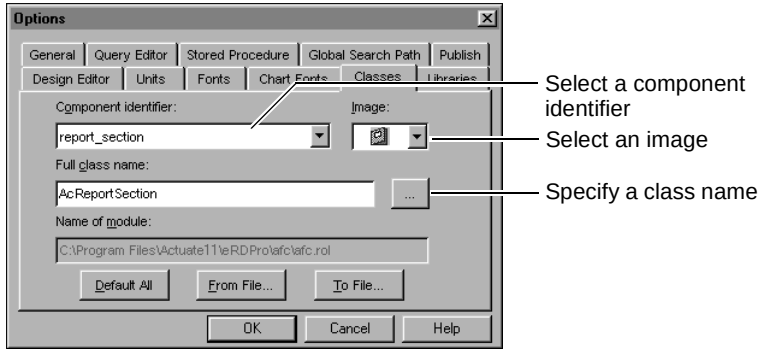


Figure 24-18 Associating components with custom classes

2 Modify the component settings as follows:

- From the Component identifier drop-down list, select the component to modify or select a custom class.
- To change the default image associated with this component, select an image from the image list. You can select only images supplied with the application.

- Under Full class name, type the class name that you want to associate with this component.
  - The class associated with the component must derive from the default class for the component.
  - When you add your own class name and this class name does not have a global scope, you must specify the qualified class name, for example, ReportApp::OrderFrame::OrderIDControl.
  - To reuse the component in other report designs, you must place the class in a library.

You can also choose Browser to display a list of global classes from which you can select a class.

When you finish modifying the class or classes, choose OK.

How to restore the default class association of all components in the toolbox

1. Right-click a component in the toolbox. Choose Options. Options—Classes appears.
2. Choose Default All, as shown in Figure 24-19.

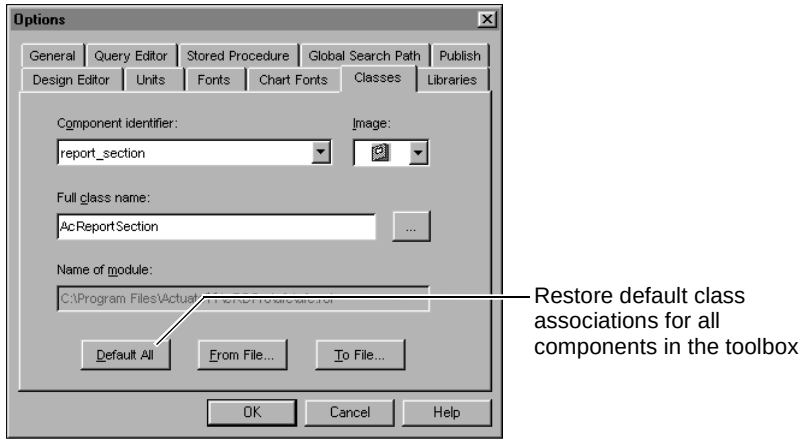


Figure 24-19 Restoring default class associations

Choose OK.

Saving and reusing class associations of components

You can create a modified set of components in the toolbox and share these with others in your workgroup. After you associate classes with components in the toolbox, you can save the configuration to a Class Representation (.apr) file. Others in your work group can use this APR file to duplicate the toolbox classes. You can create different representations for a variety of recurring tasks. By saving

each of the class representations to an APR file, you can restore them later by reading the appropriate APR file.

How to save a toolbox representation to a file

1 Right-click the toolbox. Choose Options. Options—Classes appears, as shown in Figure 24-20.

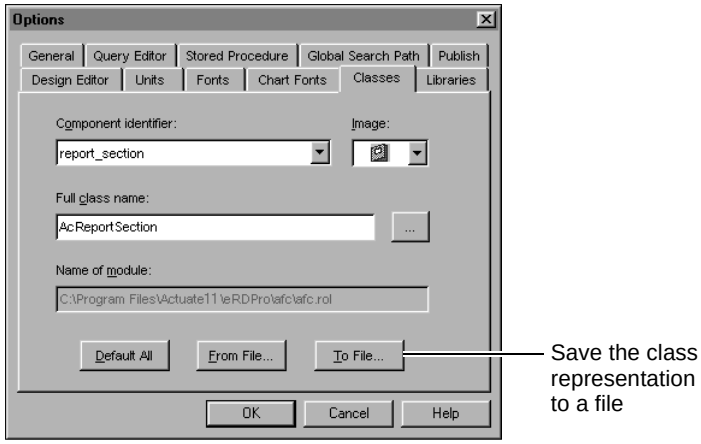


Figure 24-20 Saving a toolbox representation to a file

2 Choose To File. Save Class Representation appears.

3 In Save Class Representation:

- In File name, type the name of the file in which to save your class associations.
- In Save as type, select Class Representation (*.apr), as shown in Figure 24-21. Choose Save.

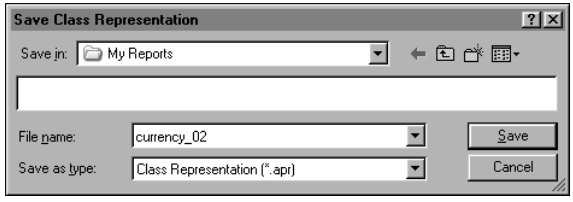


Figure 24-21 Specifying toolbox representation name, type, and location

4 In Options—Classes, choose OK.

How to apply a previously defined toolbox representation

1 Right-click the toolbox. Choose Options.

2 In Options—Classes, choose From File, as shown in Figure 24-22.

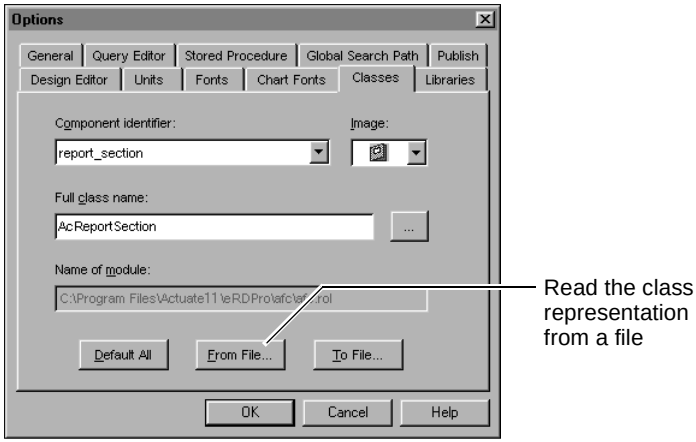


Figure 24-22 Selecting a saved toolbox representation

3 In Read Class Representation, select a Class Representation (.apr) file or type a file name, as shown in Figure 24-23. Choose Open.

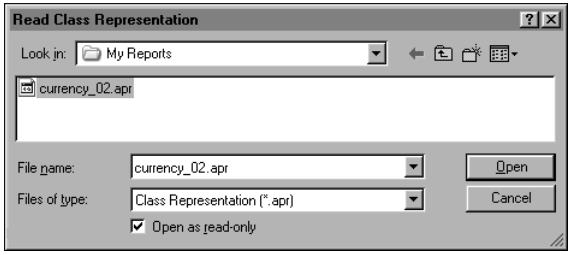


Figure 24-23 Opening a saved toolbox representation

4 In Options—Classes, choose OK.

Changing the item list for Create New Report

In e.Report Designer Professional, using the default settings, the following items appear in Create New Report:

- Quick Report Wizard
- Listing Report Wizard
- Blank Report
- Component Library
- Sample Template

The first four items provide access to report wizards. You can hide report wizard items that appear in the list. The last item is a sample template provided as an example. You can add or remove templates from this list.

Hiding a report wizard item that appears in Create New Report

Using Windows registry key settings, you can hide one or more of the following default items that appear in the item list in Create New Report:

- Blank Report
- Component Library
- Listing Report Wizard
- Quick Report Wizard

Use caution when you edit Windows registry keys. Changes to registry key entries and values can affect the Windows operating system. Before you begin, back up your system.

How to hide a report wizard item using a registry key entry

1. Choose Start➛Run. Run appears.
2. Type:

   ```
   regedit
   ```

   Choose OK.
3. In Registry Editor, navigate to HKEY_CURRENT_USER\Software\Actuate \e.Report Designer Professional 11.0\Settings.
4. To hide an item in Create New Report:
   1. Right-click a registry key name that appears in Table 24-3 and choose Modify. Edit DWORD Value appears.

      Table 24-3 Registry key names

Item in Create New Report	Registry Key name
Blank Report	Hide Blank Report Wizard
Component Library	Hide Library Wizard
Listing Report Wizard	Hide Listing Report Wizard
Quick Report Wizard	Hide Quick Report Wizard

   2. In Value data, replace 0 with 1. Choose OK.
5. Close Registry Editor.

Adding or removing a template in the item list

A template is a report object design (.rod) file that provides a basis for use or for further report development. After selecting a template, a report designer can make modifications to the design. When saving a modified template, e.Report Designer Professional prompts the user to provide a path and file name.

You can add templates to the list or remove templates, including the Sample Template. The available templates appear as report creation choices when a report designer chooses File➤New, as shown in Figure 24-24.

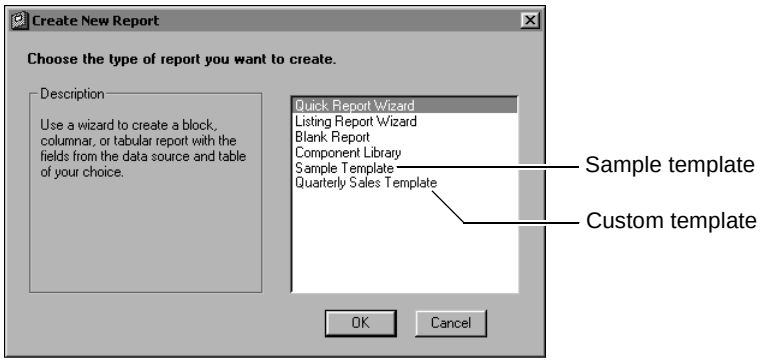


Figure 24-24 Available templates for creating a new report

You can make one or more templates available to report designs by using a configuration file. Configuration files provide easy access to report templates, data connections, and libraries. If you select a configuration file for a report design, that report design has access to the templates specified in the configuration file. This section describes how to use the section of the configuration file that make templates available to a report design. For general information about selecting a configuration file for a report design or creating a configuration file, see *Accessing Data using e.Report Designer Professional.*

You use a Template element within a configuration file's top-level Design element to specify including a template in report designs. You remove a Template element from the configuration file to remove the template from the item list of Create New Report. After modifying a configuration file, e.Report Designer Professional must reload the configuration file before the changes appear in Create New Report.

By default, Sample Template appears in the list in Create New Report because it is specified in the example configuration file provided by e.Report Designer Professional. To hide Sample Template in Create New Report, you can perform one of the following actions:

- Change the configuration file associated with the report design to another configuration file that does not include the Sample Template.

- Remove the Sample Template's Template element from the configuration file.
- Change the report design to not use a configuration file.

In the following example, the Template element:

- Makes C:\Includes\FinanceTemplate.rod available to report designs as a report template
- Specifies listing the template as Finance Reports Template in the Create New Report page.
- Provides a description to appear on the left side of Create New Report if the user selects the template.

```
<Template
   Alias="Finance Reports Template"
   Description="Template for Finance Department use.">
      C:\Includes\FinanceTemplate.rod
</Template>
```

The template's alias and description appear on Create New Report, as shown in Figure 24-25.

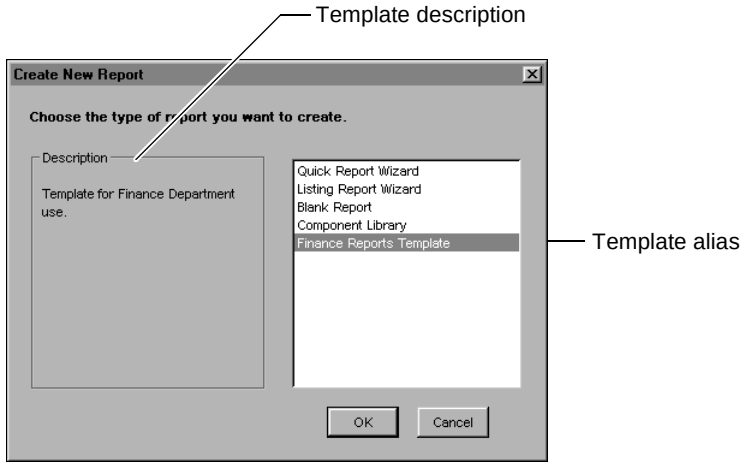


Figure 24-25 Template alias and description

You can have several Template elements, each one including a single template. To specify including a template, use the following parts of the Template element shown in Table 24-4.

Table 24-4 Template element parts and descriptions

Part within the Template element	Description
Alias attribute	Optional element that provides the text that a report designer chooses in Create New Report. If you do not set Alias, Create New Report displays the file name for the template, without the path or the .rod file-name extension.
Description attribute	Optional element that provides the text that appears on the left side of Create New Report when you select the template. Typically, this text contains the content or purpose of the template.
Element value	Specifies the path and file name of the template. The path can be absolute or relative to the directory that contains the configuration file.

Part Five

Integrating report output

Chapter

25

Embedding report content in a web page

This chapter contains the following topics:

- About embedding report content in a web page
- Identifying the report content to display in a web page
- Importing report content using a search URL
- Working with a Reportlet

About embedding report content in a web page

You can access, retrieve, and display a portion of an Actuate Basic report in a web page. A Reportlet is the portion of an Actuate Basic report that you import from a report document. One use is to incorporate the Reportlet into a web page. When a user opens the web page, the Reportlet content is integrated into the web page.

Using a Reportlet requires the following conditions:

- Both the report object instance (.roi) file and the report object executable (.rox) file must be in an Encyclopedia volume.
- Importing a Reportlet from the ROI file requires specific syntax for the search URL that Actuate Information Console uses.
- The web browser must be able to display the HTML web page and the container in which the Reportlet appears.

Figure 25-1 shows an example of the opening page of Detail.roi as a Reportlet in a web page. This example uses an IFRAME element to embed the Reportlet in the web page.

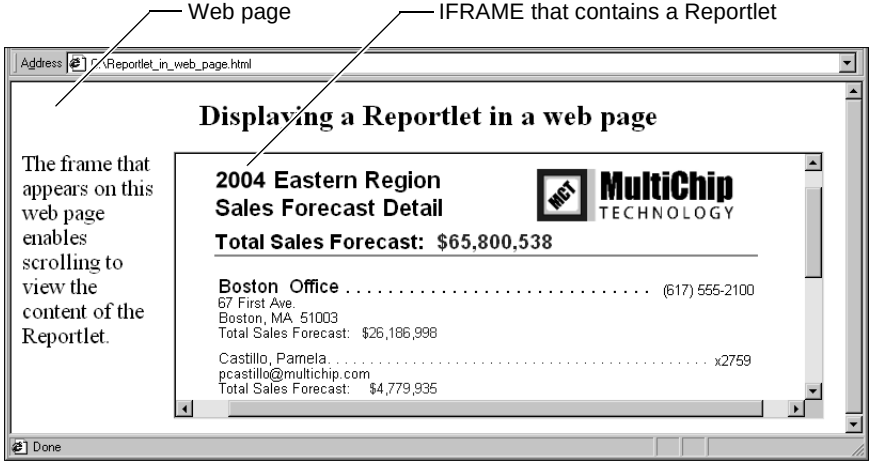


Figure 25-1 Detail.roi as a Reportlet in a web page

About the source HTML for the web page example

The web page that appears in Figure 25-1 contains the following source HTML:

```
<HTML>
   <HEAD>
      <TITLE>Displaying a Reportlet in a web page</TITLE>
   </HEAD>
```

```
   <BODY>
      <CENTER><H2><FONT SIZE=6>Displaying a Reportlet in a web
         page</FONT></H2></CENTER>

   <IFRAME SRC="http://localhost:8700/iportal/servlet
   /ViewPage?page=1&format=Reportlet&name=Detail.roi&type=roi
   &scalingFactor=60 "NAME "A" HEIGHT=250 WIDTH=80% HSPACE=10
   SCROLLING=YES ALIGN=RIGHT>
   Your browser cannot display content in an IFRAME
   </IFRAME>

   <P><FONT SIZE=5>The frame that appears on this web page
      enables scrolling to view the content of the Reportlet.
      </FONT>
   </BODY>
</HTML>
```

The HTML formatting tags in a Reportlet differ from the tags in a DHTML report that a user views in a browser. The Reportlet HTML does not contain <HTML>, </HTML>, <HEAD>, </HEAD>, <BODY>, and </BODY> tags. A Reportlet does not support JavaScript. The Reportlet's <DIV> and </DIV> tags designate the divisions between portions of the Reportlet content. These divisions support adding a Reportlet to a web page.

About supported components for embedded report content

From a report object instance (.roi) file, you can import each of the following items as a Reportlet:

- Report section
- Group section
- Sequential section
- Page
- Frame

When you design a web page to import a Reportlet from a report object instance (.roi) file, consider the following content- and appearance-related issues:

- You cannot import a control, such as a chart, as a Reportlet. You can, however, add one or more controls to a frame and import the frame as a Reportlet.
- A report designer sometimes includes a control such as a column heading or label in a page header or footer. When you import a Reportlet that contains a page header or footer, consider where that page header or footer appears in the web page content. The following conditions determine the appearance of a page header or footer:

- If the target for a search URL contains page headers, and the Visibility➤ShowInReportlet property value is True, the first page header at the beginning of a report or group section appears in the Reportlet.
- If the target for a search URL contains page footers, and the Visibility➤ShowInReportlet property value is True, the first page footer at the end of a report or group section appears in the Reportlet.

About changes to report capabilities in embedded report content

Some of the following Actuate report capabilities change when you import content as a Reportlet:

- A Reportlet spans the full width of the frame.
- The view perspective toolbar does not appear. Because there is no toolbar, there is no access to the table of contents, search functionality, and PDF generation.
- Choosing a hyperlink in a Reportlet opens a web page in the same browser window.
- Browser Scripting Controls in a Reportlet do not work.
- Actuate generated JavaScript does not work.

Identifying the report content to display in a web page

To import a portion of a report as a Reportlet and display its content on a web page, you can

- Set the Visibility➤ShowInReportlet property of components to show or hide them in the Reportlet.
- Set the Searching➤SearchValueExp property value to identify a portion of a report to import as a Reportlet.

Displaying or hiding report content that is embedded in a web page

In the report content that you embed in a web page, you can specify the visibility of each component.

How to set visibility property values for a Reportlet

1. Right-click a report component and choose Properties. The Properties page for the component appears.

2 Set the Visibility➤ShowInReportlet property to one of the following values:

- True is the default value, which enables a report component to appear in a Reportlet.
- False hides a report component in a Reportlet.

Identifying the report content to embed in a web page

To identify the report content to embed in a web page, you can set the Searching➤SearchValueExp property value to enable dynamically identifying a component in a Reportlet.

For example, Figure 25-2 shows the layout and properties for a frame. The Searching➤SearchValueExp property value is [customers.customName]. Using an Actuate Information Console directive, the value of [customers.customName] determines the content of the frame to import as a Reportlet.

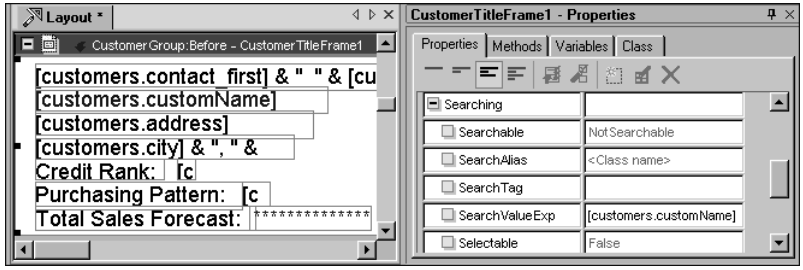


Figure 25-2 Layout and properties for a frame

How to identify the report content to embed in a web page

1 Right-click a report component that you want to import as a Reportlet. Choose Properties. The Properties page for the component appears.

2 For the Searching➤SearchValueExp property value, use one of the following types of values:

- Use a constant String value to identify a component that appears in one instance in the report. For example, you can use the following value to identify a frame that contains one instance of a chart:

  ```
  "TheChart"
  ```

- Use a dynamic String value to identify a component that changes depending on the search URL value that accesses it. For example, you can use the following value to identify a frame that contains a chart that changes based on the customers.customName values from the data row:

  ```
  [customers.customName]
  ```

Importing report content using a search URL

For a section, page, or frame, a search URL identifies the portion of a report to import as a Reportlet. The Factory converts the content of the Reportlet to HTML for viewing in a browser. The following sections explain the syntax that you use to import a report page or a report section into a web page.

Importing a report page into a web page

The search URL that you use to import a section or frame as a Reportlet differs from the search URL that you use to search a DHTML report document on the web. An Actuate Information Console search URL that imports a page as a Reportlet uses the following syntax:

```
http://<web server>:<port number>/iportal/servlet
   /ViewPage?format=Reportlet&name=/<folder>/<file name>&type=
   <file type>&searchcriteria=<component name>=<component value>
   &scalingFactor=<scale>
```

To import a page that contains a frame to view its contents in a web page, you can use syntax similar to the following URL:

```
http://<web server>:<port number>/iportal/servlet
   /ViewPage?format=Reportlet&name=Detail.roi&searchcriteria=
   OrderTitleFrame::OrderNumber=1810&type=roi&scalingFactor=60
```

where

- ViewPage specifies that the Reportlet displays a page of the report.
- format=Reportlet specifies displaying report content in Reportlet format.
- name=detail.roi specifies the report document name.
- searchcriteria=OrderTitleFrame::OrderNumber=1810 specifies importing the page that contains the frame named OrderTitleFrame that contains the OrderNumber control with the value 1810.

  In this URL, in the search conditions parameters syntax, the first equality uses an equals sign (=). As in this example, if the search conditions parameters require more than one equality, use the %3D hexadecimal equivalent for the equals sign for equalities after the first one.

The following example uses an IFRAME element to embed a Reportlet in a web page:

```
<IFRAME SRC="http://siamese:8700/iportal/servlet
   /ViewPage?format=Reportlet&name=detail.roi
   &searchcriteria=OrderTitleFrame::OrderNumber=1810
   &scalingFactor=65" NAME "A" HEIGHT=350 WIDTH=70% HSPACE=10
   SCROLLING=YES ALIGN=RIGHT>Your browser cannot display the
   content in an IFRAME</IFRAME>
```

A web page that contains the preceding HTML imports the Reportlet and displays it in a frame, as shown in Figure 25-3.

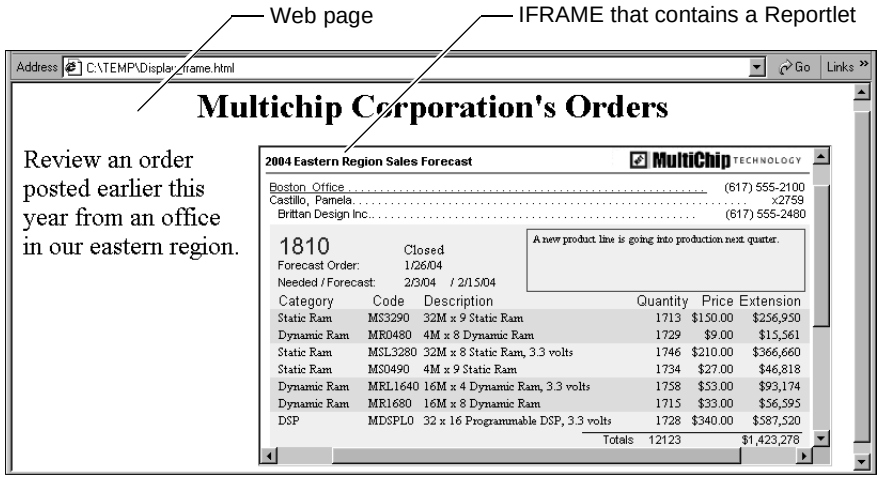


Figure 25-3 Reportlet in a web page

Importing a report section into a web page

To import a report section as a Reportlet, you can use the GetReportData directive and specify the values for componentName and componentValue. A search URL that imports a report section uses the following syntax:

```
http://<web server>:<port number>/iportal/servlet
   /GetReportData?format=Reportlet&name=/<folder>
   /<file name>&version=<version number>&type=<file type>
   &componentName=<component name>&componentValue=<component
   value>&scalingFactor=<scale>
```

For example, the following IFRAME SRC specifies importing the OrderGroup1 group section component to display it as a Reportlet:

```
<IFRAME SRC="http://athena:8700/iportal/servlet
   /GetReportData?format=Reportlet&name=/Detail
   /Detail01.roi&componentName=SalesDetail::OrderGroup1
   &componentValue=1340&scalingFactor=60" NAME "A" HEIGHT=250
   WIDTH=80% HSPACE=10 SCROLLING=YES ALIGN=RIGHT>
```

where

- GetReportData specifies that the Reportlet displays a part of the report.
- format=Reportlet specifies displaying report content in Reportlet format.
- name=Detail01.roi specifies the report document name.

- componentName=SalesDetail::OrderGroup1 specifies the scope and the class of the group section.
- componentValue=1340 specifies a value of the group key and the Searching➤SearchValueExp for this component.

Alternatively, you can use a componentID value to identify a component to extract as a Reportlet. Because the componentID is dynamic and can change from one report object executable (.rox) file run to another, identifying a part of a report by componentID can cause unexpected results.

For more information about using a search URL with Actuate Information Console, see *Information Console Developer Guide.*

Working with a Reportlet

A Reportlet supports inserting a portion of a report document into an HTML web page. Presenting the report content in a web page changes how you work with the report. The following sections explain how to work with a Reportlet in a web page.

Printing and downloading a Reportlet

You can print or save the web page that contains a Reportlet as an HTML document. You cannot download or print the Reportlet itself in PDF format.

Enabling a hyperlink in a Reportlet

Actuate iServer supports using absolute and relative hyperlinks in a Reportlet. Absolute and relative URLs are supported. In the web page, a relative URL is relative to the location of the web page. When a search URL imports a Reportlet from a report document, relative URLs in the Reportlet become relative to the web page location. Relative URLs are not relative to the location of the report document from which the web page imports the Reportlet.

To avoid unexpected results, use absolute URLs in reports from which you plan to import a Reportlet.

Chapter

26

Providing web functionality in a report

This chapter contains the following topics:

- Using web browser code inside a report
- Including custom browser code in a report design
- Including global custom browser code
- Creating an HTML form
- Using Flash within a browser scripting control
- Generating custom browser code dynamically
- Printing and viewing a browser scripting control in non-DHTML formats
- Understanding an example browser scripting library for Internet Explorer

Using web browser code inside a report

A report can display and use any type of item that you see on a web page. A browser scripting control supports writing code for a web browser inside a report design.

You must provide code for the browser scripting control to instruct the browser to display an item, interact with the user, or perform specific actions. You can write the code in e.Report Designer Professional or use another tool to generate an external source code file or an applet. The code can be in any form that a web browser can interpret, including:

- HTML
- JavaScript
- Java applets
- VBScript

A web browser interprets custom browser code in a report when a user views the report in DHTML format.

You can also use browser code elsewhere in e.Report Designer Professional. Global custom browser code supports calling browser code functions from anywhere in a report. You also can convert a frame into an HTML form.

Including custom browser code in a report design

The browser scripting control supports including custom browser code in a report design. This control is in the Controls toolbox, as shown in Figure 26-1.

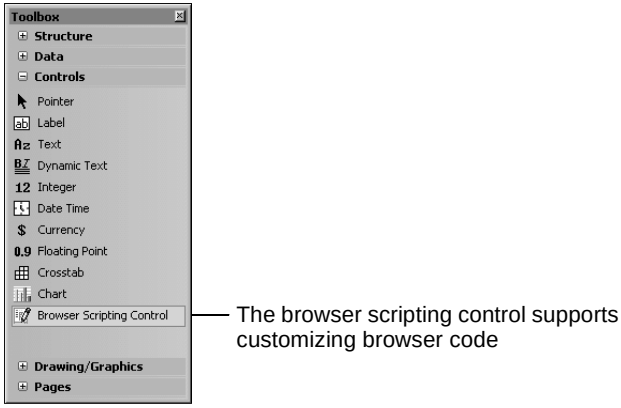


Figure 26-1 Controls toolbox

The DHTML converter treats the browser scripting control differently from other controls. Ordinarily, the DHTML converter escapes characters that have special meaning for the web browser, such as:

- <
- >
- "
- \n
- Spaces

For a browser scripting control, however, the DHTML converter does not convert special characters. For example, if a label control contains the following string:

```
<b>MyText</b>
```

the DHTML converter converts the string to:

```
&lt;b&gt;MyText&lt;/b&gt;
```

The string then goes to the web browser, which converts the string back to:

```
<b>MyText</b>
```

Figure 26-2 shows this conversion process.

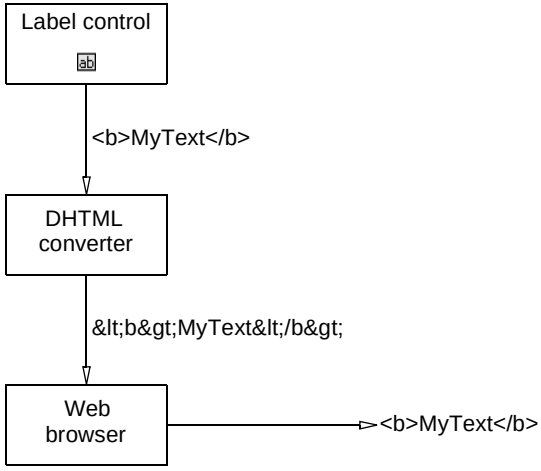


Figure 26-2 Conversion process for a label control

If a browser scripting control contains the same string:

```
<b>MyText</b>
```

the DHTML converter creates a block of HTML code called the context block. The context block contains the original string:

```
<b>MyText</b>
```

The DHTML converter does not convert special characters in the browser scripting control. Figure 26-3 shows the conversion process for a browser scripting control.

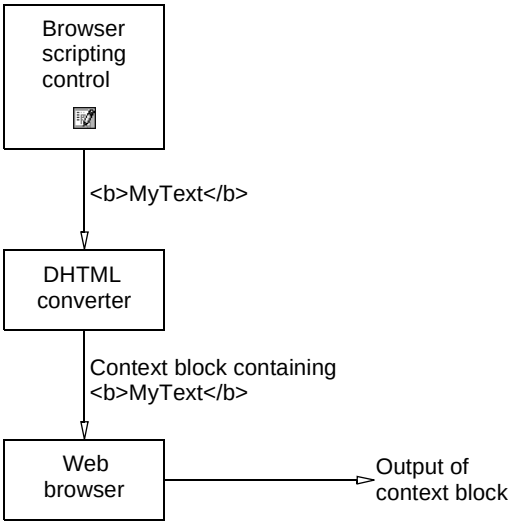


Figure 26-3 Conversion process for a browser scripting control

About the context block

The DHTML converter generates a context block, a block of HTML code that contains the custom browser code. The context block is the custom browser code's immediate parent in the HTML document's hierarchy of objects. The context block applies a browser scripting control's visual properties, such as background color and border style. In Microsoft Internet Explorer, the context block is in the DIV element.

The following example shows a context block in Microsoft Internet Explorer:

```
<DIV    ID="I550"  CLASS=C148
onMouseOver="mouseOver (…) "
STYLE="position:absolute;
left:36.00pt;        top:54.00pt;        height:73.00pt;
   width:127.00pt;
font-size:8.00pt;
text-align:left;
overflow:hidden;
padding-top:0.00pt;
">
```

```
      <!— [START Custom browser code -- >
      <SCRIPT LANGUAGE="JavaScript">
           document.write("...")
      </SCRIPT>
      <!— END] Custom browser code -->
</DIV>
```

How to include custom browser code in a report design

1. Drag a browser scripting control from the Drawing and Graphics toolbox and drop it in a frame.
2. Double-click the control. BrowserScriptingControl—Properties appears.

3. Select BrowserCode. Choose Ellipsis.
4. In Custom Browser Code Editor, type the custom browser code. For example, type:

   ```
   <b>MyText</b>
   ```

   Choose OK. If you chose to display sample data when you set up e.Report Designer Professional, the control displays the value of the SampleValue property in Design Editor. Otherwise, the control displays the value of the AlternateText property.

   The value of the AlternateText property of a control is displayed when a report is viewed in a format or environment, such as the view perspective, that does not interpret HTML and JavaScript. You can control the value of AlternateText when a user views the report by overriding the browser scripting control's GetText( ) method.

Clipping the output of custom browser code

You can specify the clipping behavior for custom browser code output using the browser scripting control's BrowserClipping property. For Microsoft Internet Explorer, the BrowserClipping setting maps to the overflow cascading style sheet (CSS) attribute for the DIV element.

Table 26-1 shows how the BrowserClipping setting maps to the overflow CSS attribute for Internet Explorer.

Table 26-1 BrowserClipping settings and behaviors for Internet Explorer

Setting of BrowserClipping property	Setting of overflow CSS attribute	Behavior in Internet Explorer
AutoScrollbar	Auto	Shows scroll bars when necessary.

(continues)

Table 26-1 BrowserClipping settings and behaviors for Internet Explorer (continued)

Setting of BrowserClipping property	Setting of overflow CSS attribute	Behavior in Internet Explorer
Scrollbar	Scroll	Always shows scroll bars, active when necessary.
ClipToControlSize	Hidden	Clips to the control size.
NoClipping	Visible	Does not clip. Resizes the DIV block as necessary.

Aligning the output of custom browser code

You can specify the alignment of custom browser code output using the browser scripting control's TextPlacement properties. The Horizontal and Vertical properties behave as they do for other textual controls. The DHTML converter ignores the following properties:

- Clip
- Ellipsis
- FillPattern
- MultiLine
- WordWrap

When a browser scripting control displays the value of the AlternateText property, all TextPlacement properties apply. A browser scripting control displays the value of the AlternateText property in a PDF Viewer or in a web browser if DebugOption is set to True.

Debugging custom browser code

Because Actuate software does not parse or check custom browser code for errors, it is possible for custom browser code output to cause a problem in a DHTML report page. Problems can include visual distortion or a JavaScript error message. To troubleshoot this type of problem, take one of the following steps:

- Set the browser scripting control's DebugOption property to True.

  Setting DebugOption to True tells the web browser to display the value of the browser scripting control's AlternateText property rather than interpret the custom browser code.

- Check the value of the BrowserCode property when the user views the report.

You can locate custom browser code in the browser source viewer by searching for tags that the DHTML converter inserts before and after custom browser code:

```
<!— [START Custom browser code -- >
.
.
.
<!— END] Custom browser code -- >
```

You can also display the custom browser code specified by a browser scripting control in the view perspective by right-clicking the control and choosing View Browser Code.

For example, if a report that contains a browser scripting control generates a JavaScript error when you view the report in a web browser, you can determine whether the custom code causes the error. Regenerate the report with DebugOption set to True and view it in the web browser again. The web browser displays the value of the AlternateText property instead of interpreting the custom browser code. If the browser does not raise the JavaScript error this time, you know that the custom browser code causes the error.

Including global custom browser code

You can include several types of global custom browser code in a report design, including:

- JavaScript functions and variables
- Common event handlers
- CSS STYLE elements

To include global custom browser code, use the GlobalDHTMLCode property of the root component of the report design to refer to a file that contains global custom browser code. Global custom browser code appears in the <HEAD> element of every DHTML page the converter generates. For more information about what the <HEAD> element can contain, see your DHTML documentation.

A browser scripting control in a frame can refer to global custom browser code. Do not place the global custom browser code in a browser scripting control. The DHTML converter generates a separate document for every page in the report. Placing global custom browser code in a frame can cause unresolved references when a web browser interprets the report page. For example, in a three-page report, page 1 contains a report header frame. The frame contains a browser scripting control that defines a global onChange event handler. Pages 2 and 3 contain content frames that contain browser scripting controls that refer to the global onChange event handler. The DHTML documents for pages 2 and 3 raise

JavaScript errors because the definition of the onChange event handler is in a different document, the one the converter generated for page 1.

How to include global custom browser code in a report design

1 Double-click the AcReport component.

2 In <ComponentName>—Properties, select DHTML➛GlobalDHTMLCode. Choose Ellipsis.

3 In Custom Browser Code Editor, type the global custom browser code or a reference to a file that contains the code, as shown in the following example:

```
<STYLE TYPE="text/css">
   /*Make menu float to the left of the text*/
   #menu {float:left; width:50pt; background:lightgrey;
      border:2px white outset; cursor:default}
   #menu.popup {position:absolute; display:none;
      background:lightgrey; border: 2px white outset;
      width:135pt; margin:2pt}
   #menu P {margin-top:0pt; margin-bottom:0pt; font:10pt}
      .over{color:navy; font-weight:bold}
</STYLE>
<script language="JavaScript">
var curPop = null;

function clearCurrent()
{
…
```

4 Choose OK.

Creating an HTML form

You can create an HTML form using browser scripting controls, such as combo boxes and lists, and a frame component. Actuate software places the custom browser code specified by the controls inside a <FORM> element. You can create a library containing frequently used browser scripting controls.

How to create an HTML form using browser scripting controls

1 Drag browser scripting controls for the form elements from the library to a frame.

2 Double-click the frame.

3 In <ComponentName>—Properties, select DHTML➛CustomDHTMLHeader. Choose Ellipsis.

4 In Custom Browser Code Editor, type the <FORM> tag in the following format:

```
<FORM METHOD=POST ACTION=http://roadrunner/iportal
   /roadrunner/MyReport.rox?Submit>
```

5 Choose OK to return to Properties.

6 Select DHTML➛CustomDHTMLFooter. Choose Ellipsis.

7 In Custom Browser Code Editor, type:

```
</FORM>
```

Choose OK.

Using Flash within a browser scripting control

Flash is an animation software package from Adobe Software. It supports the creation of lightweight animated graphics on web pages. Flash can be used in addition to HTML as a way to build interactive web sites and applications.

The rich application interface is a very visible part of Web 2.0. For business intelligence and reporting, the use of interactive tables, graphs, and gauges are important tools which allow you to try various what if scenarios on your data. Flash can be used to allow the creation of these types of tools. Flash technology provides e.Report Designer Professional rich media and interactivity features that can create new UI experiences in reports.

Flash objects can be interacted with through JavaScript and ActionScript code. Adobe Software provides an open license Flash/JavaScript integration kit. The kit supports the ability for JavaScript or ActionScript to call code in each environment.

If you are using Flash 8 or above, use the ActionScript ExternalInterface class. The ExternalInterface class enables communication between ActionScript and the Flash Player container such as an HTML page with JavaScript. ActionScript supports the ability to call any JavaScript function on the HTML page. You can pass any number of arguments of any data type, and receive a return value from the call. From JavaScript you can call an ActionScript function in the Flash Player. The ExternalInterface class is included as part of Flash 8 or later releases.

To place a Flash object into a browser scripting control, you use the same tags as you would to place Flash into any HTML page. Flash objects are only viewable when a user views the report in DHTML format, and can only be seen if the Flash plug-in has been installed into the user's web browser. You can place a Flash object in a browser scripting control with the following code:

```
<html>
<object width="550" height="400">
<param name="movie" value="banner2.swf">
<embed src="~\banner2.swf" width="550" height="400">
</embed>
</object>
</html>
```

In this example, the src parameter of the embed tag points to the location of the Flash object file, ~\banner2.swf. The Flash file, banner2.swf, has been placed into the same folder on iServer that the report has been published to. You can locate the Flash files anywhere on iServer that the report users have access to.

Generating custom browser code dynamically

You can generate custom browser code dynamically when a user views the report. To generate custom code dynamically, override the browser scripting control's BrowserCode( ) method. You also can set BrowserCode and AlternateText during report generation by overriding the BuildFromRow( ) or OnRow( ) methods.

Use GetText( ) to alter the appearance of a control's value at view time. You cannot modify the actual value of the control by changing the value of its DataValue variable. Overriding DataValue at view time is not a supported operation and can cause unpredictable behavior.

Generating output specific to the viewing environment

You can generate output specific to the current viewing environment by overriding the BrowserCode( ) and GetText( ) methods. You can use the following functions:

- GetAppContext indicates which Actuate application is currently running.
- GetReportContext indicates whether the report is in the generation, viewing, or printing stage.
- GetUserAgentString returns the unparsed user agent string the web browser sends with every HTTP request.
- GetReportScalingFactor returns the current scaling factor.

For more information about these functions, see *Programming with Actuate Basic.*

Determining the application execution context

You can use the GetAppContext and GetReportContext functions to determine the application execution context, as the following example shows:

```
Function GetText( ) As String
   Dim rptCtx   As Integer
   Dim appCtx   As Integer
   Dim str      As String
   Dim NL       As String
   NL = Chr$(10)
   appCtx  = GetAppContext( )
   rptCtx  = GetReportContext( )
   str = "Application context: " + Str$(appCtx) + NL
   str = str + "Report context: " + Str$(rptCtx) + NL + NL
   If appCtx = ServerContext And rptCtx = ViewerReportContext
      Then
   GetText = str + "We are in the view perspective! "
   Else
      GetText = str + Super::GetText( )
   End If
End Function
```

Generating output for different browsers

Use the GetUserAgentString function to generate different output for different browsers, such as Microsoft Internet Explorer, as shown in the following example:

```
Function GetText( ) As String
   Dim str As String
   Dim NL As String
   NL = Chr$(10)
   str = GetUserAgentString( )
   If Len(str) = 0 Then
      GetText = "…Not in a web browser."
   Else
      If InStr(str, "MSIE") = 0 Then
      'Not MSIE.
         GetText = str + NL + "…In Firefox"
      Else
         GetText = str + NL + "…In MSIE"
      End If
   End If
End Function
```

Adjusting for the current scaling factor

When the report user changes the scaling factor while viewing a report page in a web browser, the View process regenerates the page. The DHTML converter

scales the controls on the report page. The DHTML converter does not scale or change the output of custom browser code. You must therefore ensure that the output of custom browser code can use different scaling factors, especially 75% and 100%. You can adjust the size of the browser scripting control's rectangle to accommodate the output of custom browser code at different zoom levels.

To adjust the size of the browser scripting control's rectangle based on the current scaling factor, override the BrowserCode( ) method and use the GetReportScalingFactor function. By overriding BrowserCode( ), you can change the height and width of the HTML element that the custom browser code produces. The GetReportScalingFactor function returns the scaling factor. A value of 1.0 corresponds to 100 percent. For example, you can override BrowserCode( ) as shown in the following example:

```
+  Dim         dhtmlCode As String
   Dim         zoom As Double
   zoom = GetReportScalingFactor()
   dhtmlCode = "<SPACER TYPE=""block"" HEIGHT=" + Str$(150 *
      zoom)
   dhtmlCode = dhtmlCode + " WIDTH=" + Str$(250 * zoom) + ">"
   BrowserCode = dhtmlCode
End Function
```

Displaying values from one or more rows in a control

To display values from a single row in a control, place a browser scripting control in a content frame and override OnRow( ). To display values from multiple rows, place a browser scripting control in a header or footer frame and override BuildFromRow( ). You can override BuildFromRow( ) or OnRow( ) by setting AlternateText and BrowserCode when you generate a report.

To generate custom browser code dynamically when you generate a report, you must take the following steps:

- Create a string variable at the class or instance level.
- Override OnRow( ) or BuildFromRow( ) to set the string variable.
- Override BrowserCode( ) to return the string variable as a result.

The following code shows how to set AlternateText and BrowserCode during report generation:

```
'Declare dhtmlStr as new class variable
Dim dhtmlStr As String
Function BrowserCode( ) As String
   BrowserCode = dhtmlStr
End Function
```

```
Sub OnRow( row As AcDataRow )
   Super::OnRow( row )
      AlternateText = "Scripting Control Placeholder"
      dhtmlStr = "<P>Headline News</P>"
End Sub
```

Printing and viewing a browser scripting control in non-DHTML formats

The PDF converter and the view perspective are non-DHTML formats that support the browser scripting control. However, neither the PDF converter nor the view perspective interprets DHTML code. When a report prints or appears in these formats, e.Report Designer Professional substitutes the value of the AlternateText property for custom browser code output.

You can change this behavior by overriding the GetText( ) method. GetText( ) is called when a user views a report.

To specify the displayed string in a non-DHTML environment, take one of the following steps:

- Specify a value for the browser scripting control's AlternateText property.
  The string appears in the output as the content of the browser scripting control.
- Override the GetText( ) method.
  GetText( ) is called at view time. By default, GetText( ) returns the value of the AlternateText property.
- Override the BuildFromRow( ) or OnRow( ) method.
  The Factory calls these methods.

The string clips to the size of the control. If AlternateText is empty, and no other method is used to supply a displayed string, then nothing is displayed for the browser scripting control.

Other display properties, such as background color and border style, apply to a browser scripting control just as they apply to other textual controls. Settings for ShowWhenPrinting and ShowWhenViewing also apply.

In the view perspective, you can display the custom browser code specified by a browser scripting control by right-clicking the control and choosing View Browser Code, as shown in Figure 26-4.

Figure 26-4 View Browser Code

Understanding an example browser scripting library for Internet Explorer

It is helpful to create a library containing the following components:

- A frame component for creating a DHTML form
- Frequently used browser scripting controls

Figure 26-5 shows an example of this type of library.

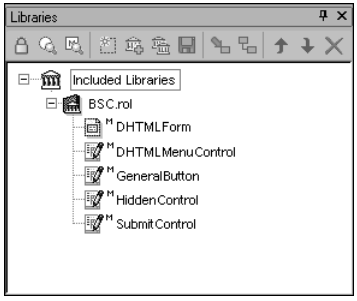


Figure 26-5 Example library for Internet Explorer

Designing a frame to support an HTML form

In this example library, the frame's class name is DHTMLForm. DHTMLForm defines several new variables and overrides two methods. Figure 26-6 shows the variables DHTMLForm defines.

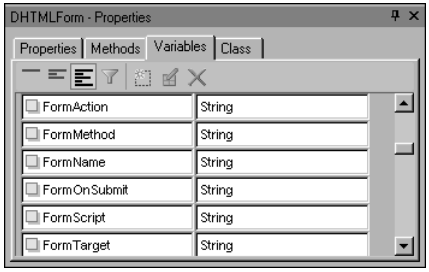


Figure 26-6 DHTMLForm variables

FormAction, FormName, FormOnSubmit, and FormTarget have a visibility of Commonly Used Property. FormMethod has a visibility of Essential Property. These variables appear on both the Variables page and the Properties pages, as shown in Figure 26-6 and Figure 26-7.

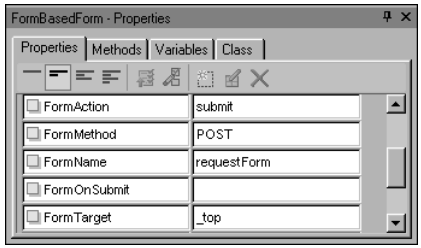


Figure 26-7 DHTMLForm properties

Overriding methods in DHTMLForm

DHTMLForm overrides the CustomDHTMLHeader( ) and CustomDHTMLFooter( ) methods. These methods generate a <FORM> element.

The following example shows how to override CustomDHTMLHeader:

```
Function CustomDHTMLHeader( ) As String
   CustomDHTMLHeader = Super::CustomDHTMLHeader( )
   CustomDHTMLHeader = "<FORM NAME= ' " + FormName + " ' METHOD=
      ' " + FormMethod + " ' ACTION= ' " + FormAction + " '
         ONSUBMIT= ' " +
      FormOnSubmit + " ' >"End Function
```

The following example shows how to override CustomDHTMLFooter:

```
Function CustomDHTMLFooter( ) As String
   CustomDHTMLFooter = Super::CustomDHTMLFooter( )
   CustomDHTMLFooter = "</FORM>" + Chr$(10) + FormScript
End Function
```

Subclassing DHTMLForm

When you subclass DHTMLForm from a library, you can set the FormAction, FormMethod, and FormName properties, as Table 26-2 shows. The CustomDHTMLHeader( ) method uses values of these properties to generate a FORM element.

Table 26-2 Subclassing DHTMLForm

Property	Value
FormAction	http://radium:8700/iportal/iportal/activePortal /viewer/executereport.do?__requesttype= immediate&__executableName= %2fSales%20Conferences%2f SummaryTimeSeries%2erox
FormMethod	POST
FormName	FormBasedForm

The example in the preceding table submits a request for immediate report generation to Actuate iServer. The report user typically initiates the request by clicking a button in a form.

Designing a browser menu control

The example library's browser scripting control named DHTMLMenuControl generates a menu in a web browser. DHTMLMenuControl defines several new variables and overrides several methods. Figure 26-8 shows the variables DHTMLMenuControl defines.

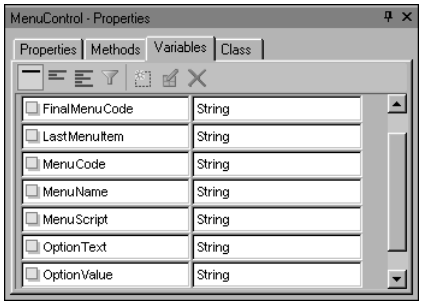


Figure 26-8 DHTMLMenuControl variables

MenuName, MenuScript, OptionText, and OptionValue have a visibility of Commonly Used Property. These variables appear on the Variables and Properties pages, as shown in Figure 26-8 and Figure 26-9.

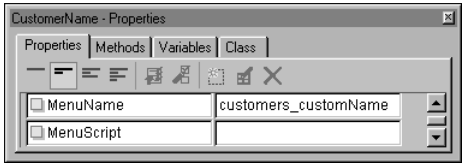


Figure 26-9 DHTMLMenuControl properties

Overriding methods in DHTMLForm

DHTMLMenuControl overrides the following methods:

- BuildFromRow( )
- OnRow( )
- Finish( )
- BrowserCode( )

Using BuildFromRow( )

DHTMLMenuControl overrides BuildFromRow( ) as shown in the following example:

```
Function BuildFromRow( row As AcDataRow ) as AcBuildStatus
   BuildFromRow = Super::BuildFromRow( row )
   BuildFromRow = ContinueBuilding
End Function
```

Using OnRow( )

OnRow( ) defines an Option element for each menu option and assigns the code to MenuCode. OnRow( ) is called repeatedly because BuildFromRow( ) returns ContinueBuilding. DHTMLMenuControl overrides OnRow( ) as shown in the following example:

```
Sub OnRow ( row as AcDataRow )

   Dim NL as String
   Dim TAB as String
   Dim i as Integer

   Dim text as String
   Dim value as String

   NL    =  Chr$(10)
   TAB   =  Chr$(9)
   i     =  0

   text  =  row.GetValue(OptionText)
   value =  row.GetValue(OptionValue)

   MenuCode = MenuCode + NL + " <option value= ' " + value + "
      '>" + text +
      "</option>"

End Sub
```

Using Finish( )

Finish( ) generates a Select element and assigns the code to FinalMenuCode. DHTMLMenuControl overrides Finish( ) as shown in the following example:

```
Sub Finish( )
   Super::Finish( )
   FinalMenuCode = "<SELECT NAME=' " + MenuName + " '>"
   FinalMenuCode = FinalMenuCode + MenuCode + Chr$(10)
   FinalMenuCode = FinalMenuCode + "</SELECT>" + Chr$(10)
   FinalMenuCode = FinalMenuCode + MenuScript
End Sub
```

Using BrowserCode( )

The BrowserCode( ) method assigns FinalMenuCode to the BrowserCode variable. The BrowserCode variable passes to the DHTML converter. DHTMLMenuControl overrides BrowserCode( ) as shown in the following example:

```
Function BrowserCode( ) As String
   BrowserCode = FinalMenuCode
End Function
```

Creating a drop-down menu in a web browser

By subclassing DHTMLMenuControl from a library, you can create a browser scripting control that generates a drop-down menu when a report appears in a web browser. For example, you can create a control called CustomerMenu that generates a drop-down menu listing all customers in a database.

Set the MenuName, OptionText, and OptionValue properties as Figure 26-10 shows. The OnRow( ) method uses values of OptionText and OptionValue to generate an Option element for each menu option. Because the OptionText setting is customers.customName, the customer name appears in the menu. The Finish( ) method uses the value of MenuName to generate a SELECT element.

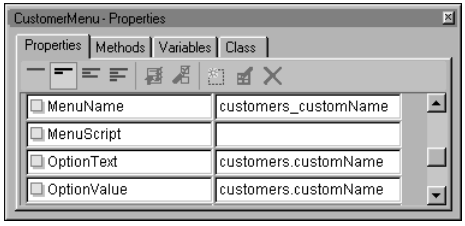


Figure 26-10 Setting DHTMLMenuControl properties

Displaying custom browser code in the view perspective

You can display the custom browser code generated by the CustomerMenu control in the view perspective. Right-click the control and choose View Browser Code. The code appears in Custom Browser Code Editor, as shown in Figure 26-11.

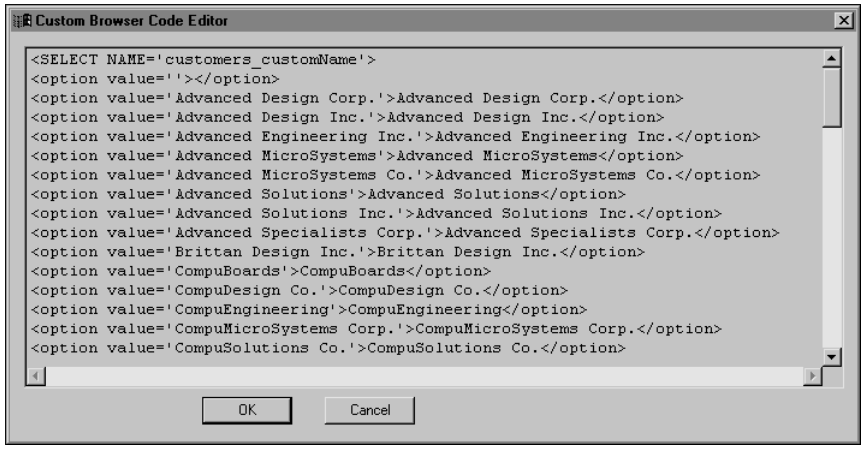


Figure 26-11 Custom Browser Code Editor

The output of the custom browser code is a drop-down menu that lists all customers in a database, as shown in Figure 26-12.

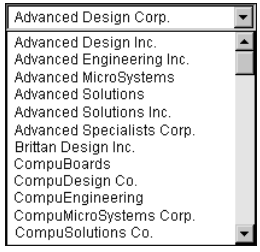


Figure 26-12 Custom browser code output

Chapter

27

Designing for XML output

This chapter contains the following topics:

- About the XML data format
- About the View process
- Publishing an Actuate report as XML

About the XML data format

Extensible Markup Language (XML) is a meta-language that provides a language- and platform-independent format for representing data. You can view data extracted to an XML format using any software that supports XML, including a web browser. XML supports processing data in another application.

XML is a subset of the Standard Generalized Markup Language (SGML), which adds descriptive and structural information to data. XML uses mark-up tags to specify the content, structure, and display of data. For example, XML tags specify whether the data is a parts catalog title or an order amount.

Using Actuate e.Report Designer Professional, you can design a report to conform to a specific XML data format. The XML data format can extract data for an entire Actuate report or for selected elements of the report. A report converted to XML retains its original design and graphics. The XML data format ignores page components such as pages, flows, page headers, and page footers.

XML data supports page-level security. A user's privileges determine what data he views. Page-level security is available only for a report published to Actuate iServer that the user views in a web browser using the DHTML Viewer.

About document type definitions

An application that processes XML conforms to a document type definitions (DTD) specification. A DTD is one or more files that contain the formal definition of a document. A DTD defines specific data formats.

An XML document that adheres to an associated DTD is a valid XML document. An XML document that adheres to the XML standard is a well-formed XML document. If an XML document does not have an associated DTD, it must include a declaration that it is a stand-alone document. Like a valid XML document, a stand-alone XML document must be well-formed according to XML rules.

You determine the XML data that a report produces by setting properties or overriding methods in a report design.

Examining a simple XML document

The following example is a simple XML document defines two lines of text, a greeting and a response:

```
<?xml version="1.0" standalone="yes"?>
<conversation>
   <greeting>Hello, sales staff!</greeting>
   <response>Thank you for the customer information!</response>
</conversation>
```

This document consists of the following contents:

- A prolog. The prolog identifies which version of XML the document supports and whether the document has an associated DTD. In this example, the prolog is the following line:

  ```
  <?xml version="1.0" standalone="yes"?>
  ```

  This prolog specifies that the document uses XML version 1.0. The standalone="yes" element indicates that this document does not have an associated DTD.

  Actuate software generates the prolog for an XML report. You can generate a custom prolog by overriding the XMLDataProlog( ) method.

- A conversation element. This user-defined element contains a greeting element and a response element. The tag <conversation> begins the conversation element. The tag </conversation> ends the conversation element.

The following fragment shows a more complicated XML document that points to the DTD to which the document adheres, and creates a report heading:

```
<?xml version="1.0" standalone="no" encoding="UTF-8"?>
   <!DOCTYPE titlepage QTRLYRPT "http://www.MyCorp.com/RptDTDs
      /qtrly.dtd"
   [<!ENTITY % active.links "INCLUDE">]>
   <titlepage>
      <white-space type="vertical" amount="36"/>
      <title font="Palatino" size="30"
         alignment="centered">MyCorp Quarterly Report</title>
      <white-space type="vertical" amount="12"/>
      <image location="http://www.MyCorp.com/RptDTDs/imgs
         /logo.img" type="URL" alignment="centered"/>
      <white-space type="vertical" amount="24"/>
      <author font="Palatino" size="18/22" style="italic">MyCorp
         Financials Group</author>
   </titlepage>
```

This document uses XML version 1.0 and UTF-8 encoding. The document adheres to the quarterly report DTD, qtrly.dtd, located at http://www.MyCorp.com/RptDTDs. The title is 30-point Palatino font with the text MyCorp Quarterly Report, followed by the company logo. The MyCorp Financials Group authors the document.

About the View process

The View process supports viewing an Actuate report in a variety of formats. The View process converts Actuate report pages and other report data from an internal XML format into DHTML for viewing or creates a PDF for printing. This

conversion supports viewing and searching an Actuate report using a web browser. The View process can also send the XML data to another application for processing without first converting it to another format.

The View process also supports the following functions:

- Searching report data on the web. The View process can extract search results in a variety of formats, including Excel and comma- or tab-separated formats.
- Creating a dynamic table of contents (TOC) for navigating a report on the web.
- Page-level security. Page-level security restricts access to data that the user does not have permission to see.

Figure 27-1 shows the View process architecture.

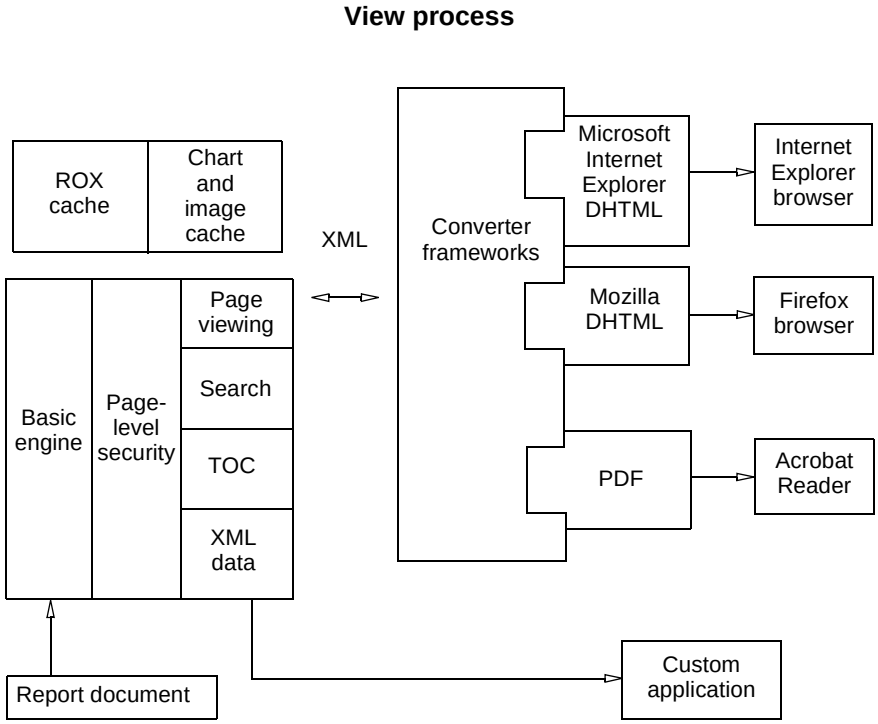


Figure 27-1 View process architecture

Publishing an Actuate report as XML

To design an Actuate report to produce XML data output, you set XML properties for components and override methods to achieve custom XML data extraction. You can extract the data for an entire report to produce XML output or extract

only the data for a specific component. The XML properties for a component determine whether Actuate software produces XML data for that component.

Actuate does not provide a document type definition (DTD) for XML data. The format depends upon the report requirements. The XML output can conform to an existing DTD or you can create a DTD.

How to produce XML output

1. In e.Report Designer Professional, set XML properties for all report components that you want to view in the XML output.
2. Build and run the report.
3. In the view perspective, choose File➤Save As➤XML Data. Specify a directory in which to save the file. An XML file appears in the directory.
4. Double-click the file name to view the file in a web browser, as shown in Figure 27-2.

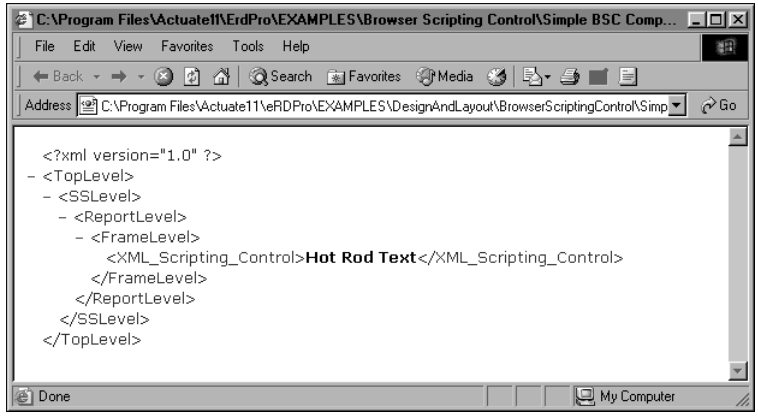


Figure 27-2 XML file

Mapping XML to an Actuate report component

The Actuate Foundation Classes (AFC) include XML-specific properties. These properties, which you set when you design the report, specify how to generate XML data.

When designing a report for XML, consider the following guidelines:

- Know your DTD. The DTD determines how to represent your report's information.
- Identify the report's consumer. If another application, such as a catalog or an Excel spreadsheet, processes the report's XML output, the XML design can differ from that of a report designed for an end user.

Use the following guidelines to convert an Actuate report to XML. Check your DTD for specific requirements for your XML documents.

- The AcReport component is the top-level component of the report. In the component, set the properties that determine the data that appears in the report. These properties are the XML prolog information.
- A section or frame typically maps to an XML element. Set the component's XMLTag property to the element name. Set the XMLType property to XMLElement.
- To use a control to represent an attribute, set the control's XMLType property to XMLAttribute.
- If your DTD's requirements do not readily map to an Actuate report's XML, set the XMLType property to XMLCustom and override the GenerateXML( ) method of the component to generate the required XML.

About XML properties

All AFC report component classes include XML properties and methods to control the generation of XML data. This section describes the classes that provide properties and methods and the AFC methods that support generating XML data.

The property group XML Data in Properties lists the XML properties that e.Report Designer Professional supports for a component. If XML properties do not apply to a specific component, the XML Data property group does not appear in the Properties page for that component. Figure 27-3 shows the XML Data property group for the top-level report component.

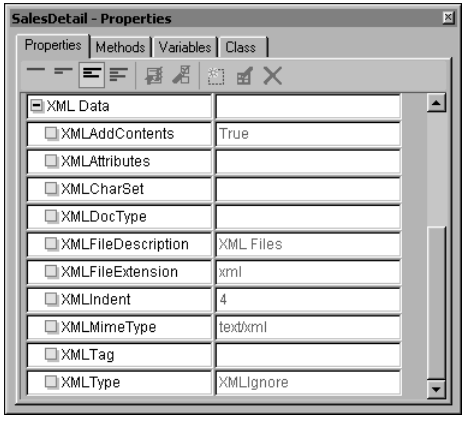


Figure 27-3 XML Data property group

XML properties are available for AcReportComponent and its subclasses, except for subclasses of AcBasePage.

XML properties apply to the following components:

- AcReport
- AcSection and its subclasses
- AcFrame
- AcControl and its subclasses

About AcReport XML properties

Table 27-1 describes the XML properties for AcReport. These properties control the overall structure and appearance of the XML output file.

Table 27-1 XML properties

Property	Description
XMLCharSet	The encoding declaration to insert in the XML prolog. If not specified, Actuate software does not include an encoding declaration in the XML prolog.
XMLDocType	The document type string to display in the document's DOCTYPE declaration.
XMLFileDescription	The description of the XML file to build. The default value is XML Files.
XMLFileExtension	The file extension of the XML file to build. The default value is xml.
XMLIndent	The number of spaces to indent each level in the XML file. Set the value to 0 to improve the performance of XML generation and reduce the XML file size. The default value is 4. Use the default value when viewing and debugging the XML report.
XMLMimeType	The MIME type for the XML document. The default value is text/xml.

About AcReportComponent XML properties

Table 27-2 describes the XML properties for AcReportComponent.

Table 27-2 XML properties for AcReportComponent

Property	Description
XMLAddContents	Determines whether to search through a component's contents recursively to find other components for which to generate XML. The default value is True. Set this value to False to skip over sections or frames for which you do not want to generate XML data.
XMLAttributes	A set of additional attribute values to add to the current XML element. Use XMLAttributes to add fixed attributes. Use controls to add attributes that vary depending on the data in the data row.
XMLTag	The XML element or attribute name for this component.
XMLType	The type of XML object the component represents. The values are ■ XMLAttribute. The component is an XML attribute. ■ XMLCustom. A custom XML element. AFC calls the GenerateXML( ) method to generate the custom element. ■ XMLElement. The component is an XML element. ■ XMLEmptyAttribute. The component is an empty XML element. ■ XMLIgnore. Do not generate XML for the component. XMLIgnore is the default value for XMLType.

About AFC XML methods

Table 27-3 describes the AFC methods that control generation of XML from an Actuate report.

Table 27-3 AFC methods

AFC Class	Method	Description
AcReport Component	Function GetXMLText( ) As String	Returns the value for an element or attribute. The default value for a data control is the value of GetText( ) formatted for XML. The default value for all other components is an empty string. Override this method to provide custom formatting.

Table 27-3 AFC methods

AFC Class	Method	Description
AcReport Component	SubGenerateXML(visitor AsAcXMLData Visitor)	Generates the XML for a custom component. The framework calls this method if the component's XMLType is XMLCustom. Override this method to produce custom XML. The default behavior for this method is to do nothing.
AcReport	Function GenerateXML DataFile (fileName As String) As Boolean	Generates an XML file for the report.
AcReport	Sub XMLDataProlog (channel As Integer)	Generates the prolog portion of the XML report then writes the prolog to a channel. Override XMLDataProlog( ) to create a custom prolog. This method generates a standard prolog. Actuate software generates the XML version and encoding attributes in the prolog. The XML version number is the version of XML that Actuate software supports. The encoding is the type of encoding used to build the report. The default encoding is UTF-8.

The following report, the Design Emporium product catalog, illustrates how to map a report component to an XML element as shown in Figure 27-4.

Each component in the report corresponds to one of the following XML elements:

- The Catalog element contains the entire catalog.
- The Address element contains addresses. Its attribute, Type, specifies the address type, such as Mailing.
- The Product element contains information about the items in the catalog. Its attribute, Category, specifies the item type, such as Accessories. Product contains sub-elements that describe each catalog item in detail.
- The Part element is a sub-element of Product. Part contains the item's catalog number as an attribute.

- The Description element is a sub-element of Part. Description contains a text description of the item.
- The Price element is a sub-element of Part. Price contains the item's list price.

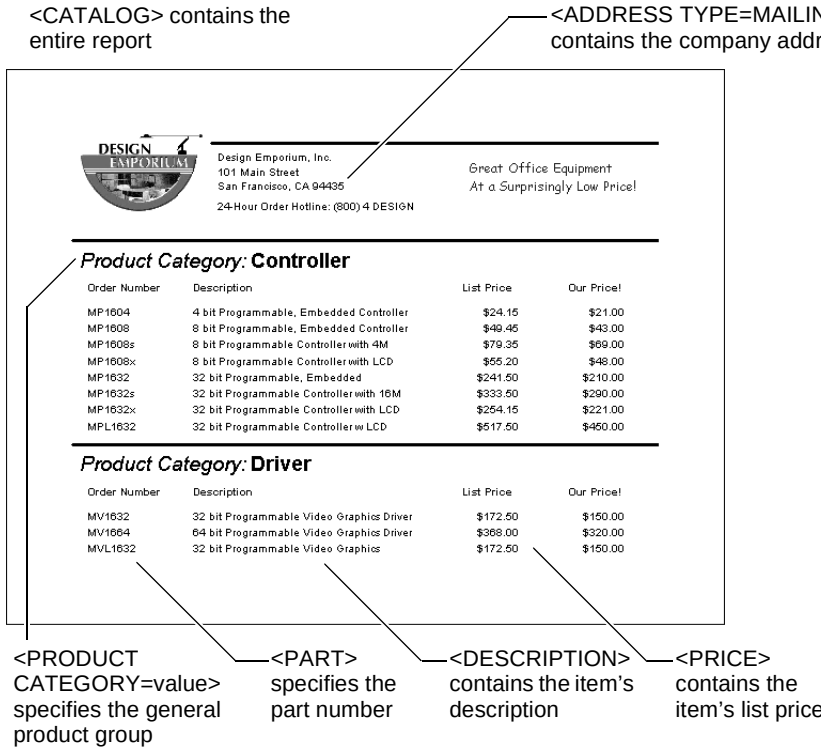


Figure 27-4 Mapping report components to XML elements

The report design defines the Category element using the values shown in Figure 27-5.

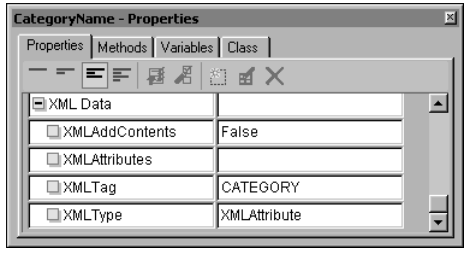


Figure 27-5 Category element

The report design defines the Address element using the values shown in Figure 27-6.

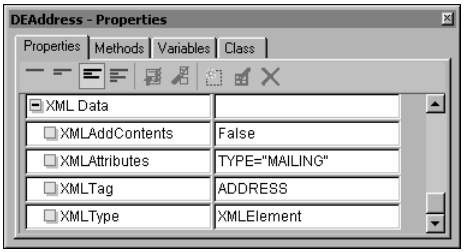


Figure 27-6 Address element

The report design defines the List Price element using the values shown in Figure 27-7.

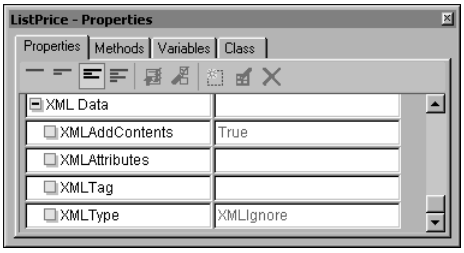


Figure 27-7 List Price element

The XML output for this report is similar to the following code example:

```
<?xml version="1.0" encoding="UTF-8"?>
<CATALOG>
   <ADDRESS TYPE=MAILING>Design Emporium, Inc.&lf;101 Main
      Street&lf;San Francisco, CA 94435</ADDRESS>
   <PRODUCT CATEGORY="Accessories">
   <PART PARTNO="MR0410">
      <DESCRIPTION>Pencil Cup</DESCRIPTION>
      <PRICE>$4.00</PRICE>
   </PART>
   <PART PARTNO="MR0440">
   …
</CATALOG>
```

Part Six

Previewing reports in Actuate e.Report Designer Professional

Chapter

28

Running a report

This chapter contains the following topics:

- About running a report
- Running a report that uses parameters

About running a report

You run a report when you want Actuate e.Report Designer Professional to generate a report object instance (.roi) file using the most current data. A report object executable (.rox) file contains compiled code that specifies how Actuate e.Report Designer Professional generates a report and what data it retrieves for the report. When you run an ROX that uses parameters, you can use the default parameter settings or modify those settings to filter report data.

A report developer can run a report locally to view and test the report design. Report users typically run the report on the web, from either Actuate iServer or another application server. This chapter explains how to run a report locally. For information about running a report from Actuate iServer, see *Using Information Console.*

Running a report locally

When you run a report locally, the report generates immediately and appears in the view perspective.

How to run a report executable file locally

1 Choose File➛Open.

2 In Select File, in Files of type, choose Report Executable (*.rox), as shown in Figure 28-1.

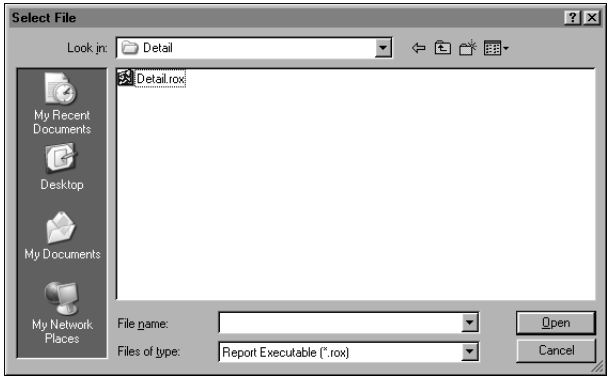


Figure 28-1 Select File

3 Select the ROX you want to run. Choose Open.

The Factory window appears, as shown in Figure 28-2. The Factory window displays the status of the generation process and the number of the page being generated.

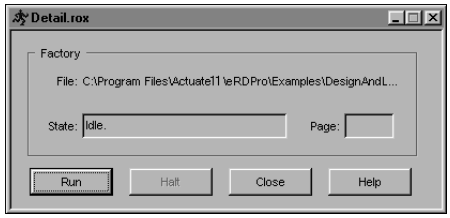


Figure 28-2 Choosing Run to generate a report

4 Choose Run to start the report generation process.

5 In Requester, you can choose OK to accept the default parameter values, or provide parameter values to narrow the range of data the report displays.

To stop report generation at any time, choose Halt.

6 Choose Close after the report finishes running.

Viewing pages during report generation

Typically, when you run a report executable file, the report appears in the view perspective as soon as the first page generates. There is one exception. Requester displays the Bundle Rox in Roi output parameter. If you set this parameter to True, the view perspective displays the report only when the report is complete.

As the report generates, you can use all the view perspective features except the Table of Contents. The Table of Contents is unavailable because it relies on all the data in a completed report. Figure 28-3 shows the Factory window as a report generates in Actuate e.Report Designer Professional.

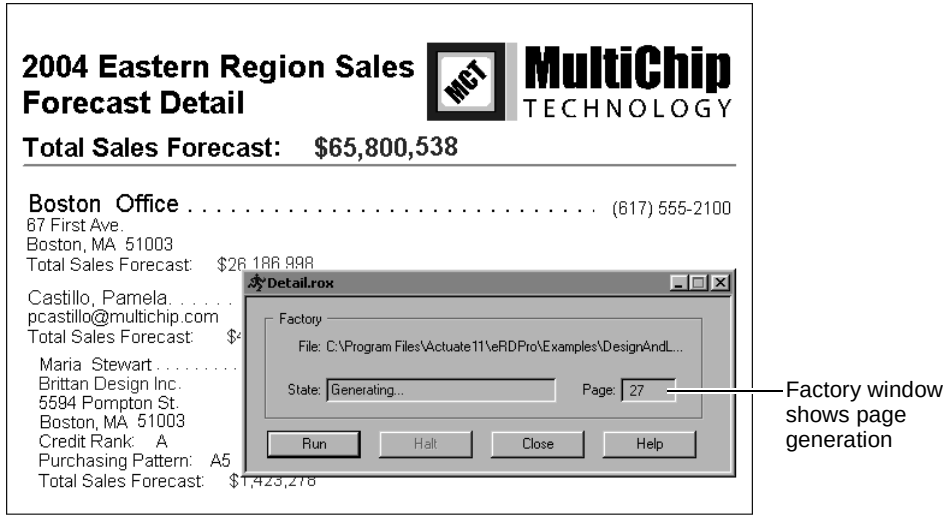


Figure 28-3 The view perspective as a report generates

Understanding checkpoints

As the Factory generates the pages of the report, it writes the completed pages to the report at designated checkpoints. The checkpoints occur after the completion of the 1st, 5th, 10th, 25th, and 50th pages, and every 50 pages after the 50th.

The view perspective displays the portions of the report that were written to a file when you manually refresh the report.

Refreshing the report in the view perspective

To refresh a report in the view perspective as the report runs, choose any of the paging commands—First Page, Previous Page, Next Page, or Last Page—from the toolbar or the View menu.

Sometimes when you choose a paging command, the report in the view perspective does not immediately refresh. The delay occurs when the report has not reached a new checkpoint since the last refresh. For example, if the last page of the displayed report is 50, the report does not refresh until the 100th page has been processed.

Use the Page information in the Factory window to refresh the report. For optimal results, choose a paging command immediately after a checkpoint.

Running a report that uses parameters

A report developer designs a parameter to appear on Requester so that you can provide parameter values conveniently and effectively.

In some cases, parameter values can be displayed in a drop-down list or as a series of radio buttons. In other cases, a text box is the best way to capture your input.

Specifying parameter values on Requester

In Requester, you can set parameter values using the following tools:

- A text box
- A drop-down list
- A set of radio buttons
- A check box

Figure 28-4 shows the various ways to enter parameter values on Requester.

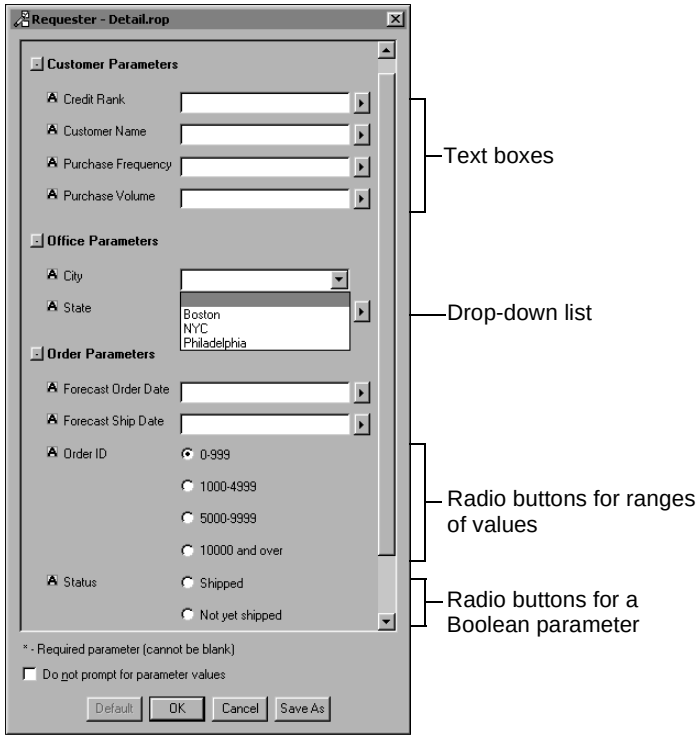


Figure 28-4 Providing parameter values in Requester

Using a text box to set a parameter value

If the parameter uses a text box on Requester, what you type in the field beside the parameter name depends on the type of parameter. For a static parameter, you can type a single parameter value. The report displays data based on the value you specify.

For ad hoc parameters, you can type a single value, a list of values, or an expression in the text box. For example, for an ad hoc parameter titled Customer Name, you can type a single customer name or a comma-separated list of customer names. You also can display information about all customers except Brittan Design Inc., for example, by writing the following expression:

```
!Brittan Design Inc.
```

An ad hoc parameter gives you a great deal of flexibility in the data you can request for a report. To help you write an expression, you can use the ad hoc parameter builder, shown in Figure 28-5, to view operators, such as greater than (>) and less than (<).

You can type alphanumeric values in text boxes. Requester does not support using thousands separators for numeric values.

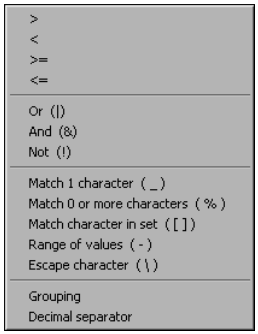


Figure 28-5 Ad hoc parameter builder

Using radio buttons or a drop-down list to set a parameter value

Drop-down lists and radio button series display a static list of values from which you can select one and only one value. In some drop-down lists, however, you can type a custom value in the list, if the report developer set up the parameter to accept custom values.

Setting values for date parameters

You can set values for date parameters in several ways, depending on how the report developer set up the parameter. For a date parameter that uses a text box, you can provide a single date as the parameter value by typing the date in the text box. If Ellipsis appears beside the text field, you can choose Ellipsis to open the Select Date and Time dialog box, shown in Figure 28-6.

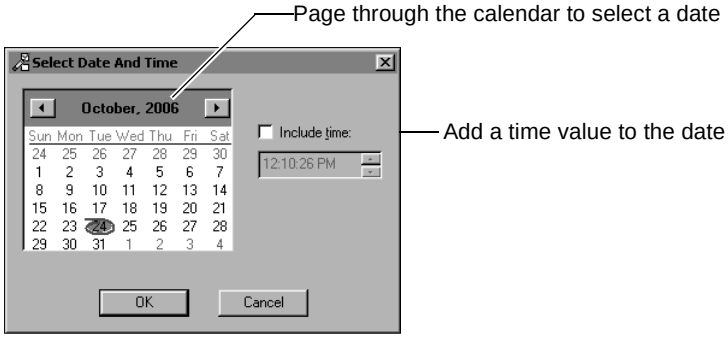


Figure 28-6 Select Date and Time

Using Select Date and Time, you can page forward and back through the calendar pages to select a date. You also can include a time value with the date, if you know the data consists of both a date and a time. Select Date and Time is available only if the report developer set up the parameter to display a date picker.

Using null values with structure and table parameters

Actuate e.Report Designer Professional assigns the keyword Null as the default value for a structure field or table column. This keyword implies that a table or structure parameter can contain a row of null values for all its columns. If you specify the keyword Null as a parameter value, Actuate e.Report Designer Professional treats the keyword as a null input value for that field. In order to specify the literal string Null as a value for a table or structure parameter, enclose it within single quotation marks, as shown in the following example:

```
'Null'
```

Specifying a value for a structure parameter

A structure parameter consists of a number of descendant elements that have their own data types. These elements appear as a group of parameters on Requester. Expand the structure parameter to see the descendant elements.

Use one of the following methods to supply values for a structure parameter:

- If the report supports text boxes, type a value next to the parameter name.
- If the report supports drop-down lists, select one of the options available in the drop-down list.
- If the report supports typing a custom value in the list, type a custom value.
- If the report uses date parameters, type a date or use the date picker to specify a value.

Setting a value for a table parameter

Unlike other report parameters, you cannot specify a value for a table parameter on Requester. Instead, you use Table Parameter Editor to provide values. To display Table Parameter Editor, choose Table Editor beside a table parameter on Requester. Table Parameter Editor appears, as shown in Figure 28-7.

A table parameter contains one or more columns. In Table Parameter Editor, you can provide values for table parameters. You also can add rows to a parameter, delete rows, and edit the values of existing rows.

How to add a row

In Add row, type a value for the row. The value you type must coincide with the data type of the column. Choose Add. The new row appears under Table rows, as Figure 28-8 shows.

To save your changes, choose OK. To return to Requester without saving your changes, choose Cancel.

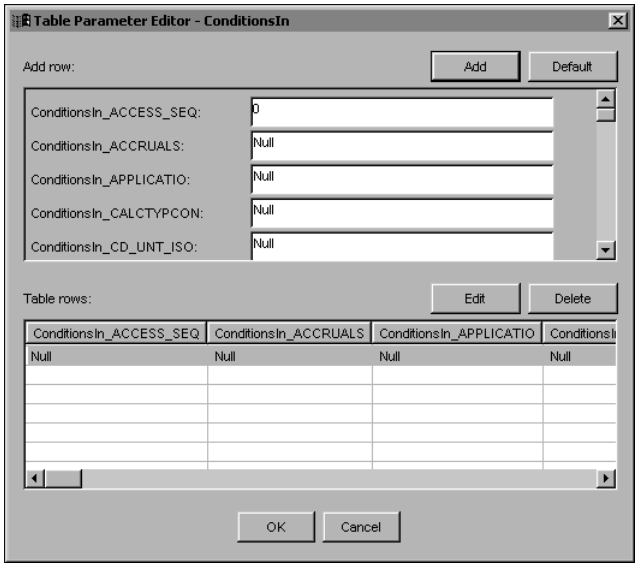


Figure 28-7 Table Parameter Editor

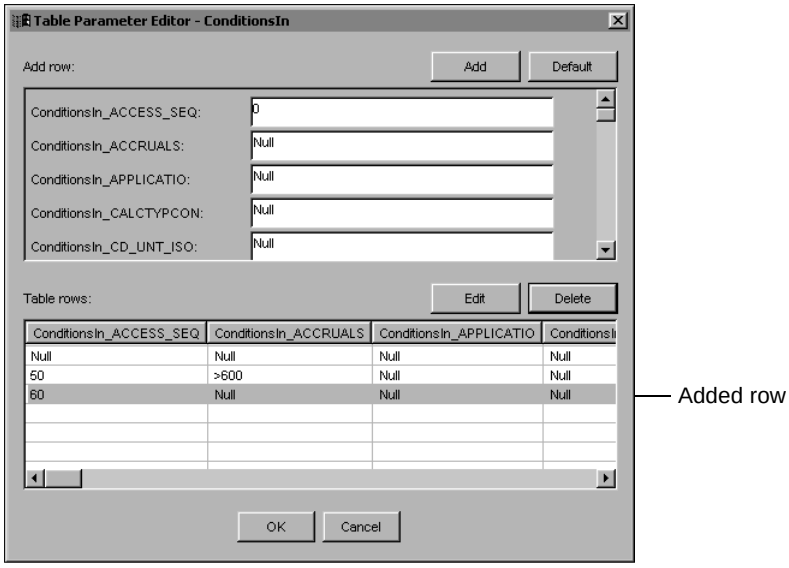


Figure 28-8 Adding a row in Table Parameter Editor

How to edit a row

You can edit the value of each row in a table.

1 In Table rows, select the row you want to edit and choose Edit.

2 In Edit row, specify a value for the row you want to modify. Choose Update. The modified row appears in Table rows.

3 To return to Requester, choose OK.

How to delete a row

To delete a single row in Table Parameter Editor, select the row you want to delete. Choose Delete.

To delete multiple rows, press Ctrl and select the rows to delete. Choose Delete.

Chapter

29

Viewing a report using the view perspective

This chapter contains the following topics:

- Opening and viewing a report
- Navigating a report
- Examining objects and classes
- Personalizing the view perspective
- Copying report data to another document
- Copying report data to another document

Opening and viewing a report

The view perspective displays an Actuate Basic report object instance (.roi) file that is stored in a file system, such as a stand-alone machine or a network file server. In the view perspective, you can test your report design, save, print, or navigate the report, and perform other tasks with the generated report.

In the view perspective, you can:

- Navigate the report, using the table of contents, paging tools, or sequential paging.
- Activate hyperlinks to data in the current report or another report.
- Use context menus to view online help and perform other actions.
- Examine objects and classes.
- Personalize the view perspective.
  - Split the screen to view two parts of the report at once.
  - Change the scale of the report by zooming in or out.
- Copy report data to a text file.

To help you get started, e.Report Designer Professional includes sample reports. We use Detail.roi as an example in this chapter. You open the report object design (.rod) file, Detail.rod, then generate a report object instance (.roi) file, Detail.roi, to view the report and work in the view perspective. With this sample report, you can explore many of the viewing capabilities.

How to run and view a report

1. In the e.Report Designer Professional design perspective, choose File➛Open.
2. In Open, navigate to and open the following file:

   `<eRDPro_HOME>\Examples\DesignAndLayout\Detail\Detail.rod`
3. Choose Report➛Build and Run.
4. In Requester, choose OK. e.Report Designer Professional generates the report, Detail.roi, and displays it in the view perspective.

   Figure 29-1 shows the first page of Detail.roi in e.Report Designer Professional.

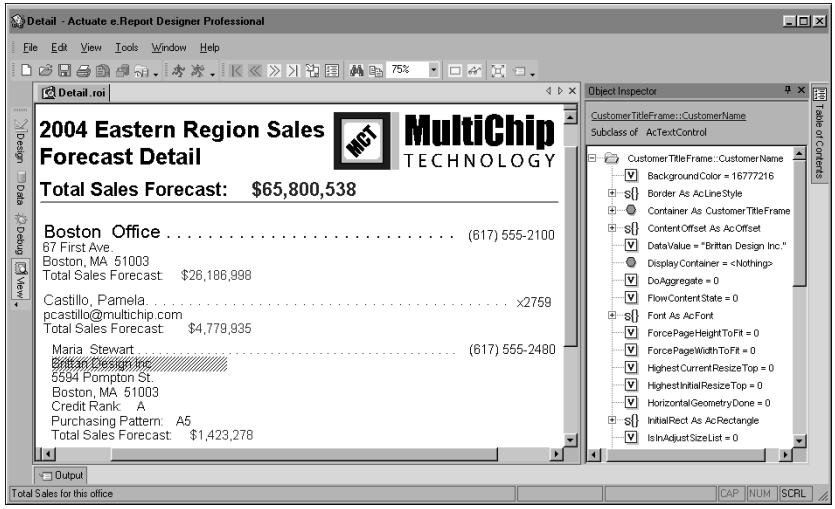


Figure 29-1 The view perspective showing the object inspector window

Navigating a report

There are a number of ways to navigate an Actuate Basic report. You can:

- Use the paging tools to page through the report sequentially or go to the first or last page.
- Go to a specific page by specifying a page number.
- Use the report's table of contents to view the content hierarchy.
- Use hyperlinks, if the report design includes hyperlinks, to go from one part of the report to another part in the same report or a different report.

Using the paging commands

Both the toolbar and the View menu include the following paging commands:

- First page
- Next page
- Previous page
- Last page

Because Actuate Basic reports store data in a highly compressed form, these commands make it possible to move quickly through large reports. Paging sequentially through the report is an option but often it is faster to choose specific

paging commands. Use either toolbar buttons, shown in Figure 29-2, or menu items to access the paging commands.

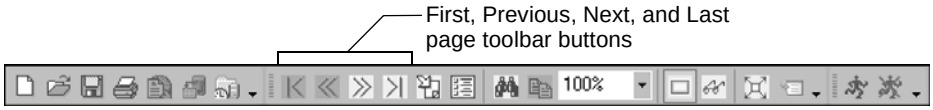


Figure 29-2 The toolbar's paging commands

Going to a specific page

If you are familiar with the page layout of the report or you are working from a printout to find particular report pages, choose the Go To Page button, shown in Figure 29-3, to access a specific page by number. You also can choose the View➛Go To menu command.

Figure 29-3 Go To Page toolbar button

Go to Page displays a dialog box, shown in Figure 29-4, that supports typing the number of the page to display.

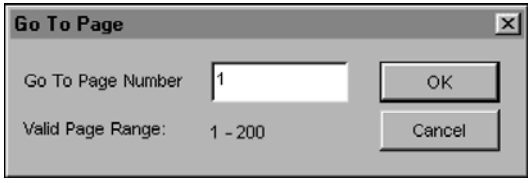


Figure 29-4 Go To Page

Using the table of contents

Reports typically contain a generated table of contents. You can view the table of contents with sections collapsed or expanded depending upon the level of detail you want to see. Figure 29-5 shows the table of contents, partially expanded, for the sample report Detail.roi.

In this example, the Eastern Region Sales Forecast includes three offices. Each office includes a group of sales representatives and each of those representatives has a group of customers. Each customer, in turn, has one or more orders.

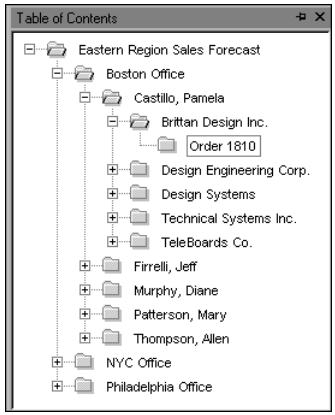


Figure 29-5 Table of contents

How to access and use the table of contents

1 In the view perspective, choose Table of Contents from the toolbar.

The table of contents for the report appears. Figure 29-6 shows the table of contents for the sample report, Detail.roi.

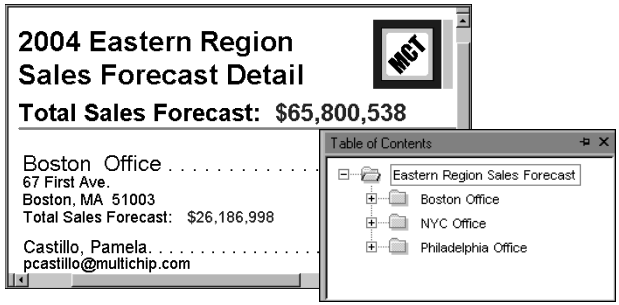


Figure 29-6 Table of contents for Detail.roi

2 Expand an entry by choosing + beside the entry. Figure 29-7 shows the Boston Office folder expanded to display sales representatives in that office.

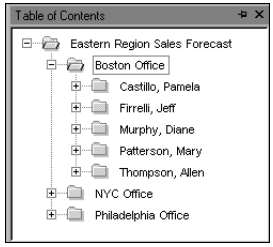


Figure 29-7 Expanded table of contents

To collapse the table of contents to show fewer levels of information, choose - beside the entry.

3 Double-click an item in the table of contents. The view perspective displays the selected item, as shown in Figure 29-8.

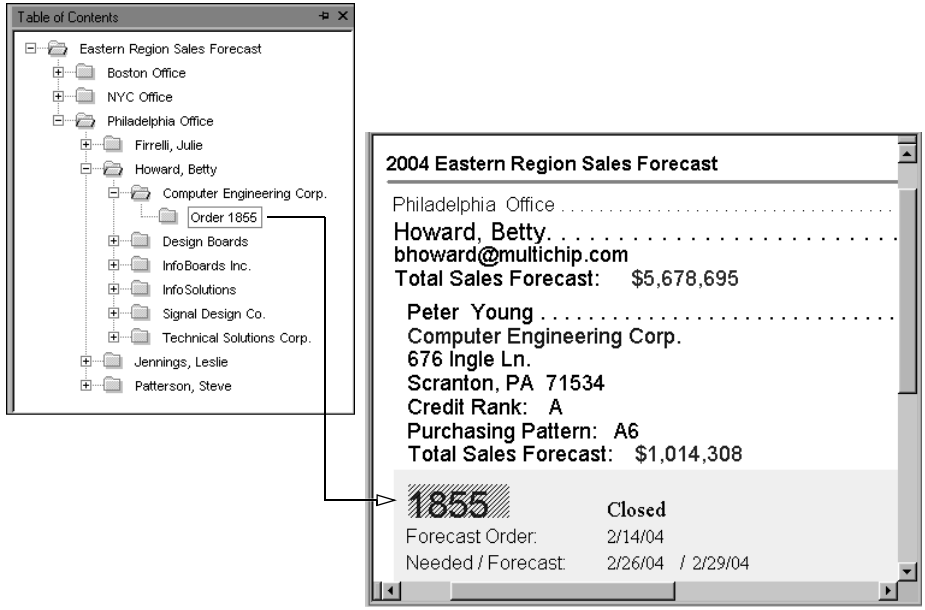


Figure 29-8 Displaying data using the table of contents

Using hyperlinks to navigate report data

During the design process, the report developer can provide hyperlinks in an Actuate Basic report. Hyperlinks make it easy to move from information in one part of a report to related information in another part of the same report or in a different report.

The report developer can create a hyperlink on an item of report data, such as an order total or office name, or on a graphic tool, such as a button. For example, a More button on a summary page can link to a detailed section of the report. In the detailed section, the developer can insert More and Back buttons to facilitate navigating through the pages of that part of the report. Figure 29-9 shows a page of Forecast.roi with More and Back buttons.

To see several examples of hyperlinks, open Forecast.roi and work with the hyperlinks on pages 1 and 2 of that report.

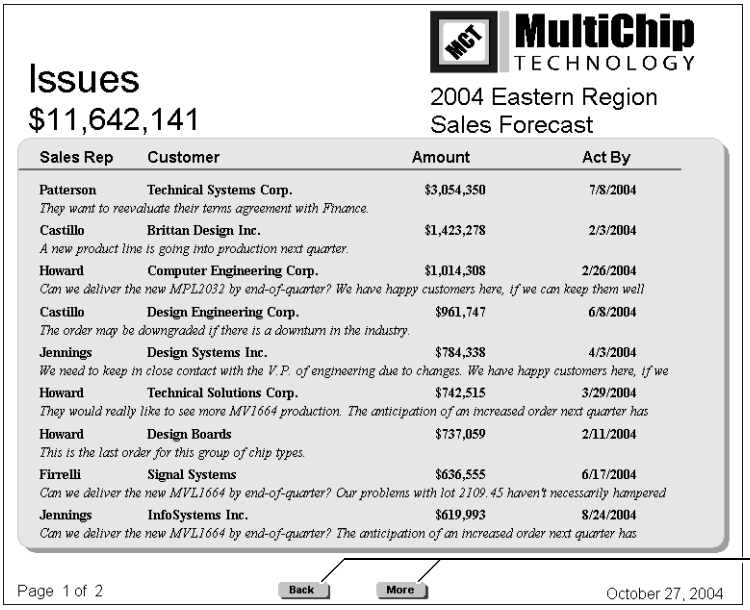


Figure 29-9 Navigating a report using hyperlinks

How to use a hyperlink

1. Hover the mouse pointer over a field in the report instance. If the field contains a hyperlink, the pointer changes to a small hand.
2. Double-click the hyperlinked field to go to the related field. Figure 29-10 shows the field to which the hyperlink connects.

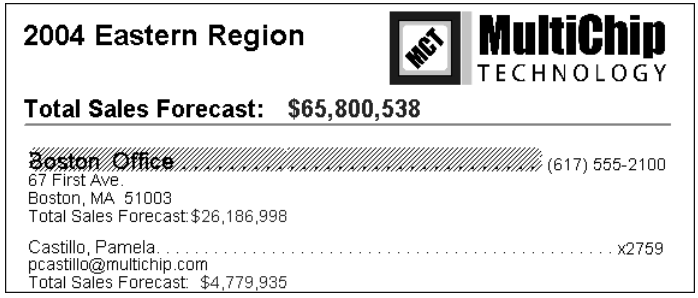


Figure 29-10 Field related to hyperlink

Examining objects and classes

Typically, when you develop a report design, you view the report many times during this process. At times in the development of the design, you notice errors in the report. e.Report Designer Professional provides object highlighting and the object inspector to support the examination of report objects during view time. This functionality is useful in debugging a report.

Highlighting occurs when the mouse pointer hovers over any report component. A red line is drawn around the object, as seen in Figure 29-11. Object highlighting is enabled by default. Selecting the Highlight Objects toolbar button disables it.

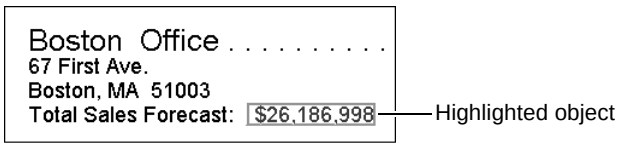


Figure 29-11 Highlighted report object

After selecting an object in the view window, the object inspector displays details about the selected item. This information includes the class and super class names of the component. Both of these items are hyperlinked to support navigation to the implementations of the classes. If the superclass for the object is part of AFC, then its name is not an active link. When you select one of these links e.Report Designer Professional returns to the design perspective. The appropriate component is selected in the report structure.

The object inspector is turned on by default. If it is closed, open it by choosing View➛Object Inspector.

Objects in the object inspector appear in a tree view. This view displays class members and their values in an alphabetical list, as shown in Figure 29-12. This window provides similar functionality to the variables window in the debug perspective.

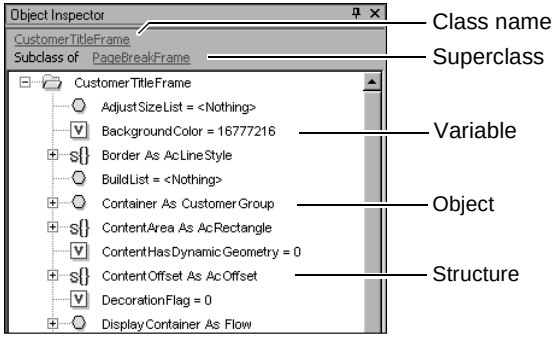


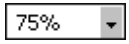
Figure 29-12 The object inspector

The object inspector also provides a way to jump to other objects' definitions in the design perspective. Right-click on a report object, such as a control's container object, and choose Go to Definition from the menu. e.Report Designer Professional returns to the design perspective, with the component representing the object selected.

Personalizing the view perspective

To personalize the view perspective, you can change or rearrange the set of windows that appear in the work area. For example, you can display windows side by side or one above the other. Actuate e.Report Designer Professional saves the current perspective when you exit the program. You can, at any time, restore a window layout to its default state.

You also can magnify the report in the view perspective by using one of the zooming tools. You can either choose View➤Zoom and choose the scale of the magnification, or you can use the Zoom tool in the toolbar.



Using screen arrangement options

The Window menu provides options that support displaying multiple windows. The windows can contain different reports or different views of a single report. As Figure 29-13 shows, you can display multiple windows side by side or one above the other.

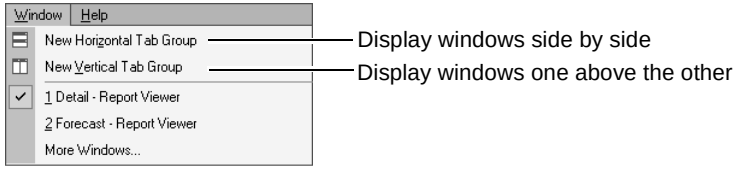


Figure 29-13 Window menu

The Window➤New Horizontal Tab Group command supports displaying report windows or different views of a single report one above another, as shown in Figure 29-14.

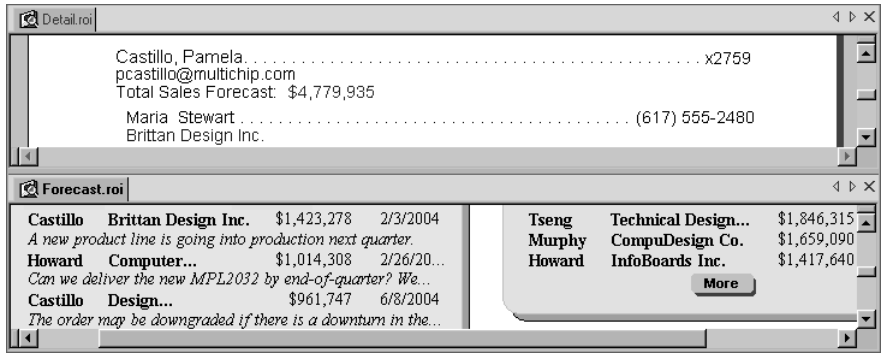


Figure 29-14 Displaying report windows one above another

The Window➛New Vertical Tab Group command supports displaying multiple report windows or different views of a single report side by side, as shown in Figure 29-15.

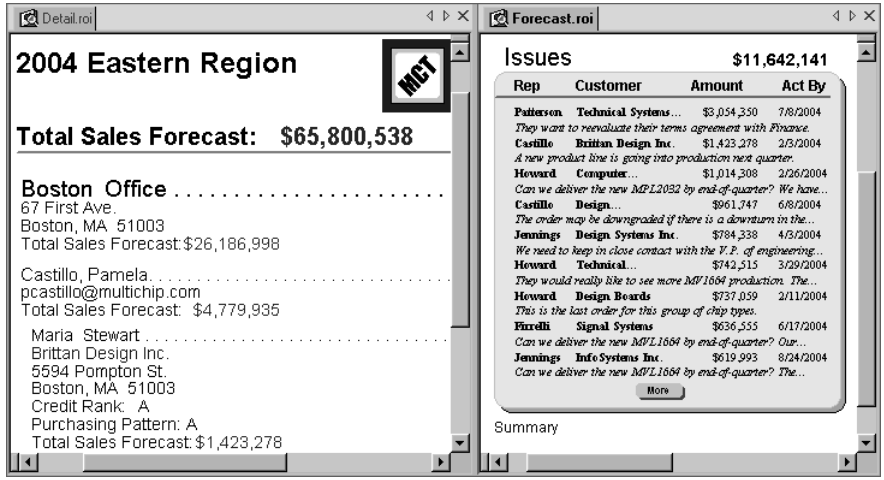


Figure 29-15 Displaying report windows side by side

Also, you can view different parts of the report simultaneously by splitting a window. Use the split box at the top of the vertical scroll bar or the left of the horizontal scroll bar to create a split window. To restore the non-split window, double-click the split bar, as shown in Figure 29-16.

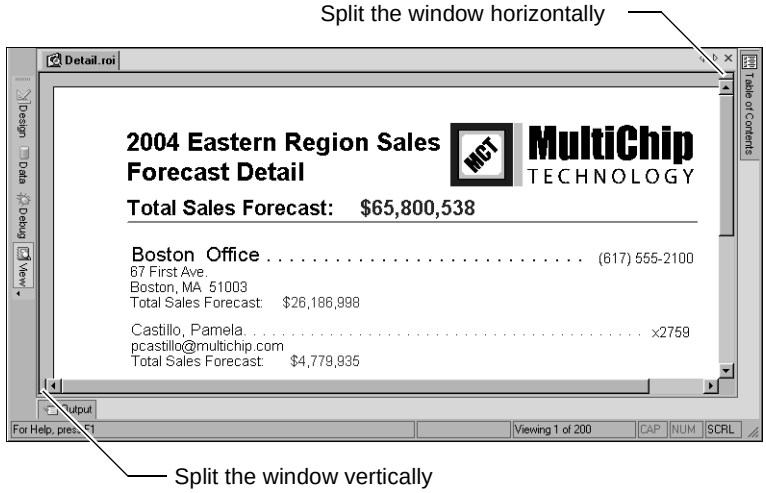


Figure 29-16 Horizontally and vertically splitting the view perspective

About zooming

Zooming in and out of a report makes it easier to view specific data. The scaling range is from 25 percent to 400 percent. Actuate e.Report Designer Professional uses the last zoom setting you set as the default magnification when opening a report. When you change the zoom scale of a report, you change the default zoom setting. When you choose a hyperlink in a report that links to another report, the linked report uses the zoom setting of the report containing the hyperlink.

Figure 29-17 and Figure 29-18 show a report at 50 percent and at 200 percent.

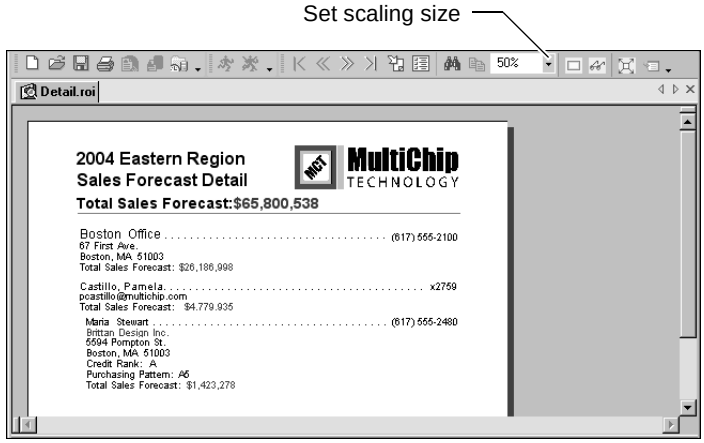


Figure 29-17 Viewing report at 50 percent

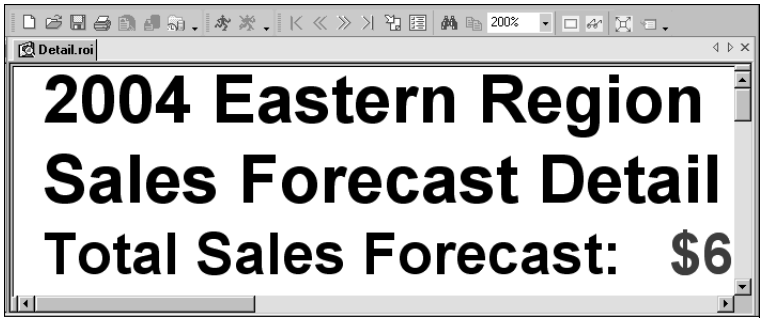


Figure 29-18 Viewing report at 200 percent

Copying report data to another document

You can copy data in the form of numbers, images, or text from an Actuate report document and paste the data into another document. As with the standard Windows copy-and-paste operation, the report data you copy is stored in the Clipboard. You can then paste the data into any other document.

First, select the data you want to copy. To select data, use one of the following techniques:

- Click a single data item, such as a customer name or an order total.
- Press the Shift key and use the left mouse button to select multiple items.
- Drag the mouse pointer to enclose multiple items.

How to copy report data

1 Select text or images within a report.

Figure 29-19 shows a selected area in page 2 of Detail.roi.

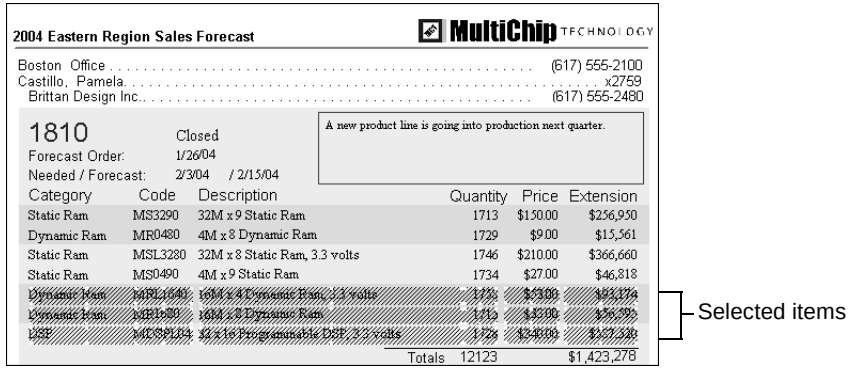


Figure 29-19 Selected area of report

2 Choose Copy from the toolbar.

3 Open the destination document in an external application, such as Microsoft Word or WordPad.

4 Paste the selection into the destination document. Figure 29-20 shows the report data pasted into a WordPad document.

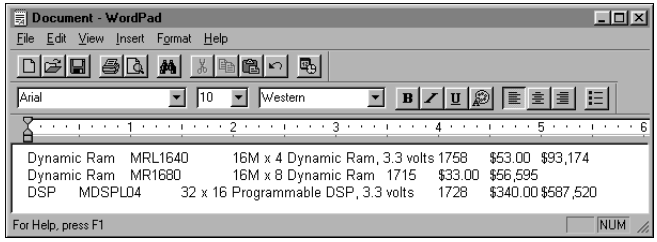


Figure 29-20 Report data in WordPad document

You can then format and organize the items in the destination document.

Chapter

30

Searching for report data

This chapter contains the following topics:

- About the search feature in Actuate reports
- Searching for report data
- Using a search expression in the search conditions
- Using search results to navigate through the report
- Specifying data to display in Results
- Creating a search definition

About the search feature in Actuate reports

Using the search feature you can

- Find data that matches specified criteria.
- Display some or all of the search results.
- Save the search results to a file.

If you run a report object executable (.rox) file for immediate viewing and the resulting report document is long, pages appear as Actuate iServer generates them. Because searching requires complete data, you can search a report only when it finishes loading in the report window.

To search report data, you use the Search window shown in Figure 30-1.

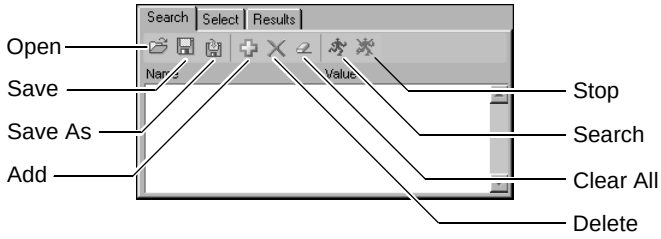


Figure 30-1 The Search window

About searchable report fields

Unlike searching capabilities in word processors or text editors, where you search on a single text value, the search feature looks for user-specified values in one or more report fields. For example, you can find all customers that begin with M by searching on a customer name field. A more complex multifield search could, for example, find all customers in a particular region with purchases over a certain amount. In this example, you would search for certain values in a customer name field, a region field, and a purchase amount field.

The report fields you can search depend on which fields the report developer has designated as searchable.

Identifying a searchable report field

You begin a search by selecting a searchable object in a report. Then, you add the selected object to the Search dialog and specify the value to find.

How to find searchable objects

As you move the cursor around the report, you notice the cursor shape changes to an arrow with an active symbol at the tip. The change in cursor shape indicates that an object is searchable.

Figure 30-2 shows a searchable object in the sample report Detail.roi.

Figure 30-2 Searchable object in report

Searching for report data

To search for specific report data, you select one or more report fields to search, and specify the value to search for each selected field. Searching on one field is straightforward. When searching on multiple fields, however, it helps to understand the effect of selecting multiple fields for a search. Sometimes, it also helps to understand the structure of the report.

Understanding how a multifield search works

In a multifield search, each field that you add to the search conditions adds an And logical operation. For example, if you select three fields and specify values for each, the search conditions, in effect, are condition1 And condition2 And condition3. There is a match only if all conditions are true.

The fields you select for a search must relate to each other. This is where it helps to understand the structure of a report. As you add each field to the search list, the search feature analyzes the relationships among the fields. If a field does not match a supported relationship, the search feature displays an error message. The search feature supports one-to-one and one-to-many relationships among fields.

Here's another way to look at this relationship concept: For a successful search, you can add any fields that are contained within a selected parent field. For example, if a report lists all sales representatives by region, the parent field is the region, and the child field is the sales representative. In this scenario, you can create a search that finds the Western region (region field = Western) and sales representatives whose last names start with T (sales rep = T*). You can apply this parent-child principle when creating search conditions no matter how many hierarchical levels the report contains.

The best way to see the structure of a report is to view its table of contents, if one exists. To do so, choose Table of Contents from the Viewer toolbar. If the report has a hierarchical structure, you can view the table of contents with sections

collapsed or expanded, depending on the level of detail you want to see, as illustrated in Figure 30-3.

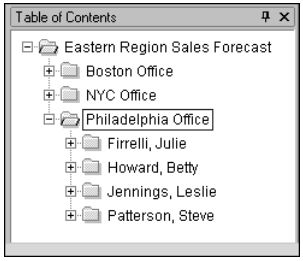


Figure 30-3 Table of contents

Creating search conditions

You create search conditions through the Search dialog. You can modify your search conditions by removing fields or changing field values. You can cancel a search any time before you execute the search process.

How to create search conditions

The following steps assume you already have a report open for viewing.

1 Choose Search in the toolbar. The Search dialog appears, as shown in Figure 30-4.

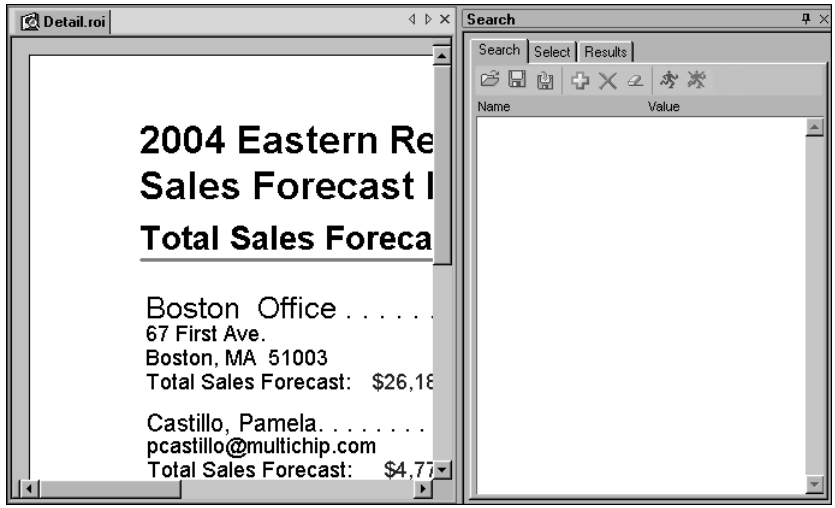


Figure 30-4 Search dialog

2 Use a single mouse-click to select a searchable object. In this example, select a company name, such as Brittan Design Inc., as illustrated in Figure 30-5. The search cursor lets you select only one object at a time.

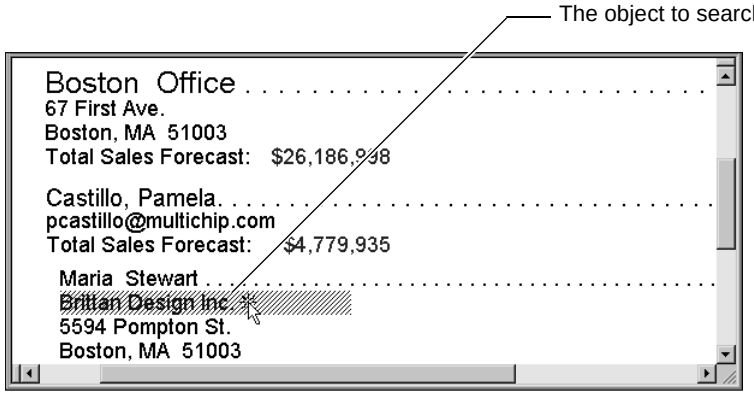


Figure 30-5 Selecting a search criterion

3 To use the selection as a criterion for your search, choose Add from the Search toolbar. The selection appears in Search, as shown in Figure 30-6. The column name appears in Name. An initial value, usually the first value the report displays for the column, appears in Value.

Figure 30-6 Result of adding search criterion

4 In the Value column, specify the value to search. In this example, type:

```
Advanced*
```

The asterisk (*) is a wildcard that specifies you want to find all values that contain Advanced in the first part of the string and any number of characters after that.

5 For a multifield search, repeat steps 2 to 4. An example of a multifield search is shown in Figure 30-7.

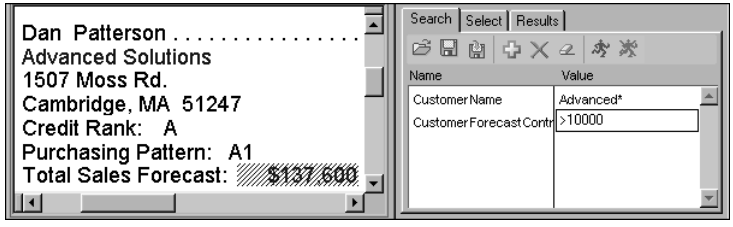


Figure 30-7 Specifying a multifield search

6 Choose Search to start the search process. Matches to the search target(s) appear in Results in order of their occurrence in the report, as shown in Figure 30-8.

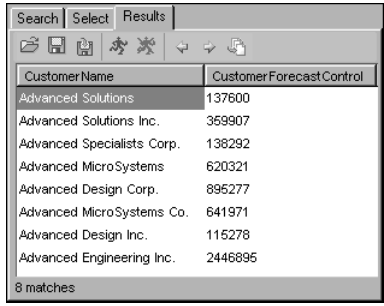


Figure 30-8 Search results

How to modify search conditions

You can change your search conditions any time before you choose Search, which starts the search process.

1 To remove a field from the search, in Search, select a search field entry and choose Delete.

2 To remove all the fields from the search, choose Clear All.

3 To stop searching altogether, choose the X option at the top of the Search window to close it.

Using a search expression in the search conditions

When you specify the value to search for in a report field, you can specify a literal value, such as Leslie Thompson, to search for one match. Typically, though, you specify a search expression to find a set of matches. For example, you can specify the expression, "*Thompson" to find all names that end with Thompson.

Actuate Basic reports support several operators and wildcard characters that you can use in search expressions. You can create complex expressions that use a combination of operators and wildcard characters.

Using operators in a search expression

Table 30-1 lists and describes the operators you can use in search expressions specified in the Value column in the search window.

Table 30-1 Search expression operators

Operator	Description	Examples	Matches
=	Equals. By default, the = operator is implied.	=MR1500	MR1500
		MR 1500	MR1500
>	Greater than or alphabetically after	>100	101, 115, 200
		>Ace	Acer, Adobe
<	Less than or alphabetically before	<100	10, 50, 99
		<Ace	Aamco, Abel
>=	Greater than or equal to	>=100	100, 101, 200
		>=Ace	Ace, Adobe
<=	Less than or equal to	<=100	10, 50, 100
		<=Ace	Ace, Aamco
-	Range. Hyphen separates upper and lower limits of the range.	10 - 20	10, 15, 20
		A - C	Ace, Bell, Core
	For strings, - can mean the following: ■ b- is equivalent to >=b ■ -b is equivalent to <=b ■ - is equivalent to * (match all values)	-Ace	Ace, Aamco
		Ace-	Ace, Adobe
,	Or. Comma separates two values.	1,2	1, 2
		Ace, Ford	Ace, Ford
!	Not	!1000	998, 999, 1001
		!MA	CA, NJ, OH

Using a wildcard character in a search expression

Use wildcard characters to do pattern matching on text objects. Table 30-2 lists and describes the wildcard characters you can use in search expressions specified in the Value column in the search window.

Table 30-2 Wildcard characters

Wildcard	Description	Example	Matches
?	Find any one character	M?1680	MR1680, MS1680
*	Find any number of characters	3M*	3M A1000, 3M B2000

(continues)

Table 30-2 Wildcard characters (continued)

Wildcard	Description	Example	Matches
#	Find any one ASCII numeric character (0 - 9)	MS##90	MS0490, MS3290

Using a pattern in a search expression

Use patterns in a search expression to qualify a search. Patterns let you search for characters in a range, or select only certain characters for searching. Use the brackets ([ ]) to place a pattern inside a search expression.

Table 30-3 lists and describes examples of patterns you can use.

Table 30-3 Pattern examples

Pattern	Description	Example	Matches
[character list]	Match any one character inside the brackets	M[PRS]16	MP16, MR16, MS16
[a-z]	Match any lowercase character	m[a-f]1800	ma1800, mc1800
[0-9]	Match any ASCII numeric character	MX150[1-5]	MX1502, MX1503, MX1505
[a-z0-9]	Match any lowercase character and ASCII numeric character	m[a-c1-3]	ma1, ma3, mb2, mc3
[a-z-]	Match any character or hyphen	m[a-z-]	ma, mb, m-
[^]	Match one caret	*[^]1650	a^1650, b^1650

Searching for the ?, *, #, and [ ] characters

Because the characters ?, *, #, and [ ] have special meanings in search expressions, you must indicate when you want to search for the characters themselves. To search for a special character itself, you can do one of the following:

- Place the character inside brackets.
- Use the backslash (\) before any special character, and enclose both the backslash and the character in quotation marks.

Table 30-4 lists and describes examples of how to search for special characters.

Table 30-4 Searching for special characters

Special characters	Description	Examples	Matches
[?] or "\?"	Match one question mark	M[?]1600 M"\?"1600	M?1600
[#] or "\#"	Match one pound sign	M[#]1600 M"\#"1600	M#1600
[*] or "*"	Match one asterisk	M[*] 1600 M"*"1600	M*1600
[[] or "\["	Match one open bracket	M[[]A[]]6 M"\["A"\]"6	M[A]6
[]] or "\]"	Match one close bracket	M[[]A[]]6 M"\["A"\]"6	M[A]6
["\["-"\]"]	Match any ASCII character between [and]	M["\["-"\]"]	M[A] M[b]

Searching for a string that contains a special character

If a search text string contains one or more special characters, you must type a \ before each special character and enclose the string in quotation marks. Special characters include the following characters that are operators in a search expression:

- Comma (,)
- Hyphen (-)
- Exclamation point (!)
- Less than sign (<)
- Greater than sign (>)
- Equal sign (=)
- Backslash (\)

For example, the string:

```
16M x 1 Dynamic Ram, 3.3 volts
```

must contain \ before the comma and the string must be enclosed in quotation marks:

```
"16M x 1 Dynamic Ram\, 3.3 volts"
```

If you do not type \ before the comma, the comma is interpreted as an OR in SQL, as follows:

```
WHERE items.itemcode LIKE '16M x 1 Dynamic Ram%'
   OR items.itemcode LIKE '3.3 volts%'
```

Using a nonprinting ASCII character in a search expression

Tabs, line feeds, and carriage returns are examples of nonprinting ASCII characters. With the backslash as an escape character, use the nonprinting ASCII characters shown in Table 30-5.

Table 30-5 Nonprinting ASCII characters

Special characters	Definition
\	Escape next character (one- or two-byte character)
\a	Match one alarm (0x07)
\b	Match one backspace (0x08)
\t	Match one tab (0x09)
\n	Match one new line (0x0a)
\f	Match one form feed (0x0c)
\r	Match one carriage return (0x0d)
\\	Match one backslash (\)
[\1-\377]	Match any one-byte character excluding NUL (0x00)
[\x100-xffff]	Match any two-byte character

Using search results to navigate through the report

The search results are linked to the report. You can select an item displayed in Results to go to the report page that contains the corresponding data. Items are listed in the order in which they appear in the report.

How to use Results to go to a specific match in the report

Results displays the matches found in the report. Figure 30-9 shows the matches retrieved from a search for all company names that begin with Advanced, and how to use the results to go to specific matches in the report.

1 Select the match you wish to see.

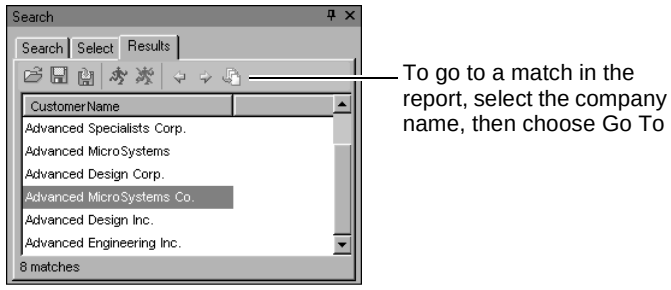


Figure 30-9 Selecting a match

2 Choose Go To or double-click the selection.

The report becomes active and displays the matching value. If you selected Advanced MicroSystems Co. as shown in Figure 30-9, the page containing the company name, Advanced MicroSystems Co., appears. The company name is highlighted.

Figure 30-10 shows the match from the search value.

3 To go to another match in the report, use the Previous, Next, or Go To button in Results.

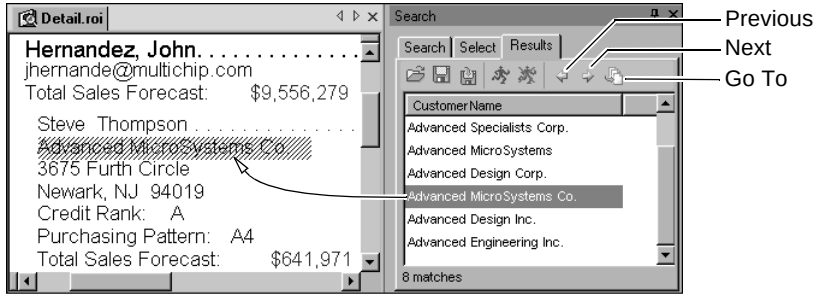


Figure 30-10 Going to a match in the report from a search value

Specifying data to display in Results

Besides displaying the results of a search, Results can display data from other objects in a report. If, for example, you search for companies by state, Results displays, by default, only the company names that match the search conditions. You might, however, also want to display the full company address and credit rank next to the company name, as shown in Figure 30-11.

Text fields displayed in Results have a limit of 255 characters. For example, each field under CompanyName can have 255 characters in length. If a text field in a report contains more than 255 characters, the first 255 characters are used.

Figure 30-11 shows results with multiple fields of data for each result.

You can specify any number of object values to display in Results. This capability is useful for creating a quick view of particular data in a report.

The data in Results usually appears as described in the preceding example and in the following procedure, however, in a search of a report designed using a hierarchy of data groups, unexpected results can appear.

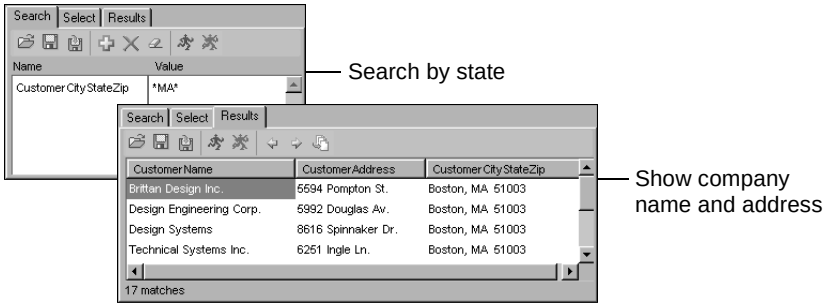


Figure 30-11 Results with multiple fields of data

How to specify object data to display in Results

The instructions in this section assume you already specified the search conditions.

1 Choose Select in Search.

Select appears. By default, the name of the object or objects you selected as criteria for the search appears in the Name column. The values of these objects, in turn, appear in Results after you execute the search.

Figure 30-12 shows how to select various criteria.

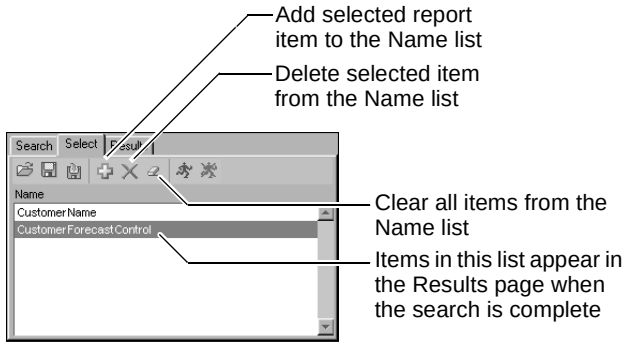


Figure 30-12 Selecting various criteria

2 If you want Results to display only some of the objects, select each object to exclude. Then choose Delete. To exclude all objects, choose Clear All.

3 To specify additional objects, select each object you want to add. Then choose Add Selection on Select.

The names of the objects you select appear in the Name list.

4 Choose Search to start the search process.

The search results appear in Results, as illustrated in Figure 30-13.

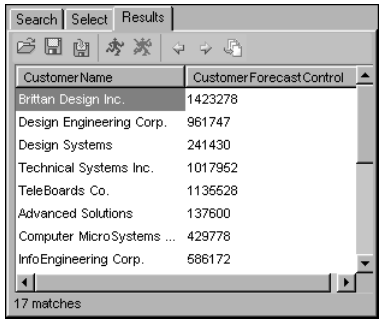


Figure 30-13 Search results

Creating a search definition

After you set up a search, you can save the search conditions and options in a search definition (.ros) file. Actuate e.Report Designer Professional saves the following information in an ROS:

- Search conditions in Search
- Data to display in Select
- Report document structure information

  Actuate e.Report Designer Professional uses the report document structure information when it runs the search on a report document.

To save the ROS, choose Save in the Search dialog. The ROS is saved in the Actuate search definition directory. If you use the default installation directory, the ROS is saved in:

```
C:\Program Files (x86)\Actuate11\Srchdef
```

You can use search definition files with different report object instance (.roi) files if the ROIs are generated from the same report object executable (.rox) file. For example, if you have an ROX that you use to create weekly reports, you can create an ROS that searches for and extracts data from the report object instance (.roi)

file. You can use the same ROS with any weekly report as long as the ROX did not change.

You can open and run a search definition (.ros) file from either the Search dialog or from the toolbar. You can open and use an ROS only from your local computer. The report object instance (.roi) file can be in the Encyclopedia volume or on a system to which you have access. The ROS must be on your system or another system to which you have access, outside the Encyclopedia volume. e.Report Designer Professional searches for the search definition files in the report document's directory and the Actuate search definition directory. For example, if you use the default installation directory on a Windows 7 platform, the search definition directory is:

```
C:\Program Files (x86)\Actuate11\Srchdef
```

If the ROS cannot be used with the current report, Actuate e.Report Designer Professional displays a message when you try to use it.

From the Search dialog, choose Open to open a search without running it. Choose Search to run the search.

Chapter

31

Printing a report

This chapter contains the following topics:

- Printing a report from your desktop
- Running a report before printing

Printing a report from your desktop

To print an Actuate Basic report, you use the standard Windows procedure to select a printer and set printer properties. In addition, Actuate e.Report Designer Professional provides printing options specific to Actuate Basic reports.

How to print a report from your desktop

1. Open the report you want to print.
2. Choose File➤Print.

   Print appears, as shown in Figure 31-1. The options available from this dialog enable you to select the printer and report printing options such as page range and number of copies.

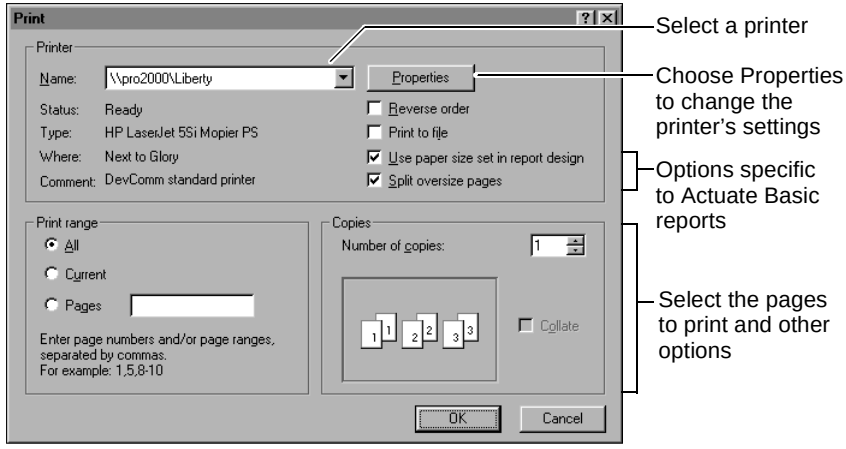


Figure 31-1 Print options

3. Select the desired printing options, the printer, the pages to print, the order in which to print the pages, and the number of copies to print. Consider the following options:

   - To select the paper size that matches the page size defined in the report design, select Use paper size set in report design. For example, if the report design specifies a page size of 8.5" by 14", selecting this option prints the report on 8.5" by 14" paper, if the printer supports this paper size.
   - To specify that large report pages print properly onto multiple sheets of paper, select the Split oversize pages.

4. To specify a printer's other properties, or to change the printer's default settings, choose Properties.

Document Properties appears. Figure 31-2 shows the Layout tab of the dialog for an HP LaserJet 5Si Mopier PS printer driver. Options and defaults are different for different printers.

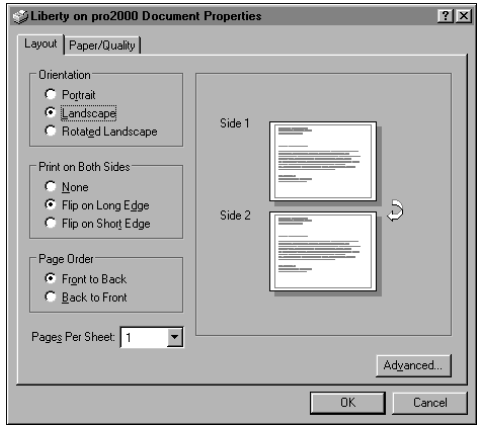


Figure 31-2 Document properties

5 Set the options in this dialog to specify layout-related settings such as orientation, duplex printing, page order, and the number of pages per sheet to print.

The layout of the report when the report is designed determines the printed report's page orientation.

6 To set advanced options, choose Advanced.

Advanced Options appears. Figure 31-3 shows HP LaserJet 5Si Mopier PS Advanced Options. Options and defaults are slightly different for different printers.

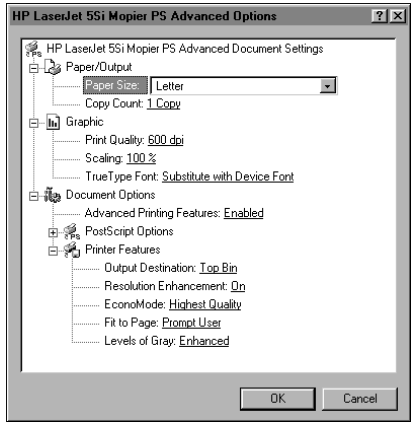


Figure 31-3 Advanced options

Specify the advanced options. Choose OK.

7 To view and change other printer settings, choose Paper and Quality.

Figure 31-4 shows Paper and Quality for an HP LaserJet 5Si Mopier PS printer. Options and defaults are slightly different for different printers.

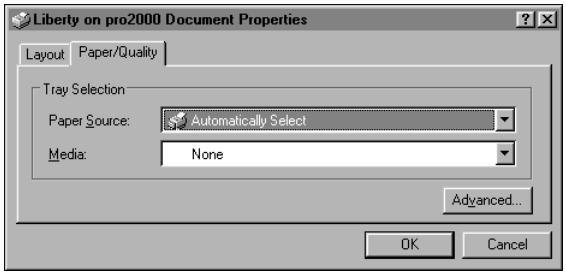


Figure 31-4 Paper and quality options

Specify the paper and quality options. Choose OK.

8 In Print, choose OK to start printing.

How to preview a report before printing

1 Choose File➛Print Preview to preview the report before printing it.

Previewing gives you an opportunity to review and change the printing options you selected. Figure 31-5 shows an example of a report preview.

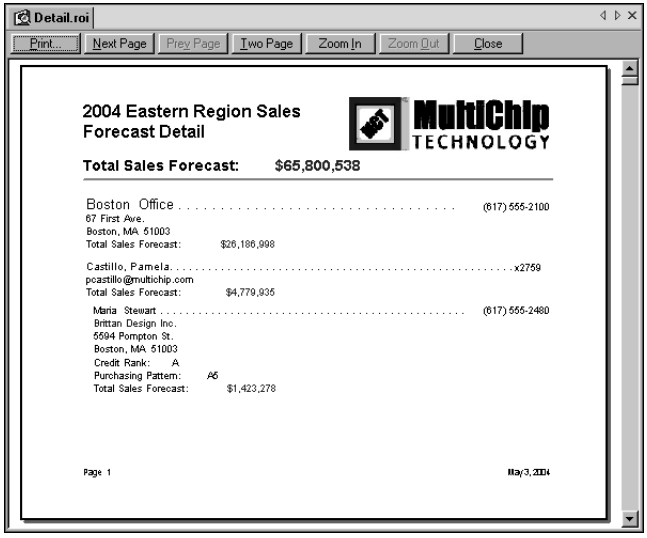


Figure 31-5 Previewing the report

2 Use Next Page, Prev Page, Two Page, Zoom In, and Zoom Out to see different pages and to see different levels of detail.

3 When you finish previewing the report, choose Close to close the preview window or choose Print to print the report.

Running a report before printing

If a report object executable (.rox) file is available, you can generate a new report to print. There are reasons for running a report just before printing:

- To use the most current data
- To specify parameter values that are more useful than the default values

Chapter

32

Rendering to PDF

This chapter contains the following topics:

- About the Render architecture
- Working with multiple Render profiles
- About the PDF Writer
- Render profile XML reference
- Render profile parameter reference

About the Render architecture

Render is an architecture in e.reports for rendering reports to various output formats. This architecture provides a foundation and functionality for future output formats and enhancements.

You customize the behavior of the Render system using render profiles. A render profile is a named set of options that specify how to render report content to output documents.

Render profiles are defined in the Render profiles file. You can customize the name and location of the Render profiles file. The default name of the Render profiles file is AcRenderProfiles.xml. For e.Report Designer Professional, the default location of the Render profiles file is e.Report Designer Professional's config folder.

To configure the name and location of the Render profiles file for e.Report Designer Professional:

1. Choose Tools➛Options.
2. In Options, choose General.
3. In Options—General, in Render profiles file path or URL, type the full path or URL for the Render profiles file. The URL must be a direct reference to the Render profiles file itself. Access to the Render profiles file through a servlet or equivalent is not supported.
4. Choose OK.

Working with multiple Render profiles

The Render profiles file can contain multiple Render profiles. You can use different Render profiles to select different rendering options for specific documents. Consider creating the following Render profiles:

- By creating a separate Render profile that enables multithreading, you can use multithreading on only the documents that are rendered when your system is lightly loaded. This Render profile prevents degrading the rendering performance at times of heavy system load.
- By creating a different Render profile that specifies a high resolution for charts, you can create documents that print with high quality without degrading performance when rendering documents that are intended for viewing on screen.

Selecting a Render profile at design time

You can specify the Render profile that is to be used for all reports generated from a particular report design. To do this, set the RenderProfileId property in the report design's root component to the id of the profile you wish to use.

Selecting a Render profile at generation time

You can specify the Render profile that is to be used for an individual report when that report is generated. To do this, associate a report parameter with the RenderProfileId property in the report design's root component. When you generate the report, set the value of the parameter to the id of the profile you wish to use.

To use data within the report to choose a Render profile, you can set the report root's RenderProfileId value in Basic code in an overridden method.

Selecting a Render profile at view time

You can specify the Render profile that is to be used for individual report when that report is rendered to an output document. To do this, override the GetRenderProfileId( ) method in the report design's root component. In the method code you can check values such as the viewing locale and user id and use those to decide which Render profile to select.

About the PDF Writer

The PDF Writer uses the Render architecture to export report documents to PDF. The PDF Writer supports only TrueType fonts.

Using the new PDF Writer in e.Report Designer Professional

To use the new PDF Writer in e.Report Designer Professional, use File➤Save As➤PDF while viewing a report.

Using the new PDF Writer in iServer

This section describes how to enable and disable the Render system.

Enabling the Render system

In iServer, the Render system is disabled by default. The old PDF Converter continues to be used. To use the new PDF Writer in iServer, complete the following tasks to enable the Render system:

1. Log in to iServer Configuration Console.
2. Choose Advanced view.
3. Choose Server Configuration Templates.
4. In Server Configuration Templates, in Templates, choose the name of the server on which you want to enable the Render system.
5. In Properties settings, expand the following folders: Viewing Service➤e.Report➤Report Rendering➤PDF Generation. Choose Render Profiles.
6. In Render Profiles, select Enable rendering using profiles.

   URL for render profiles displays the path to AcRenderProfiles.xml, for example:

   ```
   $AC_SERVER_HOME$/etc/AcRenderProfiles.xml
   ```

   Choose OK.
7. In Server Configuration Templates>ServerName:Settings. Choose OK.
8. To apply the change, complete the following tasks to restart iServer:
   1. In the side menu, choose System.
   2. In System: Status, choose Stop.

      The system status changes to System is currently offline.
   3. Choose Start System.

      The system starts and your changes apply.

Enabling Render in a report design

Once the Render system has been enabled in iServer, you then must specify in the report design that the report is to be rendered using the new PDF Writer.

In e.Report Designer Professional, to specify that a report should be rendered using the new PDF Writer:

1. Set the RenderProfileId property of the report design's root component to a valid Render profile id, such as "Default".
2. Compile the report design and upload the resulting ROX file to iServer.

Disabling the new PDF Writer

If you encounter compatibility issues when using the new PDF Writer, you can revert to the old PDF Converter. Revert to the old PDF Converter in one of the following ways:

- Disable the entire Render system by following the steps earlier in this section to enable the Render system, but deselect the Enable Render check box instead of selecting it. This approach applies to existing ROI files.
- Set the RenderProfileId property of an individual report design's root component to an empty string, recompile the report design and upload the new ROX file to iServer. This approach does not prevent use of the new PDF Writer for existing ROI files generated by ROX files that had non-empty RenderProfileId properties.

Creating a PDF document during report generation

The Render architecture allows you to create PDF documents during report generation, by writing custom Actuate Basic code. You can see sample reports that demonstrate how to do this in e.Report Designer Professional's examples \Render folder.

Render profile XML reference

Render Profies are based on XML. The first line of the Render profiles file must always be the following XML header:

```
<?xml version="1.0" encoding="UTF-8"?>
```

FontFamily element

A font family is a set of related fonts, comprising regular, bold, italic, and bold+italic styles. The FontFamily element defines a single font family. The attributes for the FontFamily element are shown in Table 32-1.

Table 32-1 Attributes for FontFamily Element

Attribute	Description
boldFileName	Optional. The name of the font file that contains the bold style of the font family. If this attribute is not present, the value of the regularFileName attribute is used.
boldItalicFileName	Optional. The name of the font file that contains the bold+italic style of the font family. If this attribute is not present, the value of the regularFileName attribute is used.

(continues)

Table 32-1 Attributes for FontFamily Element (continued)

Attribute	Description
encoding	Optional. Indicates the encoding to use when rendering the font. The default value is Windows-1252. Permitted values are: ■ UCS-2 ■ Windows-1252 ■ Chinese (Simplified) ■ Chinese (Traditional) ■ Japanese ■ Korean
italicFileName	Optional. The name of the font file that contains the italic style of the font family. If this attribute is not present, the value of the regularFileName attribute is used.
name	Required, case sensitive. The font face name of the font family. The font face name must be exactly the same as the Font.FaceName property in the report design, including spaces.
pdfEmbed	Optional. Indicates whether the font is embedded in PDF documents. The permitted values are: ■ No — The font is not embedded. ■ Subset — A subset of the font that contains only the characters used in the document is embedded. ■ All — The whole font is embedded.
regularFileName	Required if the font is embedded in documents. The name of the font file that contains the regular style of the font family.

FontFamily elements can only appear within profile elements.

On UNIX and Linux operating systems, font file names are case-sensitive. Make sure that font file names are all uppercase in both the file system and the Render profiles file on these operating systems.

Some font files contain flags that forbid or restrict embedding. If you attempt to embed a font with these restrictions, rendering fails. If you need to embed a font whose font file does not allow embedding, contact your font supplier to obtain an unrestricted version.

Multiple TrueType fonts can be contained in a single file. This is known as a TrueType collection. TrueType collection files usually have a .TTC file extension.

The specific encoding used in output documents depends on both the encoding attribute and the document format. For example, non-embedded fonts with Japanese encoding are rendered in PDF documents using the Adobe UniJIS-UCS2-H encoding.

PDF does not support the use of Unicode encoding for fonts that are not embedded. If you specify UCS-2 encoding for a font that is not embedded, a warning message is emitted and Windows-1252 encoding is used instead.

When rendering to PDF, the value of the encoding attribute is ignored for fonts that are embedded.

Embedded fonts are always encoded as Unicode UCS-2.

The license terms of some font files may forbid or restrict permit embedding, even if the flags that restrict or forbid embedding are not set. It is your responsibility to ensure that you have the appropriate licenses for any fonts that you embed.

If the font is embedded, the encoded attribute is not required, but it is strongly recommended it be used. If the font is not embedded, UCS-2 encoding is not permitted for PDF generation. The values for the encoded attribute must be used exactly as shown in Table 32-1, including hyphens, spaces and parenthesis.

Examples of the FontFamily element are shown in the following code:

```
<Profile id="Example">
   <!-Book Antiqua has bold and italic styles -->
   <FontFamily
      name="Book Antiqua"
      regularFileName="BKANT.TTF"
      boldFileName="ANTQUAB.TTF"
      italicFileName=" ANTQUAI.TTF"
      boldItalicFileName="ANTQUABI.TTF"
      pdfEmbed="Subset"
      encoding="UCS-2"
   />
   <!-- Arial Unicode MS has no bold or italic styles -->
   <FontFamily
      name="Arial Unicode MS"
      regularFileName="ARIALUNI.TTF"
      pdfEmbed="Subset"
      encoding="UCS-2"
   />
   <!-- Chinese font name encoded as UTF-8 -->
   <FontFamily
      name="???"
      regularFileName="SIMHEI.TTF"
```

```
   encoding="Chinese (Simplified)"
   pdfEmbed="Subset"
/>
<!-- Gungsuh is part of a TrueType collection -->
<FontFamily
   name="Gungsuh"
   regularFileName="BATANG.TTC"
   pdfEmbed="Subset"
   encoding="Korean"
/>
```

IncludeProfile element

Render profiles can include the definitions of other Render profile. The IncludeProfile element specifies that the definition of one Render profile is included in the definition of another Render profile. The attribute for the IncludeProfile element is shown in Table 32-2.

Table 32-2 IncludeProfile attribute

Attribute	Description
id	Required. The id of the Render profile whose definition is to be included.

IncludeProfile elements can only appear within Profile elements.

Included Render profiles must be defined before they are used.

If a Render profile contains multiple definitions of any value or attribute, the last definition of that value or attribute is used.

An example of the IncludeProfile element is shown in the following code:

```
<Profile id="Default">
   <IncludeProfile id="Base Fonts">
   …
</RenderProfiles>
```

Parameter element

The Parameter element defines a single parameter. The contents of the Parameter element are the value of the parameter. The attributes for the Parameter element are shown in Table 32-3.

Table 32-3 Parameter element attributes

Attribute	Description
name	Required. The name of the parameter.

Table 32-3 Parameter element attributes

Attribute	Description
type	Required. The data type of the parameter. The supported values are Boolean, Double, Integer, and String.

The following code shows an example of the Parameter element:

```
<ParameterGroup name="Render">
   <Parameter name="AddTOC" type="Boolean">False</Parameter>
   <Parameter name="Author" type="String">Actuate</Parameter>
   <Parameter name="DrawingDPI" type="Double">192</Parameter>
</ParameterGroup>
```

ParameterCollection element

A Render profile can contain multiple sets of parameters specific to individual output formats. The ParameterCollection element defines a single set of parameters within a Render profile. The attribute for the ParameterCollection element is described in Table 32-4.

Table 32-4 ParameterCollection element attribute

Attribute	Description
format	Required. The output format for which the parameter collection is used. If the value is “Default”, the parameter collection is used for all output formats.

ParameterCollection elements may only appear within profile elements.

If a profile contains a default parameter collection and a format-specific parameter collection, the two parameter collections are merged. If a parameter has values in both parameter collections, the value in the last parameter collection is used.

An example of the ParameterCollection element is shown in the following code:

```
<Profile id="Default">
   <ParameterCollection format="Default">
   …
   </ParameterCollection>
   <ParameterCollection format="PDF">
   …
   </ParameterCollection>
   …
```

ParameterGroup element

A parameter collection can contain multiple named groups of parameters. The ParameterGroup element defines a single group of parameters within a parameter collection. The attribute for the ParameterGroup element is described in Table 32-5.

Table 32-5 ParameterGroup element attribute

Attribute	Description
name	Required. The name of the parameter group. This must always be “Render”.

ParameterGroup elements can only appear within ParameterCollection elements.

Each parameter collection within a Render profile must contain a single parameter group whose name is “Render”.

An example of the ParameterGroup element is shown in the following code:

```
<ParameterCollection format="Default">
   <ParameterGroup name="Render">
      …
   </ParameterGroup>
      …
```

Profile element

The Render profiles file can contain multiple profiles. The Profile element defines a single Render profile within the Render profiles file. The attribute for the Profile element is shown in Table 32-6.

Table 32-6 Profile element attribute

Attribute	Description
id	Required. The unique id of the Render profile.

Profile elements can only appear within the RenderProfiles element.

The Render profiles file must always contain a profile whose id is “Default”.

An example of the Profile element is shown in the following code:

```
<RenderProfiles>
   <Profile id="Default">
      …
   </Profile>
   <Profile id="Debug">
```

```
      …
   </Profile>
      …
</RenderProfiles>
```

RenderProfiles element

The RenderProfiles element must always be present as the outermost element in the Render profiles file. There are no attributes for the RenderProfiles element.

An example is shown in the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
   <RenderProfiles>
      …
   </RenderProfiles>
```

Render profile parameter reference

Render parameters are used to specify how report content should be rendered into output documents.

AbortOnInvalidImage parameter

AbortOnInvalidImage is a Boolean parameter that specifies whether rendering halts if an image to be displayed within a report cannot be found.

The default value of AbortOnInvalidImage is True.

AddDebugPages parameter

AddDebugPages is a Boolean parameter that specifies whether to insert additional pages that show debugging information in output documents.

The default value of AddDebugPages is False.

AddTOC parameter

AddTOC is a Boolean parameter that specifies whether to add a table of contents to output documents.

The default value of AddTOC is False.

Author parameter

Author is a String parameter that identifies the person or organization that created a report.

The default value of Author is Actuate Corporation.

ClipText parameter

If a control contains a single line of text, and the font size is too large for the height of the control, the default behavior in some output formats is for the text to extend below the bottom edge of the control. ClipText is a Boolean parameter that forces text to be clipped to the bottom edge of controls in this situation.

The default value of ClipText is False.

The effect of ClipText=False is shown in Figure 32-1. The effect of Cliptext=True is shown in Figure 32-2.

Figure 32-1 ClipText = False

Figure 32-2 ClipText = True

Setting ClipText to True degrades performance for some output formats, even if no text overflows the bottom edges of controls. It is not normally necessary to set ClipText to True.

CompressionLevel parameter

CompressionLevel is a String parameter that specifies the level of compression to apply to output formats that support variable compression. The supported values are None, Low, Medium, and High.

The default value of CompressionLevel is Medium.

Using a higher level of compression degrades performance. For most situations, Medium is the optimum value for CompressionLevel.

DrawingDPI parameter

DrawingDPI is a Double parameter that specifies the default resolution of bitmap images rendered from drawings and charts, in dots per inch. A larger value produces images that appear crisper on high resolution devices such as printers. A smaller value improves performance and produces significantly smaller documents.

The default value of DrawingDPI is 192.0.

DrawingQuality parameter

DrawingQuality is a Double parameter that specifies the default quality multiplier for bitmap images rendered from drawings and charts. The value of the DrawingDPI parameter is multiplied by the value of DrawingDPI to get the actual dpi for the images. For example, if DrawingDPI is 192.0 and DrawingQuality is 2.0, drawings and chart s are rendered at 384 dpi.

A larger value produces images that appear crisper on high resolution devices such as printers. A smaller value improves performance and produces significantly smaller documents.

The default value of DrawingQuality is 1.0.

EmbedDebugInfo Parameter

EmbedDebugInfo is a Boolean parameter that specifies whether to embed debugging information within output documents.

The default value of EmbedDebugInfo is False.

EnableExternalHyperlinks parameter

EnableExternalHyperlinks is a Boolean parameter that specifies whether to add external hyperlinks, or links that begin with “http:” or “https:”, to output documents.

The default value of EnableExternalHyperlinks is False.

InitialView parameter

InitialView is a String parameter that specifies how an external application displays a rendered document when it is opened. The supported values depend on the output format and are described in Table 32-7.

The default value of InitialView is an empty string.

Some output document formats do not support control over the initial display of documents. The value of InitialView is ignored for those formats.

The value of InitialView might be ignored by some applications.

Table 32-7 InitialView values for PDF

Value	Description
Empty string or OneColumn	Display pages scaled to fit the window's width. Scrolling down displays the bottom of the current page and the top of the next page.
SinglePage	Display one page at a time, scaled to fit the window's height and width. Scrolling down jumps to the next page.
TwoColumnLeft	Display two pages at a time, scaled to fit the window's width, with odd numbered pages on the left. Scrolling down displays the bottom of the current pair of pages and the top of the next pair of pages.
TwoColumnRight	Display two pages at a time, scaled to fit the window's width, with odd numbered pages on the right. Scrolling down displays the bottom of the current pair of pages and the top of the next pair of pages.
TwoPageLeft	Display two pages at a time, scaled to fit the window's height and width, with odd numbered pages on the left. Scrolling down jumps to the next pair of pages.
TwoPageRight	Display two pages at a time, scaled to fit the window's height and width, with odd numbered pages on the right. Scrolling down jumps to the next pair of pages.

Generator parameter

Generator is a String parameter that identifies the program used to create a report.

The default value of Generator is Actuate e.Reports.

MaximumNumberOfPages parameter

MaximumNumberOfPages is an Integer parameter that specifies a limit on the number of pages that can be rendered into a single output document. A value of 0 indicates that there is no limit.

The default value of MaximumNumberOfPages is 0.

Multithread parameter

Multithread is a Boolean parameter that specifies whether to use multiple threads to render a document. If your system has unused CPU capacity, using multiple threads can reduce the time it takes to render large documents. However, if your system is heavily loaded, using multiple threads can degrade performance.

The default value of Multithread is False.

PickPrinterTrayByPageSize parameter

PickPrinterTrayByPageSize is a Boolean parameter that specifies the default behavior of the print dialog in external applications that display a rendered document. If PickPrinterTrayByPageSize is True, the print dialog automatically selects a paper tray that matches the page size of the document. If PickPrinterTrayByPageSize is False, the print dialog selects the default printer tray.

The default value of PickPrinterTrayByPageSize is True.

Some output document formats do not support printer tray selection. The value of PickPrinterTrayByPageSize is ignored for those formats.

The value of PickPrinterTrayByPageSize might be ignored by some applications.

SplitOversizePages parameter

SplitOversizePages is a String parameter that specifies how to render pages that are larger than a target size to output documents. The supported values are described in Table 32-8.

Table 32-8 SplitOversizePages parameter settings

Setting	Description
Default	Split oversize pages if the report or user requested splitting.
No	Do not split oversize pages.
Yes	Split oversize pages.

The default value of SplitOversizePages is Default.

Writer parameter

Writer is a String parameter that identifies the program to use to render a document. The default value of Writer is Actuate PDF Writer.

Appendix

A

Working with RTF and HTML tags in a dynamic text control

This appendix contains the following topics:

- Working with RTF tags
- Working with HTML tags

Working with RTF tags

The RTF tags that you use in a dynamic text control must be well formed. An RTF tag starts with a \ and ends with a space or a new line character. To turn off the effect of a tag, use the same tag with a 0 after it. For example, the following RTF code:

```
Hello world\par
\b Hello stars\b0\par Hello moon
```

produces the following text that contains bold text and multiple paragraphs:

```
Hello world
Hello stars
Hello moon
```

You can concatenate tags. For example, the following RTF code:

```
\b\i Hello\i0  stars\b0
```

produces the following text that is both bold and italic:

***Hello* stars**

You do not need to provide a complete RTF document by providing an RTF header.

Table A-1 lists the supported RTF structural and definition tags. For more information about using RTF tags, refer to RTF documentation, such as "Rich Text Format (RTF) Specification, version 1.6," which is available at the following URL:

`http://latex2rtf.sourceforge.net/rtfspec.html`

Table A-1 RTF structural and definition tags

RTF tag supported	Description
*	This extended feature group indicator specifies the contents of groups identified in this way are ignored.
\blueN	Specifies blue value in color table.
\colortbl	Specifies the color table.
\fonttbl	Specifies the font table.
\greenN	Specifies the green value in color table.
\redN	Specifies the red value in color table.
\rtfN	Specifies the RTF dynamic text. N indicates the major RTF version number. The value of N must be 1.

Table A-1 RTF structural and definition tags

RTF tag supported	Description
\ucN	Specifies the number of text characters to skip following a \uN tag. RTF text can contain alternative representations for Unicode characters for use by RTF readers that do not support Unicode. RTF readers that do not support Unicode skip the \uN tag and display the following character. For example, \rtf1\u8364 E displays the Euro currency symbol on an RTF reader that supports Unicode and E on a reader that does not support Unicode. RTF text that contains alternative representations for Unicode characters uses \ucN to enable RTF readers that support the \uN tag to display the Unicode character(s) and skip the following character(s).
\uN	Specifies a Unicode character.

Table A-2 lists the supported RTF paragraph format tags.

Table A-2 RTF paragraph format tags

RTF paragraph tag supported	Description
\fiN	Specifies first line left indent. N indicates the indent in twips. \fi is additive with \li and N can be a negative number. For example, \li1440\fi-720 makes the first line indented 0.5" and remaining lines indented 1.0".
\keep	Specifies keeping the paragraph intact. This tag prevents page breaks in the middle of a paragraph.
\keepn	Specifies keeping the paragraph with the next paragraph. This tag adjusts the page break so that both paragraphs are rendered on the same page.
\liN	Specifies left indent. N indicates the indent in twips.
\pard	Specifies resetting all paragraph formatting attributes.
\qc	Specifies centering text.
\qj	Specifies justified alignment.
\ql	Specifies left-aligned text.
\qr	Specifies right-aligned text.
\riN	Specifies right indent. N indicates the indent in twips.

(continues)

Table A-2 RTF paragraph format tags (continued)

RTF paragraph tag supported	Description
\saN	Specifies vertical space after the paragraph. N indicates the space in twips.
\sbN	Specifies vertical space before the paragraph. N indicates the space in twips.
\slN	Specifies vertical line spacing. N indicates the total line height in twips.
\slmultN	Specifies vertical line spacing multiplier. N indicates the multiplication type.
\txN	Specifies the tab stop. N indicates the tab stop position in twips.

Table A-3 lists the supported RTF character format tags.

Table A-3 RTF character format tags

RTF character tag supported	Description
\b	Specifies start bold text.
\b0	Specifies end bold text.
\cfN	Specifies color number. N indicates the color number in the color table.
\fN	Specifies font number. N indicates the font number in the font table.
\fsN	Specifies font size. N indicates the font size in half points. Because AFC does not recognize fractional font sizes, half point sizes round to integers. For example, \fs21 indicates font size 10.5pt, which rounds to 10pt.
\i	Specifies start italic text.
\i0	Specifies end italic text.
\plain	Specifies end all character attributes, such as bold, font, and italic.
\ul	Specifies start underline.
\ul0	Specifies end underline.

The dynamic text control recognizes any Unicode character encoded in the standard RTF form \un where n is the decimal Unicode value of the character. If

the specified character does not exist in the requested font, the null symbol (Unicode value 127) is used.

The dynamic text control recognizes any character encoded in the standard RTF format \'xx where xx is the 2-digit hex value of the character. This format is treated as a Unicode value, regardless of the original document encoding. If the specified character does not exist in the requested font, the null symbol (hex value 0x7f) is used.

Table A-4 lists the supported special characters.

Table A-4 Special characters

Character	Unicode	Tag	Description
\	\u92	\\	Backslash
		\par	Paragraph break
		\<LF>	Paragraph break. <LF> is ASCII 10.
		\<CR>	Paragraph break. <CR> is ASCII 13.
		\line	Line break
		\tab	Move to next tab stop.
		<TAB>	Move to next tab stop. <TAB> is ASCII 8.
		\emspace	Space equal in width to m
		\enspace	Space equal in width to n
{	\u123	\{	Left brace
}	\u125	\}	Right brace
‘	\u145	\lquote	Left single quote
’	\u146	\rquote	Right single quote
“	\u147	\ldblquote	Left double quote
”	\u148	\rdblquote	Right double quote
•	\u149	\bullet	Bullet symbol
–	\u150	\endash	En-dash
—	\u151	\emdash	Em-dash
	\u160	\~	Non-breaking space
-		\-	Optional soft hyphen
-		_	Non-breaking hyphen

Working with HTML tags

HTML tags within a dynamic text control must be well formed and complete. For example, if you need to create a bold set of text, wrap the <B> and </B> tags around the text to format. You can combine multiple HTML tags in one dynamic text control to create text that is formatted in one manner in some portion of the control and in a different manner in another portion.

For more information about using HTML tags, refer to HTML documentation, such as the W3C's "HTML 4.01 Specification," which is available on the web at the following URL:

`http://www.w3.org/TR/html4/`

Table A-5 lists the supported HTML structural tags.

Table A-5 HTML structural tags

HTML structural tag supported	Description
<HEAD> </HEAD>	Specifies HTML document header information. Actuate software ignores all content between <HEAD> and </HEAD>, including any nested tags.
<!-- -->	Specifies a comment. Actuate software ignores all content between <!-- and -->.
<COMMENT> </COMMENT>	Specifies a comment. Actuate software ignores all content between <COMMENT> and </COMMENT>.
<SCRIPT> </SCRIPT>	Specifies a script. Actuate software ignores all content between <SCRIPT> and </SCRIPT>.

Table A-6 lists the supported HTML paragraph format tags.

Table A-6 HTML paragraph format tags

HTML paragraph tag supported	Description
 	Specifies a line break. The tag inside a fully justified paragraph starts a new line without creating a new paragraph. The line containing the tag is not justified.
<CENTER> </CENTER>	Specifies centering text.
<LI> </LI>	Specifies a list item.
<OL> </OL>	Specifies an ordered list.

Table A-6 HTML paragraph format tags

HTML paragraph tag supported	Description
<PRE> </PRE>	Specifies a block of fixed-width text. The text truncates if it is wider than the dynamic text control. The text does not wrap.
<P> </P>	Specifies a paragraph. Supports the ALIGN parameter.
<UL> </UL>	Specifies an unordered list.

Table A-7 lists the supported HTML character format tags.

Table A-7 HTML character format tags

HTML character tag supported	Description
<B> </B>	Specifies bold text.
<FONT> </FONT>	Specifies the font face, size, and color. If a font is not specified, Actuate software uses a default font.
<I> </I>	Specifies italicized text.
<TT> </TT>	Specifies the default fixed-width font.
<U> </U>	Specifies underlined text.

Table A-8 lists the supported HTML parameters.

Table A-8 HTML parameters

HTML parameter supported	Tag	Description and values
ALIGN	<P>	Specifies the alignment. Use CENTER, LEFT, RIGHT, or JUSTIFY.
COLOR	<FONT>	Specifies the foreground color. Use one of the following values: ■ An RGB value specified as a 6-digit hex number in the form #xxxxxx ■ A color name from the following list: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow

(continues)

Table A-8 HTML parameters (continued)

HTML parameter supported	Tag	Description and values
FACE	<FONT>	Specifies the font name, such as Arial. If a font is not specified, Actuate software uses a default font.
SIZE	<FONT>	Specifies the font size. Use one of the following values: ■ An absolute font size specified as an integer. Absolute font size values map to point sizes by the following rules: 0 or 1 = 8 pt, 2 = 10 pt, 3 = 12 pt, 4 = 14 pt, 5 = 18 pt, 6 = 24 pt, 7 or greater = 36 pt. ■ A relative font size specified as an integer with a leading + or -. Relative font sizes are interpreted as offsets from an absolute font size 3. For example, +2 is interpreted as 5.

The dynamic text control recognizes any Unicode character encoded in the standard HTML form &#n; where n is the Unicode value of the character. If the specified character does not exist in the requested font, the null symbol (Unicode value 127) is used.

Table A-9 lists the supported special characters.

Table A-9 Special characters

Character	Unicode	Name	Description
"	"	"	Quotation mark
&	&	&	Ampersand
		&shy	Character entity reference
<	<	<	Less than
>	>	>	Greater than
€	&#8364	€	Euro currency symbol
•	•	•	Bullet point
-		0xAD	Optional hyphen
			Non-breaking space
¡	¡	¡	Inverted exclamation mark
¢	¢	¢	Cent currency symbol
£	£	£	UK pound currency symbol

Table A-9 Special characters

Character	Unicode	Name	Description
¤	¤	¤	General currency symbol
¥	¥	¥	Japanese yen currency symbol
©	©	©	Copyright symbol
®	®	®	Registered trademark symbol
°	°	°	Degree symbol
¿	¿	¿	Inverted question mark
×	×	×	Multiplication operator

Appendix

B

Upgrading reports in batches

This appendix contains the following topics:

- Upgrading reports
- Using the erdpro command to upgrade reports

Upgrading reports

The current release of e.Report Designer Professional can read files in earlier release formats. When you open a report from an earlier release, e.Report Designer Professional displays a message as shown in Figure B-1.

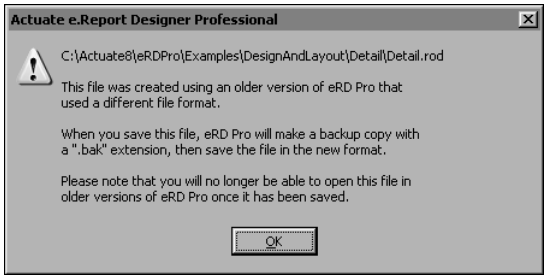


Figure B-1 Legacy file upgrade message

To prevent a user from inadvertently converting a report design or library to a new version's file format, e.Report Designer Professional creates a back-up file for an earlier version ROD or ROL before saving the file as a new version. e.Report Designer Professional saves the earlier version ROD or ROL by appending a .bak file extension, for example saving Detail.rod as Detail.rod.bak.

On the Actuate e.Report Designer Professional message as shown in Figure B-1, if you choose OK, e.Report Designer Professional opens the report. It does not, however, save the report in the new format until you save or build the report. To upgrade the report, save or build the file. To preserve the report's original format, do not save the file when you close it.

To upgrade multiple reports at once, use the erdpro command at the command line.

Using the erdpro command to upgrade reports

In a typical installation, the erdpro command is located in the <eRDPro_HOME>\bin directory. Run the following command in the Command Window or in a batch file to upgrade multiple reports:

```
erdpro [-b][-r][-u] ["<file name>.rod"] ["<file name>.rod"] …
   [-f <log file name>]
```

If you run the erdpro command without any options or file names, the command opens e.Report Designer Professional. If you run the command with file names only, the command opens the first file in the list.

To upgrade reports, use at least one of the options, -b, -r, or -u. You can use more than one option. Table B-1 describes the command options.

Table B-1 Command options

Option	Description
-b	Builds the report object executable (.rox) file, then closes e.Report Designer Professional.
-r	Builds the ROX, runs it to generate a report (.roi), then closes e.Report Designer Professional.
-u	Builds the ROX, updates all files (such as libraries) included with the specified RODs to use relative file paths, then closes e.Report Designer Professional.
-f	Writes information about the processing into the log file that you specify. You can use the -f option only if you use at least one of the other options.

The following are usage examples:

- The following command builds new ROXs for three reports located in the same directory, then closes e.Report Designer Professional when processing finishes:

```
erdpro -b "c:\Corporate Reports\customers.rod" "fundmgrs.rod"
   "performance.rod"
```

- The following command builds new ROXs and ROIs for two reports in different directories. It writes processing information to a file called actuatelog.txt, then closes e.Report Designer Professional when processing finishes:

```
erdpro -r "c:\Corporate Reports\customers.rod"
   "c:\Department Reports\employees.rod" -f actuatelog.txt
```

- The following command builds new ROXs for three reports in the same directory, updates all files included with the RODs to use relative paths, writes processing information to reports.log, then closes e.Report Designer Professional when processing finishes:

```
erdpro -u c:\corpReports\customers.rod" "fundmgrs.rod"
   "performance.rod" -f reports.log
```

- The following command builds new ROXs and ROIs for two reports in the same directory, updates all files included with the RODs to use relative paths, writes processing information to reports.log, then closes e.Report Designer Professional when processing finishes:

```
erdpro -r -u "c:\myReports\sales.rod" "performance.rod" -f
   reports.log
```

abstract base class

A class that defines the requirements and behavior of descendant classes by specifying methods and variables. An abstract base class does not support the creation of instances.

Related terms
base class, class, descendant class, method, variable

Contrast with
concrete base class, object

access control list (ACL)

A list of security IDs for data in a report. If a security ID in the access control list matches the user ID or any security role of which the user is a member, the data is accessible to that user.

Related terms
data, report, security ID, security role

Contrast with
Actuate e.Reports Page Level Security option

access type

A property specifying the shareable status of a file or folder in an Encyclopedia volume. Only the owner and an Encyclopedia volume administrator can access a file or folder that has private access type. Shared access type is a prerequisite to granting privileges to other users.

Related terms
administrator, Encyclopedia volume, privilege, property

Actuate Basic

A programming language syntax-compatible with Microsoft Visual Basic 3.0. Actuate Basic consists of standard Basic functions and statements, plus object-oriented language extensions. Report developers can write Actuate Basic code to extend the functionality of e.reports.

Related terms
e.report, function, object-oriented programming, report, statement

Actuate Basic data type

See data type.

Actuate Basic report

A report created using e.Report Designer Professional and Actuate Basic technology. An Actuate Basic report is also known as an e.report.

Related terms
Actuate Basic technology, Actuate e.Report Designer Professional, e.report, report

Actuate Basic technology

A set of tools that supports the creation and deployment of Actuate Basic reports. Actuate Basic technology includes e.Report Designer Professional, the Actuate Foundation Class library, and the Actuate Basic programming language. Actuate Basic technology integrates with Actuate e.Analysis.

Related terms
Actuate Basic, Actuate Basic report, Actuate e.Analysis, Actuate e.Report Designer Professional, Actuate Foundation Class (AFC) Library

Actuate BIRT Information Object Designer

See IO Design perspective.

Actuate BIRT iServer

A stand-alone server or a cluster of servers that stores report documents and information objects in an Encyclopedia volume, manages user information, handles report requests, and analyzes and delivers report documents.

Related terms
Encyclopedia volume, information object, report, report document, request

Contrast with
Actuate BIRT iServer System, Information Console, Management Console

Actuate BIRT iServer System

An Actuate BIRT iServer including its options. Actuate BIRT iServer System has numerous available options, which are separately purchased products.

Related terms
Actuate BIRT iServer, Actuate BIRT iServer System options

Contrast with
Information Console, Management Console

Actuate BIRT iServer System options

A set of separately licensed products for Actuate BIRT iServer. Each option extends the functionality of Actuate BIRT iServer System. For example, e.Reports option enables running and viewing e.reports. e.Reports Page Level Security option enables specifying to which parts of an e.report a user has access. e.Analysis option enables analyzing the search results from an e.report. Actuate BIRT iServer System options are separately purchased products.

Related terms

Actuate BIRT iServer, Actuate BIRT iServer System, Actuate e.Analysis option, Actuate e.Reports option, Actuate e.Reports Page Level Security option

Actuate e.Analysis

A tool that supports analysis of search results from an Actuate Basic report. This tool is an additional license option and requires Actuate e.Reports option. Actuate BIRT iServer System options are separately purchased products. Figure G-1 shows an example of the initial view of Actuate e.Analysis.

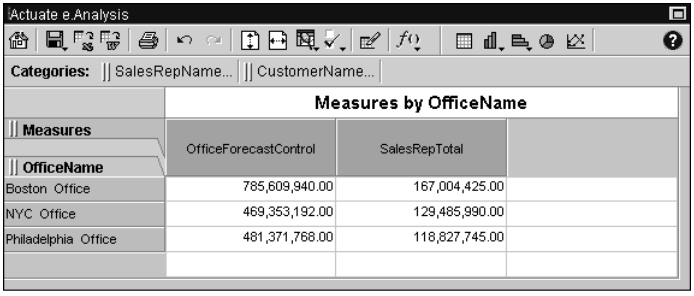


Figure G-1 The Actuate e.Analysis initial view

Related terms

Actuate Basic report, Actuate BIRT iServer System options, Actuate e.Reports option, report

Contrast with

Actuate e.Analysis option, analytics

Actuate e.Analysis option

An Actuate BIRT iServer System option that supports analysis of search results from an Actuate Basic report. This option requires Actuate e.Reports option. Actuate BIRT iServer System options are separately purchased products.

Related terms

Actuate Basic report, Actuate BIRT iServer System options, Actuate e.Reports option, report

Contrast with

Actuate e.Analysis, analytics

Actuate e.Report Designer Professional

A tool used to design and build Actuate Basic report designs and reusable components. e.Report Designer Professional also previews reports generated from the designs and distributes report object executable files to Actuate BIRT iServer. A report developer can use the Actuate Basic language and Actuate Foundation Classes to customize report designs.

Related terms
Actuate Basic, Actuate Basic report, Actuate BIRT iServer, Actuate Foundation Class (AFC), component, design, report

Contrast with
Actuate Basic technology, file types

Actuate e.Reports option

An Actuate BIRT iServer System option that supports running and viewing e.reports. This option is a prerequisite for Actuate e.Analysis option and Actuate e.Reports Page Level Security option. Actuate BIRT iServer System options are separately purchased products.

Related terms
Actuate BIRT iServer System options, Actuate e.Analysis option, Actuate e.Reports Page Level Security option, e.report

Actuate e.Reports Page Level Security option

An Actuate BIRT iServer System option that controls access to report content. A user name or security role determines access to report pages. This option is available for Actuate Basic technology. Actuate e.Reports Page Level Security option requires Actuate e.Reports option. Actuate BIRT iServer System options are separately purchased products.

Related terms
Actuate Basic technology, Actuate BIRT iServer System options, Actuate e.Reports option, page, report, security role

Contrast with
access control list (ACL), structured content

Actuate Foundation Class (AFC)

One of the building blocks of report designs. Actuate Foundation Classes include those that define charts, controls, data sources, frames, images, pages, and sections. Report developers use Actuate e.Report Designer Professional to derive classes from the Actuate Foundation Classes.

Related terms
Actuate e.Report Designer Professional, chart, class, control, data source, design, frame, image, page, report, section

Contrast with
abstract base class, Actuate Foundation Class (AFC) Library, base class, component, descendant class

Actuate Foundation Class (AFC) Library

A library that contains Actuate Foundation Classes. The AFC.rol file contains the Actuate Foundation Class Library.

Related terms
Actuate Foundation Class (AFC), class, library

Contrast with
component library

Actuate iServer

See Actuate BIRT iServer.

Actuate iServer System

See Actuate BIRT iServer System.

Actuate iServer System options

See Actuate BIRT iServer System options.

Actuate server

See Actuate BIRT iServer.

Actuate SQL

A query language based on the ANSI SQL-92 standard. Information objects encapsulate Actuate SQL queries.

Related terms
information object, query, SQL (Structured Query Language)

ad hoc parameter

A parameter associated with a database column that passes an expression to dynamically extend the query's WHERE or HAVING clause. An ad hoc parameter uses Query by Example (QBE) syntax to restrict the number of rows returned from the data source to the report. For example, in Figure G-2 the QBE value, >100, appends the expression PURCHASEVOLUME > 100 to the query's WHERE clause:

```
WHERE PURCHASEVOLUME >100
```

Related terms
column, database, data source, expression, parameter, query, Query by Example (QBE), report, row, syntax, value

Contrast with
static parameter

administrator

A user who is able to perform administrative tasks on a system or application.

1 A member of the Windows Administrators group.

2 In Actuate BIRT iServer System, a member of the Administrator security role.

Related terms
Actuate BIRT iServer System, security role

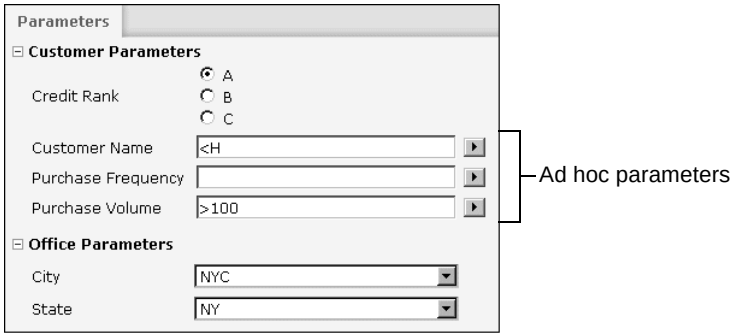


Figure G-2 Ad hoc parameters showing QBE syntax

AFC (Actuate Foundation Class)

See Actuate Foundation Class (AFC).

After A component reference property that identifies a component to print or display at the end of a section. For example, After can contain the sum of all orders from one customer. After is a slot in the layout editor and the structure. Figure G-3 shows an After slot in the Report Structure window.

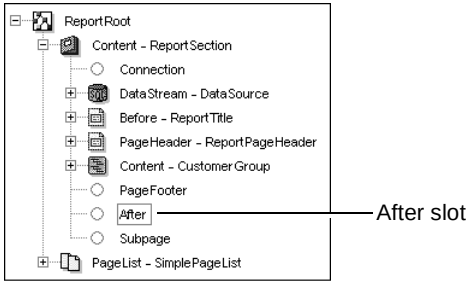


Figure G-3 After slot

Related terms
component, component reference property, layout editor, Report Structure window, section, slot

Contrast with
Before

aggregate expression

An expression that uses one or more aggregate functions to produce an aggregate value. For example, the expression, max([SPEED]), produces a value that is the maximum value of the field, SPEED, in the data rows.

Related terms
aggregate function, aggregate value, data row, expression, field, value

Contrast with
aggregate row

aggregate function

A function that performs a calculation over a set of data rows. For example, sum( ) calculates the sum of values of a specified numeric field over a set of data rows. Examples of aggregate functions include count, max, min, and sum.

Related terms
data row, field, function, value

Contrast with
aggregate row, aggregate value

aggregate row

A single row that summarizes data from a group of rows returned by a query. A SQL (Structured Query Language) query that includes an aggregate expression and a Group By clause returns one or more aggregate rows. For example, a row that totals all orders made by one customer is an aggregate row.

Related terms
aggregate expression, data, group, query, row, SQL (Structured Query Language)

Contrast with
aggregate value, data row, SQL SELECT statement

aggregate value

The result of applying an aggregate function to a set of data rows. For example, a set of data rows has a field, SPEED, which contains values: 20, 10, 30, 15, 40. The aggregate expression, max([SPEED]), produces the aggregate value, 40, which is the maximum value for the field.

Related terms
aggregate expression, aggregate function, data row, field, value

alias

An alternative name:

1. In Actuate Basic, a name given to an external procedure to avoid conflict with constants, reserved words, or variables.
2. In a SQL SELECT statement, a name given to a database table or column.

Related terms
Actuate Basic, column, constant, database, reserved word, SQL SELECT statement, table, variable

analytics

The iterative process of analyzing data to inform and plan business decisions. Analytics uses drill-down and statistical techniques to examine the same information in both detail and overview forms. Analytics tools promote business intelligence goals by supporting inspection, cleaning, and transformation of data.

Related term
data

ancestor class

A class in the inheritance hierarchy from which a particular class directly or indirectly derives.

Related terms
class, inheritance, hierarchy

Contrast with
base class, class hierarchy, descendant class, subclass, superclass

application

A complete, self-contained program that performs a specific set of related tasks.

application programming interface (API)

A set of routines, including functions, methods, and procedures, that exposes application functionality to support integration and extend applications.

Related terms
application, function, method, procedure

argument

A constant, expression, or variable that supplies data to a function, method, or subroutine.

Related terms
constant, data, expression, function, method, subroutine, variable

Contrast with
parameter

array

A data variable consisting of sequentially indexed elements that have the same data type. Each element has a common name, a common data type, and a unique index number identifier. Changes to an element of an array do not affect other elements.

Related terms
data, data type, element, variable

ascendant class

See ancestor class.

assignment statement

A statement that assigns a value to a variable. For example, in Actuate Basic:

```
StringToDisplay = "My Name"
```

Related terms
Actuate Basic, statement, value, variable

asterisk (*)

1 A wildcard character used for searches.

2 A multiplication symbol used in expressions.

Related terms
character, expression, search, symbol, wildcard

Contrast with
search expression

attribute A property of an element defined as a name-value pair. For example, in the following code, the attribute defines a uniform resource identifier (URI) that links to a web page:

```
<a href="http://www.actuate.com">
```

Related terms
element, property, uniform resource identifier (URI), web page

Contrast with
Extensible Markup Language (XML)

balloon help

A phrase created by a report developer to explain a data item in a report. Balloon help displays when the user moves the cursor onto the item.

Related terms
data, report

BAS See Basic source (.bas) file.

base chart

The part of a chart that contains the main chart data, typically, the most important data in the chart. Every chart uses a base chart. Some charts also include study charts, which display below the base chart, or data plotted on a second *y*-axis. In Figure G-4, a base chart shows the stock price trading range and a study chart shows the trading volume for each date.

Related terms
chart, data, study chart

Contrast with
dual y-axis chart

base class

In the Actuate Foundation Class Library, a class from which another class in the Actuate Foundation Class hierarchy derives. For example, Actuate Foundation Class, AcConnection, is the base class of all connection classes in a report design, as shown in Figure G-5.

Related terms
Actuate Foundation Class (AFC), Actuate Foundation Class (AFC) Library, class, design, hierarchy, report

Contrast with
abstract base class, ancestor class, concrete base class, inheritance

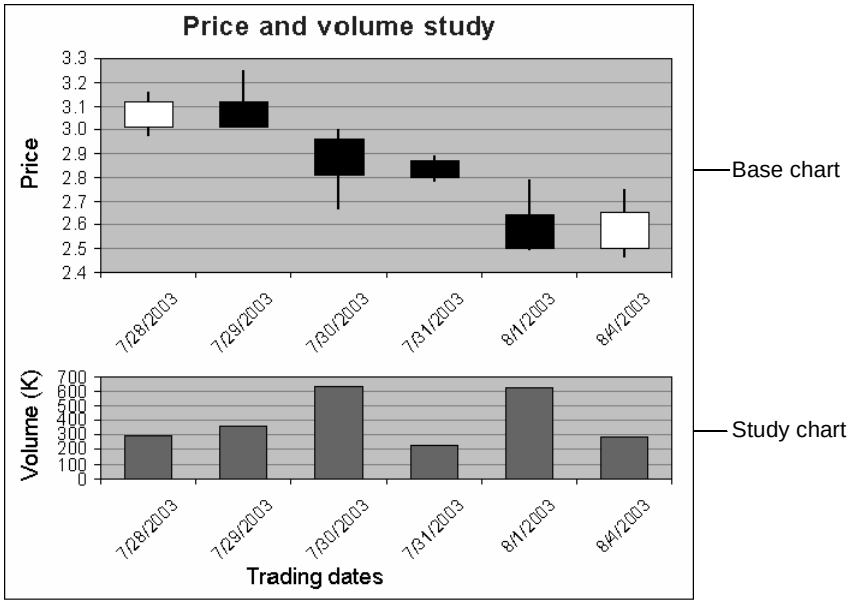


Figure G-4 Base chart with a study chart

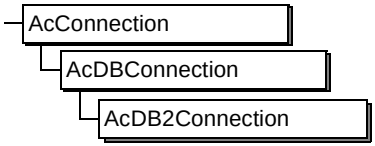


Figure G-5 Base class

base unit A unit of time displayed on a time-scale axis in a chart.

Related term
chart

Contrast with
grid, tick

Basic source (.bas) file

A file that contains Actuate Basic source code. To create a Basic source file, perform one of the following tasks:

- Compile a report object design (.rod) file, which causes generation of a Basic source (.bas) file. The factory then compiles the source file or files into a report object executable (.rox) file.
- Write a Basic source file using any text editor and save it to a file having the .bas extension.

Related terms
Actuate Basic, factory, file types, report object design (.rod) file, report object executable (.rox) file

Before

A component reference property identifying a component to be printed or displayed at the beginning of a section. For example, Before can contain column headings. Before is a slot in the layout editor and the report structure. Figure G-6 shows a Before slot in the Report Structure window.

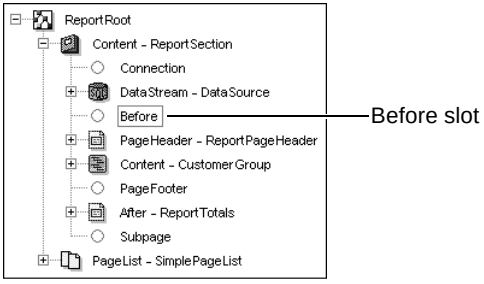


Figure G-6 Before slot

Related terms
component, component reference property, layout editor, Report Structure window, section, slot

Contrast with
After

bidirectional text

Text written in multiple languages, at least one of which reads from right-to-left (RTL) and one of which reads from left-to-right (LTR). When right-to-left text, such as Arabic, mixes with left-to-right text, such as English, in the same paragraph, each type of text is written in its own direction.

Boolean expression

An expression that evaluates to True or False. For example, Total > 3000 is a Boolean expression. If the condition is met, the condition evaluates to True. If the condition is not met, the condition evaluates to False.

Related term
expression

Contrast with
conditional expression, numeric expression

breakpoint

A place marker in a program being debugged. At a breakpoint, execution pauses so the report developer can examine problematic Actuate Basic code.

Related terms
Actuate Basic, debug

browser button

See ellipsis.

browser scripting control

A control that supports writing code inside an Actuate Basic report. The code can be in any form that a web browser can interpret, including HyperText Markup Language (HTML), Java applets, JavaScript, and VBScript (Visual Basic Script Edition). The web browser interprets the code when the user views a report. Figure G-7 shows a button and a radio set with text labels that a web browser displays based on the browser scripting control code.

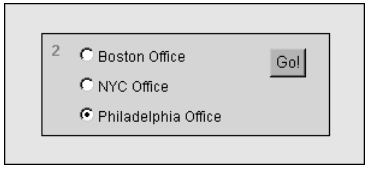


Figure G-7 Browser scripting control in a web browser

Related terms
Actuate Basic report, control, HyperText Markup Language (HTML), Java, report

Builder button

See ellipsis.

bursting See report bursting.

calculated column

See computed field.

case sensitivity

A condition in which the letter case is significant for the purposes of comparison. For example, "McManus" does not match "MCMANUS" or "mcmanus" in a case-sensitive environment.

category One of the discrete values that organizes the data on a bar chart axis. Typically, category values appear on the *x*-axis of a chart. In a pie chart, category values define which sectors appear in a pie, as shown in Figure G-8.

Related terms
chart, data, value

Contrast with
series

cell The intersection of a row and a column that displays a value in a cross tab or spreadsheet. Figure G-9 shows a cell.

Related terms
column, cross tab, row, value

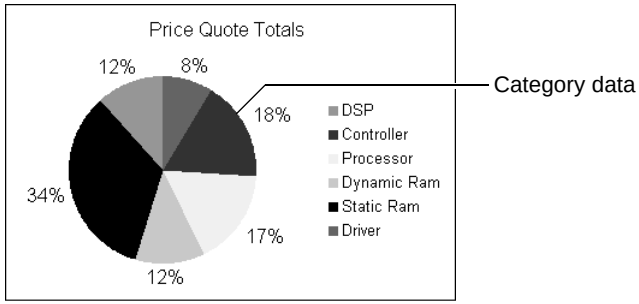


Figure G-8 Category data

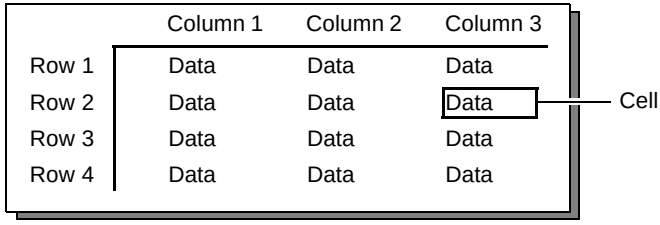


Figure G-9 Cells in a cross tab

character An elementary mark that represents data, usually in the form of a graphic spatial arrangement of connected or adjacent strokes, such as a letter or a digit. A character is independent of font size and other display properties. For example, an uppercase C is a character.

Related terms
data, font, property

Contrast with
character set, string

character set

A mapping of specific characters to code points. For example, in most character sets, the letter A maps to the hexadecimal value 0x21.

Related terms
character, code point

Contrast with
locale, locale map

chart A graphic representation of data or the relationships among sets of data, for example a bar, bubble, line, meter, pie, radar, or stock chart.

Related term
data

Contrast with
chart control

chart control

A control that displays values in the form of a chart. The relevant Actuate Foundation Class is AcChart. Figure G-10 shows a chart control in the layout editor.

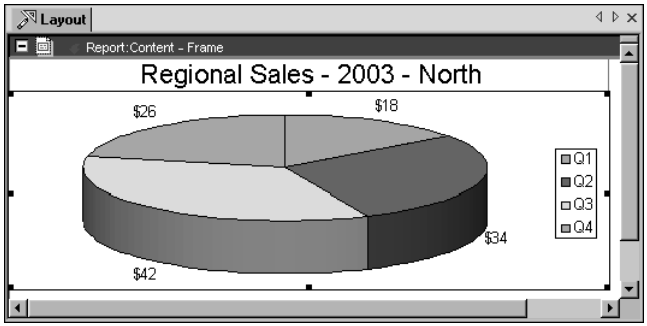


Figure G-10 Chart control in the layout editor

Related terms
Actuate Foundation Class (AFC), chart, control, layout editor, value

Contrast with
drawing control, image control

class

A set of methods and variables that defines the properties and behavior of an object. All objects of a given class are identical in form and behavior, but can contain different data in their variables.

Related terms
data, method, object, property, variable

Contrast with
Actuate Foundation Class (AFC), base class, component, subclass, superclass

class declaration

A statement that defines a class. A class declaration contains other class, method, and variable declarations.

Related terms
class, declaration, method, statement, variable

class hierarchy

A tree structure that represents the inheritance relationships among a set of classes.

Related terms
class, inheritance

Contrast with
base class

class name

A unique name for a class that permits unambiguous references to its public static methods and variables.

Related terms
class, method, static variable, variable

Class page

A page in the Properties window that displays class information about the selected component. A Class page displays information such as the superclass of the selected component, as shown in Figure G-11.

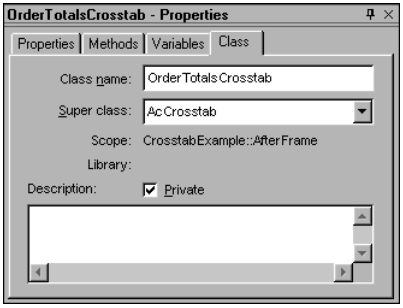


Figure G-11 Class page

Related terms
class, component, Properties window, page, superclass

Contrast with
Methods page, Properties page, Variables page

class representation (.apr) file

A file that contains class association information for the toolbox. The class association defines which component image appears in the toolbox to represent a specific class.

Related terms
class, component, file types, toolbox

class variable

A variable that all instances of a class share. An object-oriented environment makes only one copy of a class variable. The value of the class variable is the same for all instances of the class, for example, the taxRate variable in an Order class.

Related terms
class, object-oriented programming, value, variable

Clipboard

A temporary storage area that holds information to transfer from one document or application to another.

Related term
application

Contrast with
Scratch Pad

code point

A hexadecimal value in a character set. Every character in a character set is represented by a code point. The computer uses the code point to process the character.

Related terms
character, character set, value

Contrast with
locale map

color chooser

A tool that specifies colors for parts of a report such as fonts, frame backgrounds, lines, and report element backgrounds. For example, a report developer accesses the color chooser by using the ellipsis button on the Properties page. Figure G-12 shows where to access the color chooser.

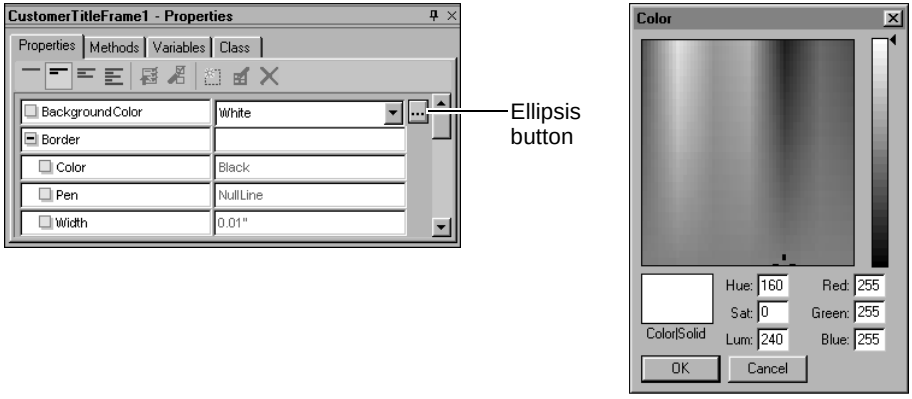


Figure G-12 Color chooser

Related terms
ellipsis, font, frame, line control, Properties page, report

column

1 A named field in a database table or query. For each data row, the column can have a different value, called the column value. The term column refers to the definition of the column, not to any particular value. Figure G-13 shows the names of columns in a database table.

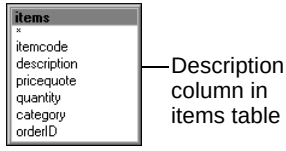


Figure G-13 Columns in a database table

2 A vertical sequence of cells in a cross tab or spreadsheet. Figure G-14 shows a column in a cross tab.

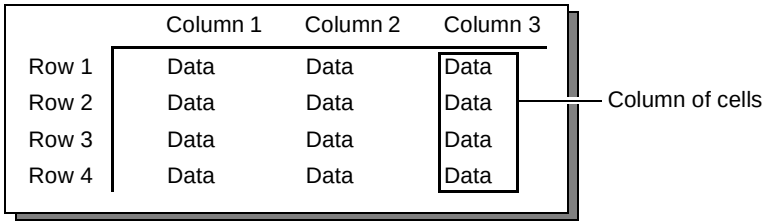


Figure G-14 Column in a cross tab

Related terms
cell, cross tab, data row, database, field, query, table, value

Contrast with
Column Editor

Column Editor

A tool used to create a column or to change column properties such as name and data type.

Related terms
column, data type, property

column key

An expression used to group data rows into columns and sub-columns in a cross-tab control.

Related terms
column, cross-tab control, data row, expression, group

Contrast with
row key

Columns page

A page in a query editor that lists columns selected for inclusion in a query. For example, Figure G-15 shows the Columns page in the query editor.

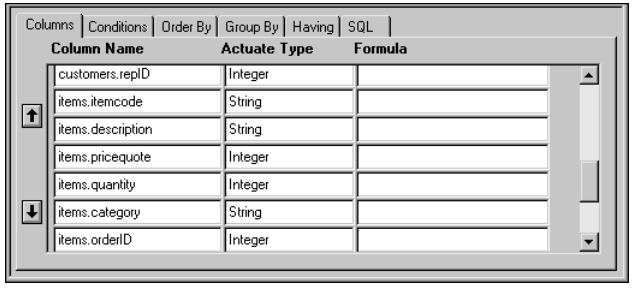


Figure G-15 Columns page

Related terms
column, page, query, query editor

comma-separated values (CSV) file

A flat file format that stores data in a tabular structure, separating the rows by new-line characters, the column values by commas, and delimiting the column values containing special characters by quotation marks.

Related terms
column, data, flat file, format, row, value

Contrast with
file types

compile

To translate code written by a programmer into object code for execution. For example, compiling an Actuate Basic source (.bas) file generates a report object executable (.rox) file.

Related terms
Basic source (.bas) file, report object executable (.rox) file

Contrast with
file types

component

A building block used to construct a report design. Components have properties and methods that define their appearance and behavior. One component can be based on or derived from another component. All components derive from AcComponent, the principal base Actuate Foundation Class.

Related terms
Actuate Foundation Class (AFC), base class, design, method, property, report

Contrast with
component library, object, toolbox, visual component

component library

A file containing components that provide consistent behavior and appearance across a suite of reports. Modifying a component in a library updates the component in all report designs that use that library. The file extension of report object library files is .rol.

Related terms
component, design, file types, library, report, report object library (.rol) file

Contrast with
Actuate Foundation Class (AFC), class

component reference property

A property that records reference relationships among components. A component reference property is an Actuate Foundation Class (AFC) property that stores the name of another component. For example, a frame Content property stores the names of controls within that frame.

Related terms
Actuate Foundation Class (AFC), component, control, frame, property

computed field

A field that displays the result of an expression.

Related terms
expression, field

concrete base class

A class created to organize a hierarchy or define methods and variables that apply to derived classes. A concrete base class supports the creation of instances.

Related terms
class, class hierarchy, method, variable

Contrast with
abstract base class, base class, object

conditional expression

n expression that returns value A or value B depending on whether a Boolean expression evaluates to True or False. For example:

```
If <condition> Then <then-statement> Else <else-statement>
```

Related terms
Boolean expression, expression, statement, value

conditional format

A format that applies to a control when a specified condition is met.

Related terms
control, format

conditional section

A section that uses a Boolean expression to determine which of two frames or sections to include in a report. Figure G-16 shows a conditional section in the Report Structure window.

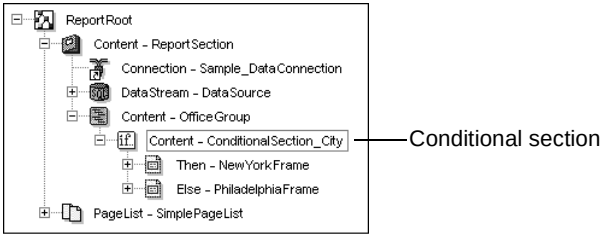


Figure G-16 Conditional section

Related terms
Boolean expression, frame, report, Report Structure window, section

Conditions page

A page in a query editor that displays selection criteria for rows that the query selects from the data source. Figure G-17 shows the Conditions page in the graphical query editor.

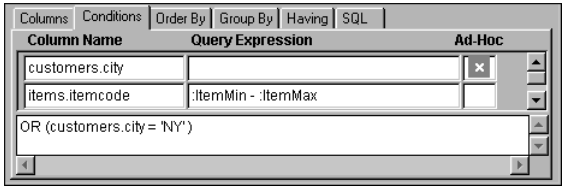


Figure G-17 Conditions page

Related terms
data source, page, query, query editor, row

Configuration Console

A tool used to configure Actuate BIRT iServer System. For example, you can configure logging and e-mail notification. In Actuate BIRT iServer, the advanced view provides more options, such as taking an Encyclopedia volume off-line.

Related terms
Actuate BIRT iServer, Actuate BIRT iServer System, Encyclopedia volume

Contrast with
Information Console, Management Console

configuration file

An Extensible Markup Language (XML) file containing the parameters and settings used to set run-time values for e.reports. For example, e.reports use the following configuration files:

1. A file that stores connection information for data sources and location information for component libraries and Actuate Basic source (.bas) files.
2. A file that specifies the ODA version of the driver and defines the contents, semantics, and structure of requests and responses between the custom data source and e.Report Designer Professional or Actuate BIRT iServer.

Related terms
Actuate Basic report, Actuate BIRT iServer, Actuate e.Report Designer Professional, Basic source (.bas) file, component library, connection, data source, open data access (ODA), open data access (ODA) driver, report

Connection

A component reference property that specifies a link to a data source. Connection is a slot in the Report Structure window. The relevant Actuate Foundation Class is AcConnection. Figure G-18 shows a populated connection slot in the Report Structure window.

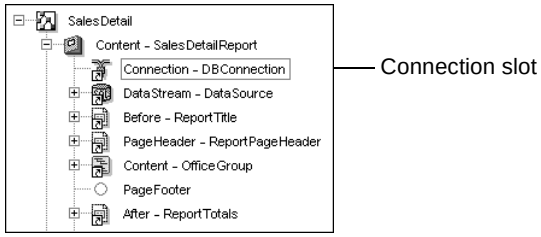


Figure G-18 Connection slot

Related terms
Actuate Foundation Class (AFC), component reference property, data source, Report Structure window, slot

Contrast with
connection, connection property

connection

A component that establishes a communication link with a data source, defined by a set of connection properties. Each of the standard connections uses a corresponding Actuate Foundation Class derived from AcConnection. Figure G-19 shows a list of connection components.

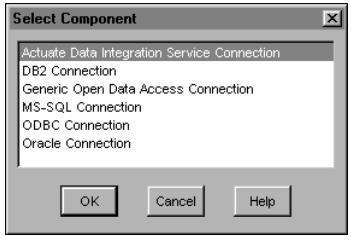


Figure G-19 Connection components

Related terms
Actuate Foundation Class (AFC), component, connection property, data source, database, property

Contrast with
Connection

connection property

A named value used to connect to a data source. The properties vary depending on the data source type. Typical connection properties are user name and password. For example, Figure G-20 shows connection properties for a Connection component.

Related terms
component, Connection, data source, password, property, value

Contrast with
configuration file

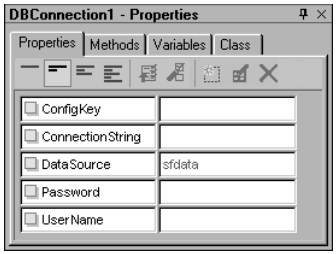


Figure G-20 Connection properties

consistent report design

A report design that compiles and executes with no errors. A design-consistency check inspects the design and any included component libraries for references to unknown classes, methods, or variables.

Related terms
class, compile, component library, design, method, reference, variable

constant

An unchanging, predefined value. A constant does not change while a program is running,.

constructor code

Code that initializes an instance of a class.

Related term
class

Contrast with
object

container

A data structure that holds one or more different types of data. For example, a frame is a container for controls.

Related terms
control, data, frame

containment

A relationship among instantiated objects in a report. One object, the container, defines the scope of other objects, the contents.

Related terms
container, instantiation, object, report, scope

containment hierarchy

A hierarchy of objects in a report. For example, a frame is a container for controls and a report section is a container for frames.

Related terms
container, control, frame, hierarchy, object, report, report section

Content A component reference property that determines which items appear in a component for every data row processed by the component. For example, the content of the innermost group section is typically a frame displaying data for each data row. Content is a slot in the layout editor and the report structure. Figure G-21 shows a frame in a Content slot in the Report Structure window.

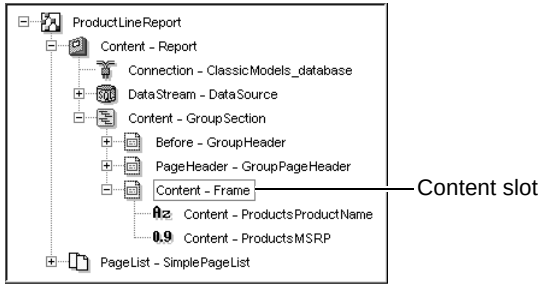


Figure G-21 Content slot containing a frame component

Related terms

component, component reference property, data, data row, frame, group section, layout editor, Report Structure window

content See structured content.

control A visual component such as a chart, image, integer, or text, that can be placed in a frame or on a page in a report design. A control can be data driven or static. Figure G-22 shows image, integer, label, and text controls in a frame in the layout editor.

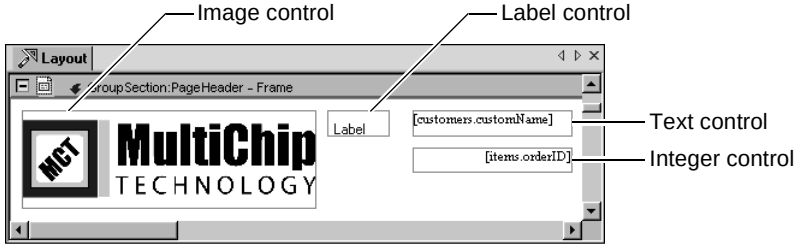


Figure G-22 Controls in a frame

Related terms

chart, chart control, component, data, data control, design, frame, image, image control, integer control, label control, layout editor, page, report, static control, text control

Contrast with

browser scripting control, cross tab, currency control, floating point control, date time control, grid, page, rectangle control

converter **1** A tool that converts data from one format to another format.

2 A tool that converts an Actuate Basic report to a viewable format such as Dynamic HyperText Markup Language (DHTML) or PDF.

Related terms
Actuate Basic report, data, Dynamic HyperText Markup Language (DHTML), format

cross tab A report that arranges data into a concise summary for analysis. Data values appear in a matrix of rows and columns. Every cell in a cross tab contains an aggregate value. A cross tab shows how one item relates to another, such as an order total aggregated by credit rating and order status. Figure G-23 shows a cross tab.

Order Totals by Credit Rating and Order Status

	A			B			C			Totals		
	Quantity	*Ave Qty*	*Amount*	*Quantity*	*Ave Qty*	*Amount*	*Quantity*	*Ave Qty*	*Amount*	*Quantity*	*Ave Qty*	*Amount*
Open	98,205	349	9,870,799	40,472	283	4,673,517	4,980	332	636,555	143,657	327	15,180,871
Closed	171,813	365	20,633,175	131,183	497	14,465,314	21,790	198	2,591,731	324,786	384	37,690,220
In Evaluation	90,969	469	12,135,326	22,168	411	2,596,936	0		0	113,137	456	14,732,262
Cancelled	23,944	855	3,029,520	0		0	0		0	23,944	855	3,029,520
Selected	10,540	527	1,375,204	0		0	0		0	10,540	527	1,375,204
Totals	395,471	398	47,044,024	193,823	420	21,735,767	26,770	214	3,228,286	616,064	390	72,008,077

Figure G-23 Cross tab displaying order totals

Related terms
aggregate value, cell, column, data, report, row, value

Contrast with
aggregate function, cross-tab control

cross-tab control

A control that displays a cross tab. The relevant Actuate Foundation Class is AcCrosstab. Figure G-24 shows a cross-tab control.

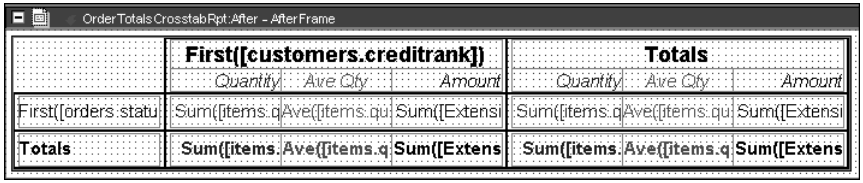


Figure G-24 Cross-tab control

Related terms
Actuate Foundation Class (AFC), control, cross tab

Contrast with
analytics, cell, grid, table

cross-tabulation

See cross tab.

currency control

A component that adds currency data to a report. A currency control must be in a flow, frame, or page. The relevant Actuate Foundation Class is AcCurrencyControl. Figure G-25 shows a currency control.

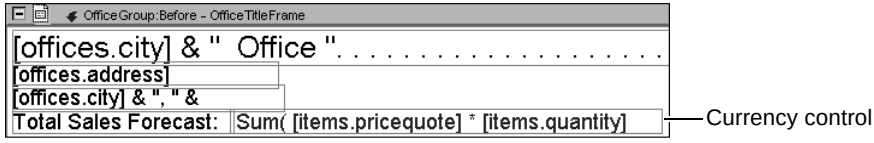


Figure G-25 Currency control displaying an expression in the layout editor

Related terms
Actuate Foundation Class (AFC), component, control, Currency data type, data, flow, frame, layout editor, page, report

Contrast with
floating point control

Currency data type

An Actuate Basic data type used for financial or fixed point calculations. The Currency data type stores numbers having up to 20 digits to the left of the decimal point and up to 9 digits to the right of the decimal point. The range of the currency data type is -39,614,081,257,132,168,796.771975167 to 39,614,081,257,132,168,796.771975167.

Related terms
Actuate Basic, data type

Contrast with
currency control, Double data type

custom data source

See open data access (ODA).

data

Information stored in databases, flat files, or other data sources.

Related terms
data source, database, flat file

Contrast with
metadata

data adapter

A data stream processing component. A data adapter filters, retrieves, sorts, or otherwise processes information. The two types of data adapters are data filters and data sources. The relevant Actuate Foundation Class is AcDataAdapter. Figure G-26 illustrates how a data adapter processes data.

Related terms
Actuate Foundation Class (AFC), component, data, data filter, data source, data stream, report, sort

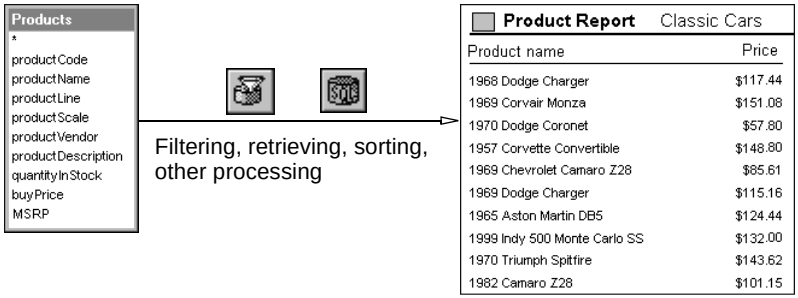


Figure G-26 Data adapter processing data field values for a report

data analysis

See analytics.

data connection definition (.dcd) file

A file created using the IO Design perspective that contains connection properties and security settings for a data source.

Related terms
connection, connection property, data source, IO Design perspective

Contrast with
file types

data control

A component that stores, then displays data. The relevant Actuate Foundation Classes are AcCurrencyControl, AcDataControl, AcDateTimeControl, AcDoubleControl, AcIntegerControl, and AcTextControl.

Related terms
Actuate Foundation Class (AFC), component, data

Contrast with
control, static control

data filter

A component that can compute new values, join data from multiple data adapters, perform custom lookup, select rows, and sort rows. A data filter is a type of data adapter. Data can pass through a series of data filters before being delivered to the report. The relevant Actuate Foundation Classes are AcDataRowBuffer, AcDataRowSorter, AcMultipleInputFilter, and AcSingleInputFilter. Figure G-27 shows data filters in a data stream.

Related terms
Actuate Foundation Class (AFC), component, data, data adapter, data stream, filter, join, report, row, sort, value

Contrast with
query, sort filter

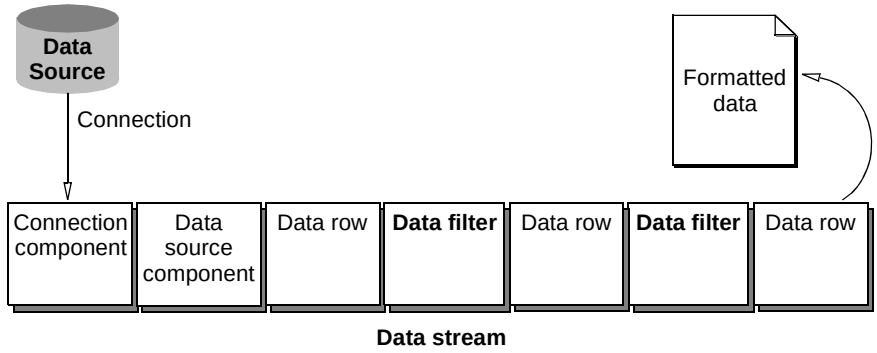


Figure G-27 Data filter components

data label Text that describes a data point in a chart. A data label line connects the data label to its data point.

Related terms
chart, data, data point

data point A point on a chart that corresponds to a particular pair of *x*- and *y*-axis values.

Related terms
chart, value

Contrast with
data label, data row

data row A component that contains and describes the output of a data adapter. That output can be used as a source of data for either a report or another data adapter. A data row provides data values as a list of variables. The relevant Actuate Foundation Class is AcDataRow. Figure G-28 shows data row components in a data stream.

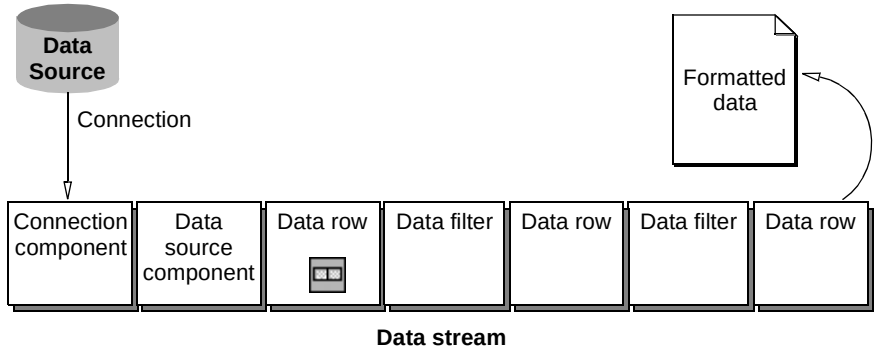


Figure G-28 Data row components in a data stream

Related terms
Actuate Foundation Class (AFC), component, data, data adapter, data source, data stream, report, row, value, variable

Contrast with
data point, DataRow, filter

Data Row Editor

A tool used to display, modify, and sort available data rows, as shown in Figure G-29.

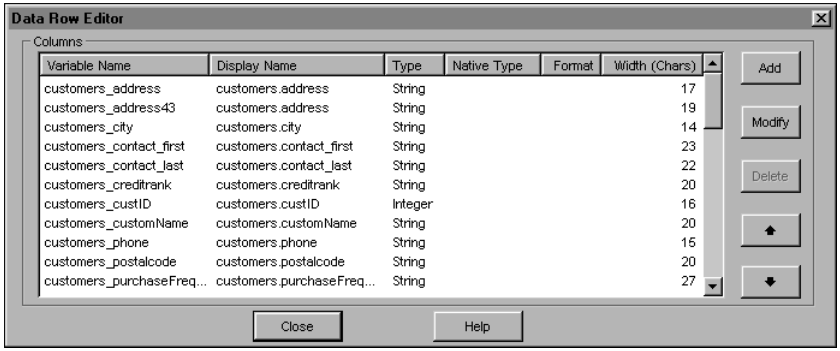


Figure G-29 Data Row Editor

Related terms
data row, sort

Contrast with
DataRow

data source

A relational database or other data repository. For example, an Extensible Markup Language (XML) file, a flat file, an information object, or a Java application can be a data source. A report can include any of these types of data. This data source provides data rows to a report through a data source component.

Related terms
application, data, data row, data source component, database, Extensible Markup Language (XML), flat file, information object, Java, report

data source component

A type of data adapter that retrieves data from a data source. The relevant Actuate Foundation Class is AcDataSource.

Related terms
Actuate Foundation Class (AFC), component, data, data adapter, data source

Contrast with
data row

data source map (.sma) file

A file created using the IO Design perspective that contains a representation of a database table, a database view, or a query written in the database's native SQL

(Structured Query Language). An information object developer uses a data source map to create an information object.

Related terms
data source, database, file types, information object, IO Design perspective, query, SQL (Structured Query Language), table, view

data stream

A sequence of one or more data adapters that collects, processes, and delivers data as input to the report layout components. Figure G-30 shows data adapters in a data stream.

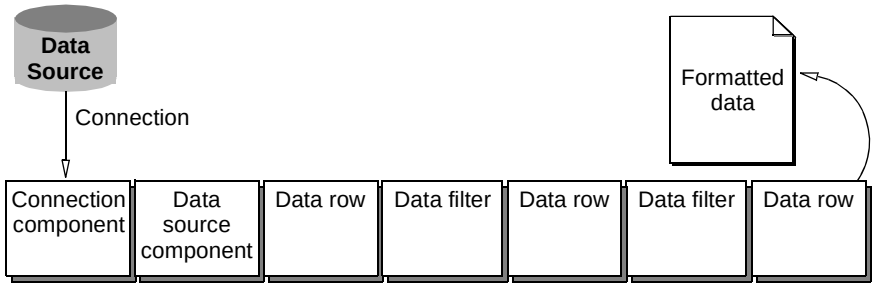


Figure G-30 Data stream

Related terms
component, data, data adapter, data filter, data row, filter, layout

Contrast with
DataRow, DataStream

data type The structure of a value that constrains its characteristics, such as the information the values can hold and permitted operations. In report development, three processes use data types: accessing data, internal processing of data, and formatting output as a report.

Data types in Actuate Basic include Currency, Date, Double, Integer, Long, Single, String, and Variant. Data source data types map to Actuate Basic data types.

Related terms
Actuate Basic, Currency data type, data source, Date data type, Double data type, Integer data type, Long data type, Single data type, String data type, value, Variant data type

Contrast with
semantic type

database An integrated collection of logically related records that provides data for information application platforms. The database model most commonly used is the relational model. Other typical models are entity-relationship, hierarchical, network, object, and object-relational.

Related terms
application, data

database connection

See connection.

database management system (DBMS)

An application that controls the storage, retrieval, and manipulation of data in a data store.

Related terms
application, data

database schema

See schema.

DataRow

A component reference property that contains and describes the output of a data adapter. That output can be used as a source of data for either a data adapter or a report section. DataRow is a slot in the Report Structure window, as shown in Figure G-31. The relevant Actuate Foundation Class is AcDataRow.

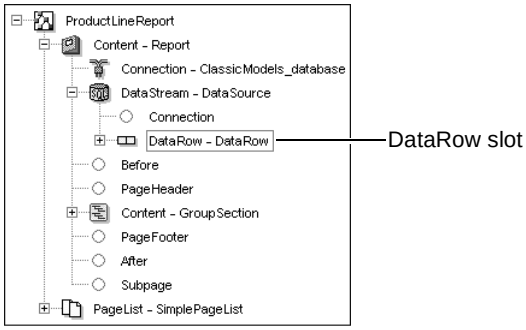


Figure G-31 DataRow slot

Related terms
Actuate Foundation Class (AFC), component reference property, data, data adapter, report section, Report Structure window, slot

Contrast with
Connection, data row, Data Row Editor, DataStream

DataStream

A component reference property that identifies a sequence of one or more data adapters that collects, processes, and delivers data as input to the report layout components. DataStream is a slot in the Report Structure window, as shown in Figure G-32. The relevant Actuate Foundation Classes, such as AcSqlQuerySource and AcDataRowBuffer, derive from AcDataAdapter.

Related terms
Actuate Foundation Class (AFC), component, component reference property, data, data adapter, layout, report, Report Structure window, slot

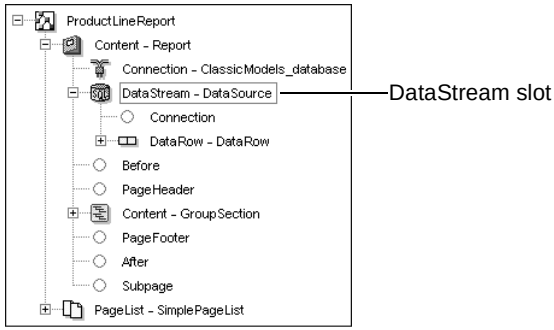


Figure G-32 DataStream slot

Contrast with
Connection, DataRow

Date data type

An Actuate Basic data type used for date-and-time calculations. The Date data type stores floating point numbers that represent dates ranging from 1 January 100 to 31 December 9999 and times from 00:00:00 to 23:59:59.

Related terms
Actuate Basic, data type

date time control

A component that adds date-and-time data to a report. A date time control must be in a flow, frame, or page. The relevant Actuate Foundation Class is AcDateTimeControl. Figure G-33 shows a date time control in a frame in the layout editor.

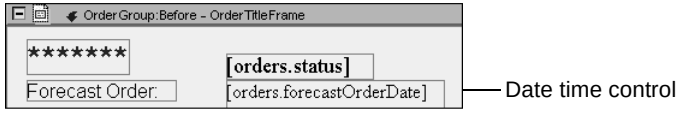


Figure G-33 Date time control placed in a frame

Related terms
Actuate Foundation Class (AFC), component, control, data, flow, frame, layout editor, page, report

Contrast with
Date data type

DBMS (database management system)

See database management system (DBMS).

DCD

See data connection definition (.dcd) file.

debug To detect, locate, and fix errors in a computer program. Typically, debugging involves executing specific portions of the program and analyzing the operation of those portions.

declaration

The definition of a class, constant, method, or variable that specifies the name and, if appropriate, the data type.

Related terms
class, constant, data type, method, variable

Contrast with
class declaration

declarations section

That portion of Actuate Basic code that contains constant, data type, and global variable declarations. Declare...End Declare encloses the declarations section.

Related terms
Actuate Basic, constant, data type, declaration, variable

delete privilege

See privilege.

dependency

See file dependency.

derived class

See descendant class.

descendant class

A class that extends another class to provide additional functionality.

Related term
class

Contrast with
Actuate Foundation Class (AFC), ancestor class, base class, subclass, superclass

design A report specification or the act of creating a report specification. Designing a report includes selecting data, laying out the report visually, and saving the layout in a report design file.

Related terms
data, layout, report

Contrast with
file types

Design Editor

A page used to specify options for the layout editor and Report Structure window. Figure G-34 shows the Design Editor page.

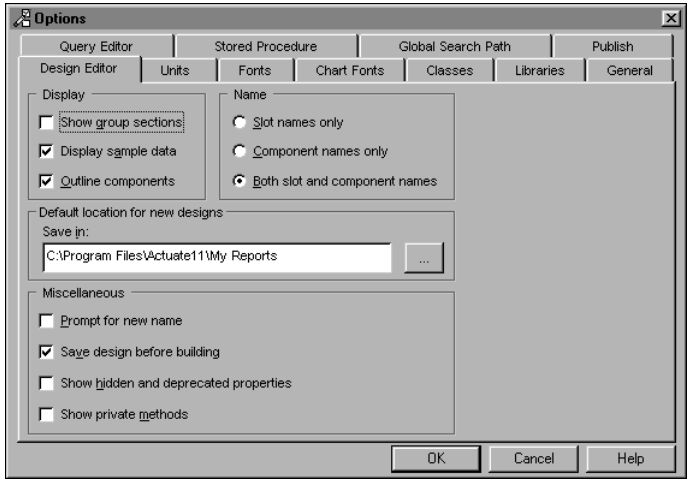


Figure G-34 Design Editor page

Related terms
layout editor, page, Report Structure window

design time

The period of time in which a report developer creates a report specification.

Related term
report

Contrast with
design, run time, view time

detail frame

A frame displayed for each data row provided by the data stream. A detail frame is in the innermost Content slot of a report or group section.

Related terms
Content, data row, data stream, frame, group section, report section, slot

DHTML (Dynamic Hypertext Markup Language)

See Dynamic HyperText Markup Language (DHTML).

DHTML report

An e.report in Dynamic HyperText Markup Language (DHTML) format viewed in the DHTML Viewer. DHTML reports use a combination of cascading style sheets, HyperText Markup Language (HTML), and scripts. e.reports are also available in Excel, PDF, PowerPoint, and RTF formats.

Related terms
DHTML Viewer, Dynamic HyperText Markup Language (DHTML), e.report, format, HyperText Markup Language (HTML)

Contrast with
XML report, report

DHTML Viewer

In e.Report Designer Professional, Information Console, and Management Console, a tool that supports viewing the output of a report in Dynamic HyperText Markup Language (DHTML) format, as shown in Figure G-35.

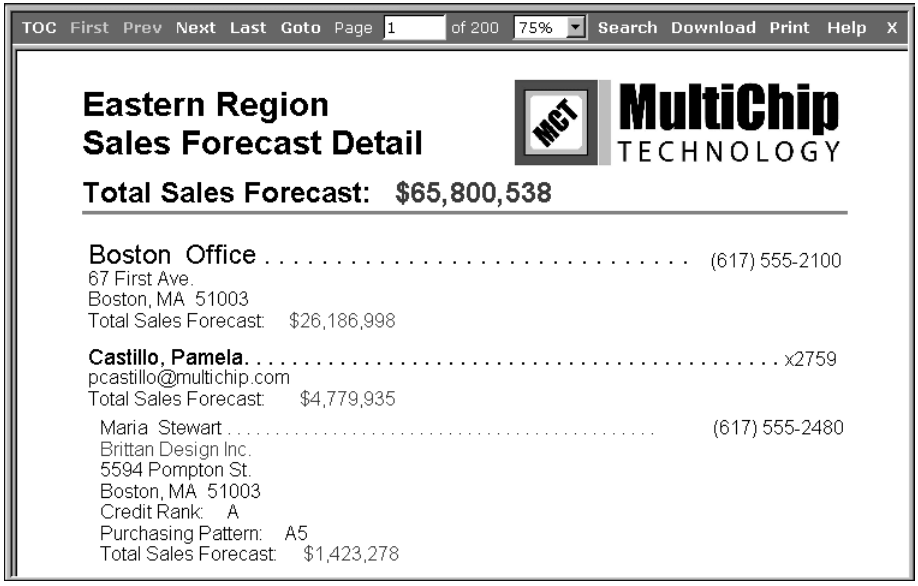


Figure G-35 DHTML Viewer

Related terms
Actuate e.Report Designer Professional, Dynamic HyperText Markup Language (DHTML), format, Information Console, Management Console, report

DLL (dynamic link library)

See dynamic link library (DLL).

document type definition (DTD)

A set of Extensible Markup Language (XML) elements and attributes that defines a schema describing the structure of an XML document.

Related terms
attribute, element, Extensible Markup Language (XML), schema

Contrast with
structured content

dot notation

Actuate Basic syntax that refers to the methods and variables of an object. The dot separates the name of an object handle from the name of an instance variable or method in the referenced object, for example, customerNameLabel.Text.

Related terms
Actuate Basic, method, object, syntax, variable

Contrast with
scope resolution operator (::)

Double data type

An eight-byte Actuate Basic data type that stores floating point numbers, ranging in value from -1.797693134862315E08 to -2.23E-308 for negative values and 2.23E-308 to 1.797693134862315E08 for positive values.

Related terms
Actuate Basic, data type, value

Contrast with
Integer data type, Single data type

drawing control

A control that uses SVG code to create an arbitrary image in a report design. The relevant Actuate Foundation Class is AcDrawing. Figure G-36 shows a drawing control in the layout editor.

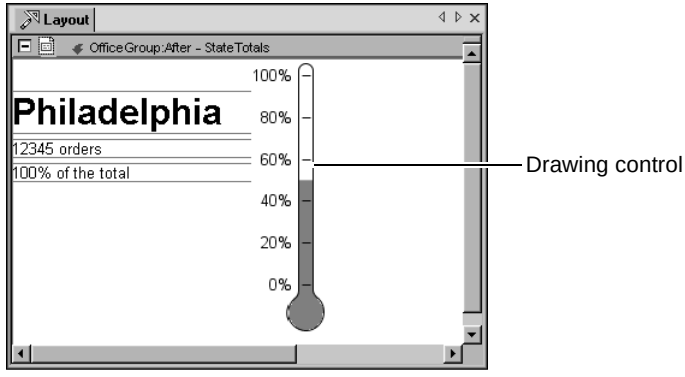


Figure G-36 Drawing control

Related terms
Actuate Foundation Class (AFC), control, design, image, layout editor, report

Contrast with
chart control, image control, line control, rectangle control

driver

An interface that supports communication between an application and another application or a peripheral device such as a printer. To communicate with databases, e.Report Designer Professional uses drivers, such as open data access (ODA) and open database connectivity (ODBC) drivers.

Related terms
Actuate e.Report Designer Professional, application, database, interface, open data access (ODA) driver, open database connectivity (ODBC)

DTD (document type definition)

See document type definition (DTD).

dual *y*-axis chart

A chart that uses two independent *y*-axes, as shown in Figure G-37. When one *y*-axis displays a different type of series from the other *y*-axis, the chart is called a combination chart. The relevant Actuate Foundation Class is AcChart.

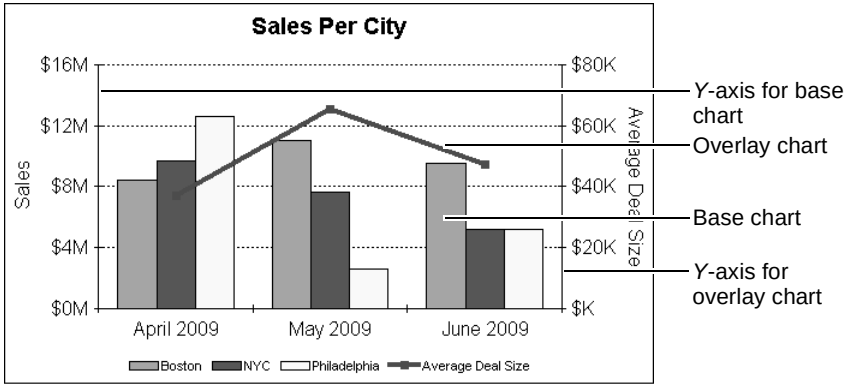


Figure G-37 Dual *y*-axis chart

Related terms
Actuate Foundation Class (AFC), base chart, chart, series

Contrast with
study chart

dynamic hyperlink

A connection from one part of a report to another part of a different report. The dynamic hyperlink first generates a new report, then links the source report to the destination report.

Related terms
hyperlink, report

Contrast with
static hyperlink

Dynamic HyperText Markup Language (DHTML)

A HyperText Markup Language (HTML) extension providing enhanced viewing capabilities and interactivity in a web page. The Document Object Model (DOM) Group of the World Wide Web Consortium (W3C) develops DHTML standards.

Related terms
HyperText Markup Language (HTML), web page

dynamic link library (DLL)

A library of routines loaded and linked into an application at run time in the Windows environment. These routines can be called from an Actuate Basic report.

Related terms
Actuate Basic, application, library, report, run time

Contrast with
shared library

dynamic page size

The ability to adjust page size based on the page's content.

Related term
page

Contrast with
structured content

dynamic text control

A control that adjusts its size to display varying amounts of HyperText Markup Language (HTML), plain, or RTF text. The relevant Actuate Foundation Class is AcDynamicTextControl. Figure G-38 shows a dynamic text control in the layout editor.

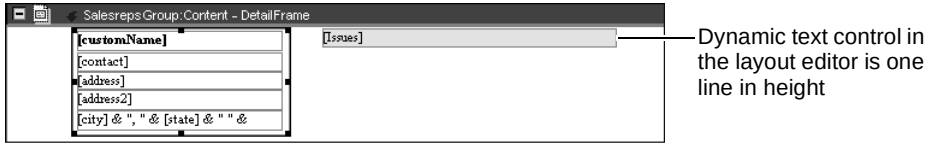


Figure G-38 Dynamic text control in the layout editor

Figure G-39 shows a dynamic text control in a generated report.

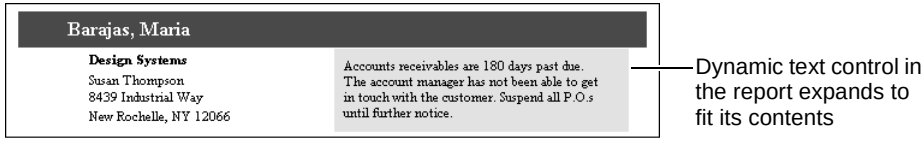


Figure G-39 A report displaying text in a dynamic text control

Related terms
Actuate Foundation Class (AFC), control, HyperText Markup Language (HTML), layout editor, report

Contrast with
text control

e.Analysis option

See Actuate e.Analysis option.

element A logical structure in an Extensible Markup Language (XML) or HyperText Markup Language (HTML) document specifying a type and optionally one or more attributes and a value. For example, the following code specifies a ConnectionParam element that has three attributes, Name, Display, and Type, and no value:

```
<ConnectionParam Name="username"
     Display="User name"
     Type="string"
  />
```

Related terms
attribute, Extensible Markup Language (XML), HyperText Markup Language (HTML), value

ellipsis

A button that opens tools that you use to perform tasks, such as navigating to a file.

embed

1. To insert an object in a report. For example, a report developer can embed an image in a report object design (.rod) file.
2. To encapsulate information in a report. For example, for each font used in a report object design (.rod) file, e.Report Designer Professional embeds information about that font in the generated report object executable (.rox) file.

Related terms
Actuate e.Report Designer Professional, design, font, image, object, report, report object design (.rod) file, report object executable (.rox) file

Contrast with
encapsulation

empty In Actuate Basic, a value that indicates a variable has not been initialized. Empty equals zero in a numeric context, zero-length in a string context, or null in the case of an object handle.

Related terms
Actuate Basic, null, object, string, value, variable

encapsulation

A technique that bundles related functions and subroutines. Encapsulation compartmentalizes the structure and behavior of a class so that parts of an object-oriented system do not depend upon or affect each other's internal details.

Related terms
class, function, object, object-oriented programming, subroutine

Encyclopedia volume

An Actuate BIRT iServer System repository for managing data and metadata. Encyclopedia volume data includes objects such as designs and documents, stored as files in partitions.

Related terms
Actuate BIRT iServer System, data, design, repository

e.report

A structured document viewable in a web browser that follows a set of rules to organize, present, and summarize data. A sample e.report appears in Figure G-40.

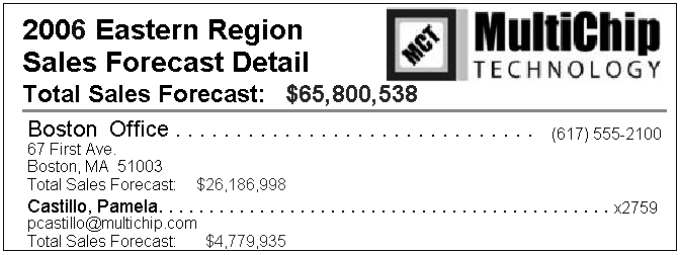


Figure G-40 A sample e.report

Related terms
Actuate Basic report, data

Contrast with
report, structured content

e.Report Designer Professional

See Actuate e.Report Designer Professional.

e.Report option

See Actuate e.Reports option.

e.reporting

An Actuate Basic technology that draws data from one or more data sources and presents it as structured content.

Related terms
Actuate Basic technology, data, data source, structured content

escape character (\)

1. A character that precedes a multibyte character or non-printing ASCII character in a search expression. For example, \t represents the Tab character.
2. A character indicating when to take a special character literally, as in a Query by Example (QBE) expression. For example, a backslash followed by a comma represents a comma character: \,

Related terms
character, Query by Example (QBE), search expression

event An action external to a program that requires handling, such as a mouse click. An event handler in the program collects information about the event and responds.

Related term
event handler

event handler

A function or method that executes when an event occurs. Report items and data sources have event handlers for which a developer can provide code.

Related terms
data source, event, function, method, report

exception An abnormal situation that a program encounters. The program handles some exceptions and returns a message to the user or application running the program. In other cases, the program cannot handle the exception, and the program ends.

Related term
application

executable file

A file that generates report output when run in an Encyclopedia volume or a report designer. For example, a report object executable (.rox) file generates report output.

Related terms
Encyclopedia volume, file types, report, report object executable (.rox) file

expression

A combination of constants, functions, literal values, names of fields, and operators that evaluate to a single value.

Related terms
constant, field, function, operator, value

Contrast with
aggregate expression, expression builder

expression builder

A tool for selecting data fields, functions, and operators to write expressions. Figure G-41 shows the expression builder.

Related terms
data, expression, field, function, operator

Extensible Markup Language (XML)

A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the Worldwide Web Consortium (W3C). XML is widely used for the representation of data structures.

Related terms
data, format

Contrast with
Dynamic HyperText Markup Language (DHTML), HyperText Markup Language (HTML)

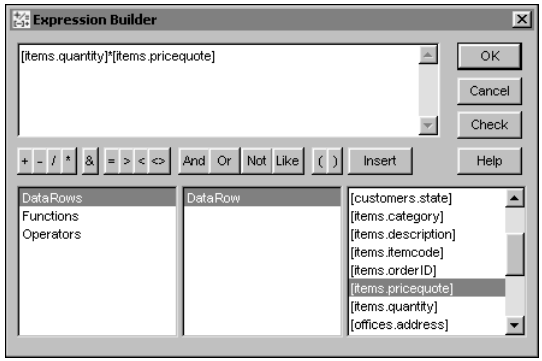


Figure G-41 Expression builder

factory A process that generates a report document from a report executable file.

Related terms
executable file, report document

Contrast with
file types

fetch To retrieve the next data row in a result set. For example, a data adapter, such as a data source or data filter, fetches data rows. Figure G-42 shows data adapters and data rows in a data stream.

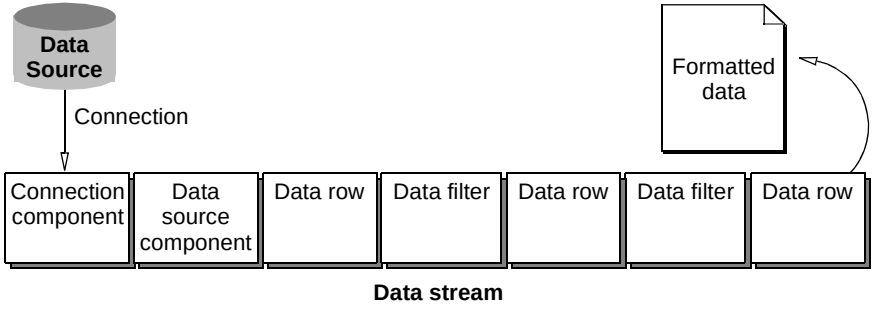


Figure G-42 Fetching data rows

Related terms
component, connection, data, data adapter, data filter, data row, data source, data stream, result set

field The smallest identifiable part of a database table. In a relational database, a field is also called a column.

Related terms
column, database, table

Fields A tool that shows all the fields in the data row returned by a data stream. Figure G-43 shows Fields.

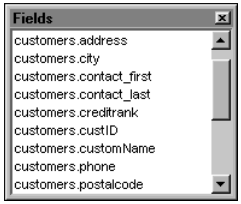


Figure G-43 Fields tool

Related terms
data row, data stream, field

file dependency

A relationship between files in which one file requires another file to compile or run. For example, in the Encyclopedia volume, both the report object instance (.roi) file and report object executable (.rox) file must be present to view an e.report.

Related terms
compile, Encyclopedia volume, e.report, report, report object executable (.rox) file, report object instance (.roi) file

Contrast with
file types

file types Table G-1 lists the file types related to working with e.Report Designer Professional.

Table G-1 File types

Management Console display name	File extension	Icon
Actuate Basic Source File	BAS	
Actuate Design Library	ROL	
Actuate Information Object	IOB	
Actuate Parameter Template	ROP	
Actuate Report Design	ROD	
Actuate Report Document	ROI	
Actuate Report Executable	ROX	

Table G-1 File types

Management Console display name	File extension	Icon
Actuate Report Parameter Values	ROV	
Actuate Search Definition	ROS	
Adobe PDF File	PDF	
Data Connection Definition	DCD	
Data Source Map	SMA	
HTML document	HTM	
HTML document	HTML	
Microsoft PowerPoint Presentation	PPT	
Microsoft PowerPoint Presentation (2007)	PPTX	
Microsoft Word Document	DOC	
Microsoft Word Document (2007)	DOCX	

Related terms
Actuate e.Report Designer Professional, Basic source (.bas) file, data connection definition (.dcd) file, data source map (.sma) file, information object, information object (.iob) file, Management Console, report object design (.rod) file, report object executable (.rox) file, report object instance (.roi) file, report object library (.rol) file, report object parameter (.rop) file, report object value (.rov) file, search definition (.ros) file

filter

A mechanism that enables a user to reduce the number of items in a list. For example, in Information Console and Management Console, filter options appear above the list of contents. Figure G-44 shows filter options in Management Console.

In e.Report Designer Professional, filter options appear on Methods, Project, Properties, and Variables as shown in Figure G-45.

Related terms
Actuate e.Report Designer Professional, Information Console, Management Console, Methods page, Project browser, Properties page, Variables page

Contrast with
data filter, sort filter

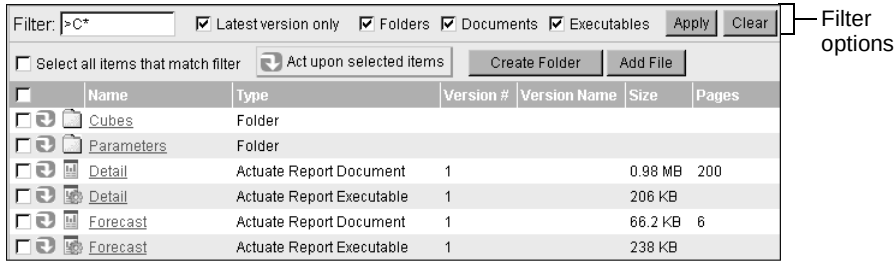


Figure G-44 Filter options In Management Console

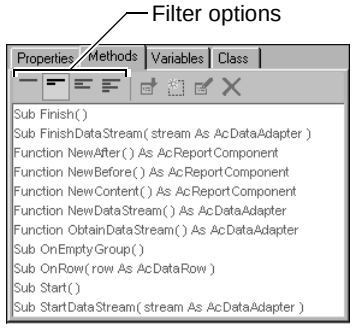


Figure G-45 Filter options on the Methods page

Flash object

Flash is an animation software package from Adobe Software. It supports the creation of lightweight animated graphics on web pages. e.Report Designer Professional supports placing a Flash object into a browser scripting control. Viewing a Flash object requires the installation of the Flash plug-in for the user's web browser.

Related terms
Actuate e.Report Designer Professional, browser scripting control, package, web page

flat file

A file that contains data in the form of text.

Related term
data

Contrast with
data source

floating point control

A component that adds data of Double data type to a report. A floating point control must be in a flow, frame, or page. The relevant Actuate Foundation Class is AcDoubleControl. Figure G-46 shows a floating point control in the layout editor and in the Report Structure window.

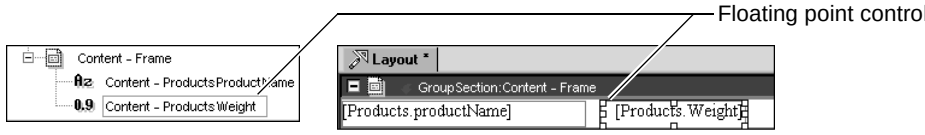


Figure G-46 Floating point control in the report structure and layout editor

Related terms
Actuate Foundation Class (AFC), component, control, data, Double data type, flow, frame, layout editor, page, report, Report Structure window

Contrast with
currency control, integer control

flow A component that defines a printable area of a page. Flows determine the columns and margins of a page. The relevant Actuate Foundation Class is AcFlow. Figure G-47 shows three flows on a page.

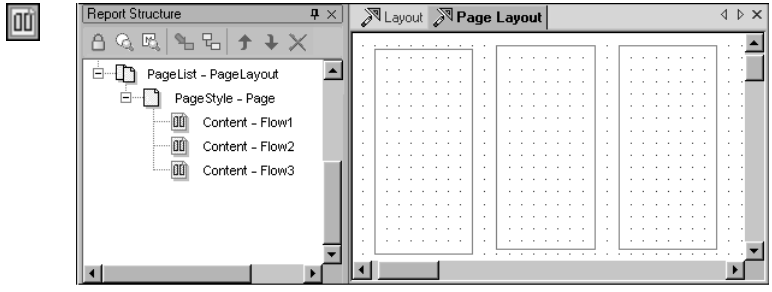


Figure G-47 Multiple flows in the report structure and layout editor

Related terms
Actuate Foundation Class (AFC), column, component, layout editor, page, Report Structure window

Contrast with
frame

font A family of characters of a given style. A font contains information that specifies posture, typeface, type size, and weight.

Related term
character

footer A logically separate unit of information that appears after the main body of content, for example date created or page number.

Contrast with
header

format

1. A specification that describes layout and properties of rich information, such as HyperText Markup Language (HTML), PDF, PostScript, PowerPoint, or RTF.

2 A set of standard options with which to display and print currency values, dates, numbers, strings, and times.

Related terms
HyperText Markup Language (HTML), layout, property, string, value

Contrast with
format toolbar, locale map, style

format toolbar

A toolbar used to change text properties. The properties available on the format toolbar include alignment, font, font color, font size, and font style. Figure G-48 shows the format toolbar.

Figure G-48 Format toolbar

Related terms
font, property, toolbar

frame

A rectangular container for graphics and text. For example, a frame in the layout editor contains image and text controls, as shown in Figure G-49.

Figure G-49 Frame

Related terms
container, control, image control, layout editor, text control

Contrast with
HTML frame

framework

A set of interrelated classes that provide an architecture for building an application, such as the Actuate Foundation Classes.

Related terms
Actuate Foundation Class (AFC), application, class

function

A code module containing a set of instructions that operate as a subroutine in a program. To invoke the function, include its name as an instruction anywhere in the program. An Actuate Basic function returns a value.

Related terms
Actuate Basic, subroutine, value

Contrast with
method

global name space

A list of names of items available throughout a report design. The list can include names of classes, functions, and variables.

Related terms
class, function, report object design, variable

Contrast with
scope resolution operator (::)

global variable

A variable available at all levels in an application. A global variable stays in memory in the scope of all executing subroutines until the application terminates.

Related terms
application, scope, subroutine, variable

grandchild class

See descendant class.

grandparent class

See ancestor class.

grid

1 A vertical and horizontal pattern that assists a report developer in positioning design components accurately. A grid is an option in the layout editor.

2 Demarcation lines that extend from axis tick marks of a chart.

Related terms
chart, component, design, layout editor, tick

group

A set of data rows organized by one or more common values. For example, in a sales report, a group consists of all the orders placed by a single customer.

Related terms
data row, report, value

Contrast with
group key, grouped report

Group By page

A page in the query editor that displays definitions of aggregate rows, as shown in Figure G-50.

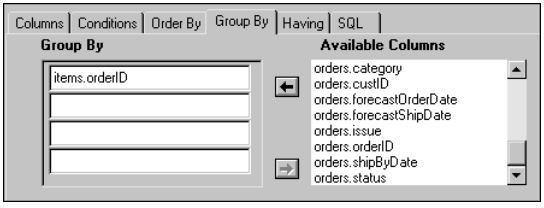


Figure G-50 Group By page

Related terms
aggregate row, group, page, query editor, row

group footer

See footer.

group header

See header.

group key An expression that groups and sorts data. For example, a report developer can group and sort customers by credit rank.

Related terms
data, expression, group, sort

group section

A component that organizes rows using a common field. For example, customers can be grouped by state. The relevant Actuate Foundation Class is AcGroupSection. Figure G-51 shows group sections in the Report Structure window.

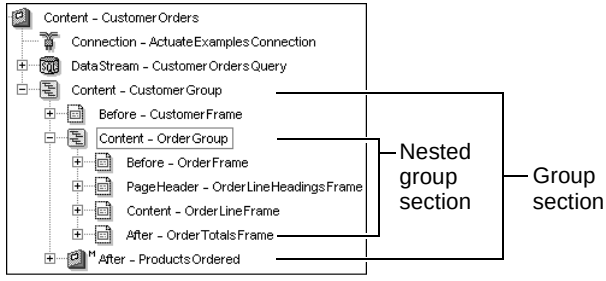


Figure G-51 Group sections

Related terms
Actuate Foundation Class (AFC), component, field, group, Report Structure window, row, section

grouped report

A report that organizes data by common values. Figure G-52 shows a grouped report organized by customer name.

Related terms
data, report, value

Contrast with
group

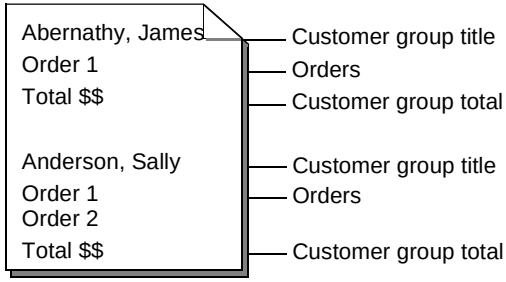


Figure G-52 Grouped report

Having page

A page in the query editor that displays selection criteria for aggregate rows, as shown in Figure G-53.

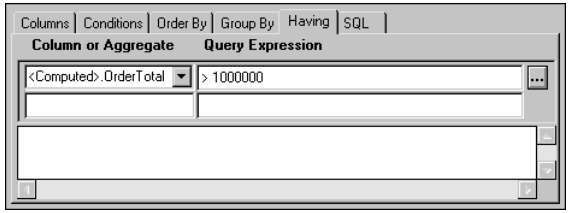


Figure G-53 Having page

Related terms
aggregate row, page, query editor

header A logically separate unit of information that appears before the main body of content. For example, a page header typically contains a document title. A group header typically contains key information about the group. For example, a group header in a sales report can contain the country name.

Related terms
group, page

Contrast with
footer

help See balloon help.

hexadecimal number

A number in base 16. A hexadecimal number uses the digits 0 through 9 and the letters A through F. Each place represents a power of 16. By comparison, base 10 numbers use the digits 0 through 9. Each place represents a power of 10.

Contrast with
character set

hierarchy Any tree structure that has a root and branches that do not converge. Figure G-54 shows an example hierarchy of classes.

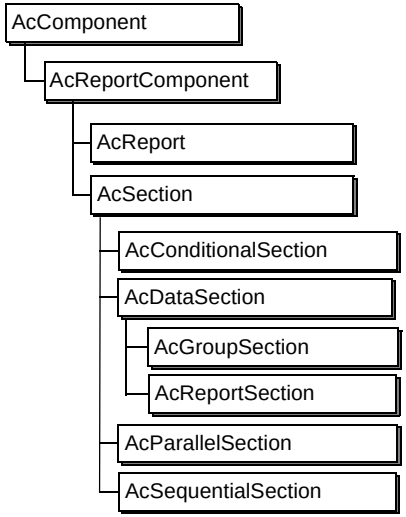


Figure G-54 Hierarchy of classes

Related term
class

home folder

A path and folder name that is the default user working environment in an Encyclopedia volume or file system repository.

Related terms
Encyclopedia volume, repository

HTML

See HyperText Markup Language (HTML).

HTML frame

An independently scrollable part of a web page dividing the page into horizontal or vertical sections. Figure G-55 shows structured content in an HTML frame.

Related terms
HyperText Markup Language (HTML), structured content, web page

Contrast with
frame

HTTP

See HyperText Transfer Protocol (HTTP).

hyperchart

A chart that supports linking to report data or other related information. For example, a pie chart segment representing the sales amount for the Boston office links to the report data for that office.

Related terms
chart, data, report

Contrast with
hyperlink

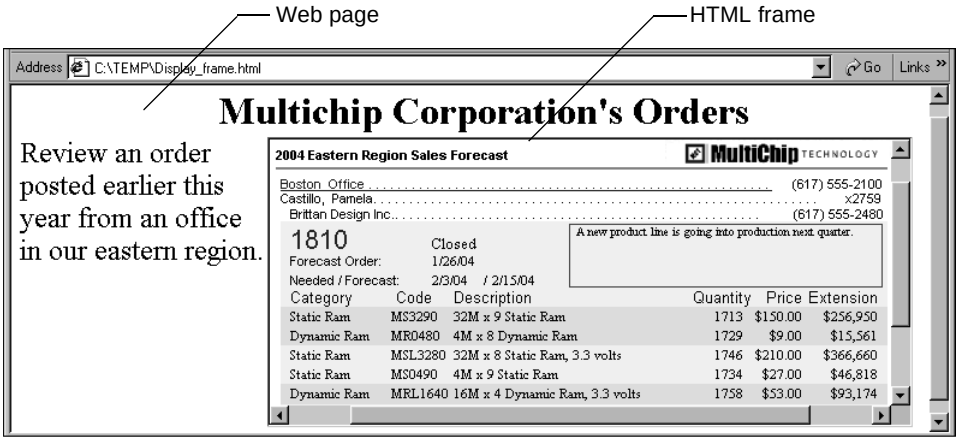


Figure G-55 HTML frame in a web page

hyperlink

An active connection in an online document that supports access to related information in the same document or an external source. The document can be an e-mail, PDF, report, or web page. A change from the standard cursor shape to a cursor shaped like a hand indicates a hyperlink.

Related terms
report, web page

Contrast with
dynamic hyperlink, hyperchart

HyperText Markup Language (HTML)

A standards-based specification that determines the layout of a web page. HTML is the markup language that a web browser parses to display a web page. The World Wide Web Consortium (W3C) specifies the standard for HTML.

Related terms
layout, web page

Contrast with
Dynamic HyperText Markup Language (DHTML), Extensible Markup Language (XML)

HyperText Markup Language page

See web page.

HyperText Transfer Protocol (HTTP)

A standard that supports request-response communication between two applications on a network. The World Wide Web Consortium (W3C) specifies the standard for HTTP.

Related terms
application, request

Contrast with
protocol

identifier

A name assigned to an item in a program, for example a class, function, or variable.

Related terms
class, function, variable

image

A graphic that appears in a report. Actuate Basic reports support .bmp, .jpg, .pcx, .tga, .tif and .tiff files and offer limited support for .gif images.

Related terms
Actuate Basic report, report

Contrast with
image control

image control

A control that adds a graphic image to a report design. The relevant Actuate Foundation Class is AcImageControl. Figure G-56 shows an image control in the layout editor.

Figure G-56 Image control in the layout editor

Related terms
Actuate Foundation Class (AFC), control, design, image, layout editor, report

Contrast with
chart control, drawing control, rectangle control

implicit declaration

A variable declaration in an assignment statement rather than in the Dim, Global, or Static statements. Visual Basic supports implicit declaration; Actuate Basic does not.

Related terms
Actuate Basic, declaration, statement, variable

Information Console

An Actuate BIRT iServer component that supports running and viewing reports stored in an Encyclopedia volume.

Related terms
Actuate BIRT iServer, Encyclopedia volume, report

information object

A named SQL (Structured Query Language) query that simplifies access to one or more heterogeneous data sources. An information object retrieves data using database tables and views, stored procedures, and ODA data source queries as well as other information objects. A data modeler writes the query in Actuate SQL. The integration service generates a native query for each data source and retrieves the data. A report developer or business user who uses the information object as a data source does not require detailed knowledge of the underlying data sources.

Related terms
Actuate SQL, data, data source, database, open data access (ODA), query, SQL (Structured Query Language), stored procedure, table, view

Contrast with
information object (.iob) file, IO Design perspective

information object (.iob) file

A file type that contains an Actuate SQL query. Data modelers create information objects using the IO Design perspective.

Related terms
Actuate SQL, file types, information object, IO Design perspective, query

inheritance

A mechanism whereby one class of objects can be defined as a special case of a more general class and includes the method and variable definitions of the general class, known as a base or superclass. The superclass serves as the baseline for the appearance and behavior of the descendant class, which is also known as a subclass. In the subclass, the appearance, behavior, and structure can be customized without affecting the superclass. Figure G-57 shows an example of inheritance.

Related terms
base class, class, descendant class, method, object, subclass, superclass, variable

Contrast with
abstract base class, hierarchy, object-oriented programming

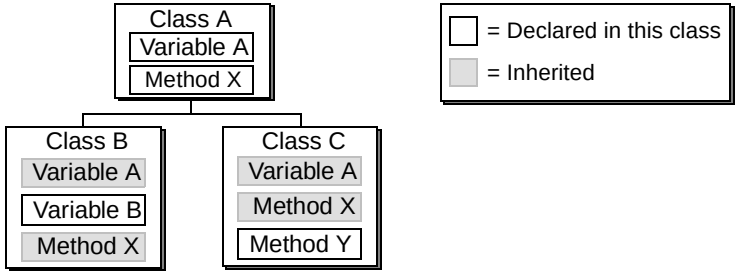


Figure G-57 Inheritance

inner join A type of join that returns records from two tables using matching values in the join field. Records for which there is no matching value are not included in the result set. For example, joining customer and order tables where the customer IDs are equal produces a result set that excludes records for customers who have no orders.

Related terms
field, join, result set, table, value

Contrast with
outer join

input source

See data source.

instance See object.

instance variable

A variable that other instances of a class do not share. The run-time system creates a new copy of an instance variable each time the system instantiates the class. An instance variable can contain a different value in each instance of a class, for example, the customerID variable in a Customer class. In Actuate Basic, all instance variables defined for a class are publicly visible. Private instance variables are not visible in subclasses.

Related terms
class, run time, subclass, value, variable

instantiation

In object-oriented programming, the process of creating an object in a run-time environment based on the class definition.

Related terms
class, object, object-oriented programming, run time

integer control

A component that adds Integer data to a report. An integer control must be in a flow, frame, or page. The relevant Actuate Foundation Class is AcIntegerControl.

Figure G-58 shows an integer control in a frame in the layout editor and in the Report Structure window.

Figure G-58 Integer control in the report structure and layout editor

Related terms
Actuate Foundation Class (AFC), component, control, flow, frame, Integer data type, page, report, Report Structure window

Contrast with
floating point control

Integer data type

1. An Actuate Basic data type that stores whole numbers, ranging in value from -2,147,483,648 to 2,147,483,647.
2. An Extensible Markup Language (XML) data type derived from the decimal data type by fixing the value of scale to be 0.

Related terms
Actuate Basic, data type, Extensible Markup Language (XML), value

Contrast with
Currency data type, Double data type, Long data type, Single data type

interface

A software component that supports access to computer resources. For example, in Java, a set of methods that provides a mechanism for classes to communicate in order to execute particular actions.

Related terms
class, Java, method

internationalization

The process of designing an application to work correctly in multiple locales.

Related terms
application, locale

Contrast with
localization

IO Design perspective

A tool used to design information objects. Information objects use Actuate SQL syntax to retrieve data from one or more data sources. Report developers use information objects in report designs.

Related terms
Actuate SQL, data, data source, information object, query, SQL (Structured Query Language)

Contrast with
file types

IOB See information object (.iob) file.

iPortal See Information Console.

iServer See Actuate BIRT iServer.

J2EE See Java Platform Enterprise Edition (Java EE).

J2SE See Java Platform Standard Edition (Java SE).

JAR See Java archive (.jar) file.

Java An object-oriented programming language.

Related terms
object-oriented programming

Contrast with
JavaServer Page (JSP)

Java 2 Enterprise Edition (J2EE)

See Java Platform Enterprise Edition (Java EE).

Java 2 Runtime Standard Edition (J2SE)

See Java Platform Standard Edition (Java SE).

Java archive (.jar) file

A compressed file format used to deploy Java applications.

Related terms
application, Java

Java Database Connectivity (JDBC)

A standard protocol that Java uses to access databases in a platform-independent manner.

Related terms
database, database connection, Java, protocol

Contrast with
database management system (DBMS), open database connectivity (ODBC), schema

Java Development Kit (JDK)

A software development kit that defines the application programming interfaces (API) used to build Java applications. As well as software tools, the kit contains documentation and examples.

Related terms
application, application programming interface (API), Java

Contrast with
Java Platform Enterprise Edition (Java EE), Java Platform Standard Edition (Java SE), JavaServer Page (JSP)

Java Naming and Directory Interface (JNDI)

An application programming interface (API) that provides unified access to named components and directory services in an enterprise system.

Related terms
application programming interface (API)

Java Native Interface (JNI)

A standard protocol that supports interaction between C or C++ applications and Java applications. Using JNI supports use of Java classes, exceptions, methods, and objects by C or C++ applications.

Related terms
application, class, Java, method, object, protocol

Java Object Interface (JOI)

A protocol that supports access to a Java object using Actuate Basic.

Related terms
Actuate Basic, Java, object, protocol

Java Platform Enterprise Edition (Java EE)

A platform-independent development environment that includes application programming interfaces (API), such as Java Database Connectivity (JDBC), Remote Method Invocation (RMI), and web services. A programmer uses Java EE to develop a highly scalable, fault-tolerant, web-based application.

Related terms
application, application programming interface (API), Java Database Connectivity (JDBC)

Contrast with
Java Development Kit (JDK), Java Platform Standard Edition (Java SE), Java Virtual Machine (JVM)

Java Platform Standard Edition (Java SE)

A smaller-scale, platform-independent development environment defining the Java programming language and application programming interfaces (API) supporting interaction with file systems, networks, and graphical interfaces. A programmer uses Java SE to develop an application to run on a virtual machine.

Related terms
application, application programming interface (API), Java

Contrast with
Java Development Kit (JDK), Java Platform Enterprise Edition (Java EE), Java Virtual Machine (JVM)

Java Virtual Machine (JVM)

The Java SDK interpreter that converts Java bytecode into machine language for execution in a specified software and hardware configuration.

Related terms
Java

JavaScript

An interpreted, platform-independent, scripting language used to embed additional processing in a web page or server.

Related terms
web page, web server

Contrast with
Java

JavaServer Page (JSP)

A standard Java extension that supports the generation of dynamic web pages. A JavaServer Page combines HyperText Markup Language (HTML) and JSP tags in one document. A servlet container interprets a JSP tag as a call to a Java class. The servlet container compiles the Java classes to generate a web page.

Related terms
class, container, HyperText Markup Language (HTML), Java, tag, web page

JDK See Java Development Kit (JDK).

JNDI See Java Naming and Directory Interface (JNDI).

JNI See Java Native Interface (JNI).

join A SQL (Structured Query Language) query operation that combines records from two tables and returns them in a result set based on the values in the join fields. Without additional qualification, join usually refers to the join in which field values are equal. For example, customer and order tables are joined on a common field such as customer ID. The result set contains combined customer and order records in which the customer IDs are equal. Figure G-59 shows joins in the query editor.

Related terms
field, query, query editor, result set, SQL (Structured Query Language), table, value

Contrast with
inner join, join condition, outer join, SQL SELECT statement

join condition

A condition that specifies a match in the values of related fields in two tables. Typically, the values are equal. Figure G-59 shows a join condition where the customer ID value in the first table equals the customer ID value in the second table.

Related terms
field, join, table, value

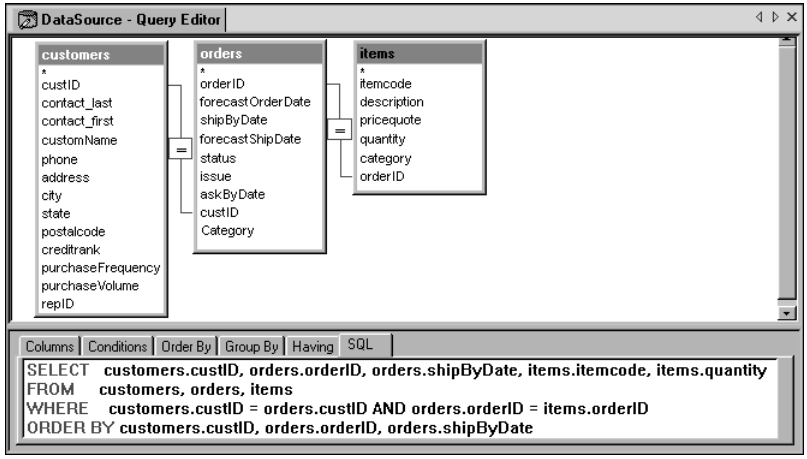


Figure G-59 Joins between customers, orders, and items tables

JSP See JavaServer Page (JSP).

JVM See Java Virtual Machine (JVM).

keyword A reserved word that is recognized as part of a programming language. For example, If, Then, and Loop are Actuate Basic keywords.

Related term
Actuate Basic

label control

A component that adds static text to an Actuate Basic report design. For example, a label control can be a heading above a column of data. A label control must be in a flow, frame, or page. The relevant Actuate Foundation Class is AcLabelControl. Figure G-60 shows label controls used as column headings placed in a frame in the layout editor.

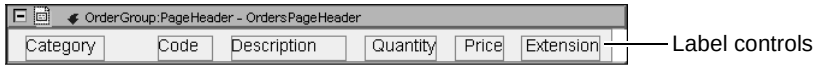


Figure G-60 Label controls in a frame

Related terms
Actuate Basic technology, Actuate Foundation Class (AFC), column, component, control, data, design, flow, frame, layout editor, page, report

Contrast with
text control

layout The designed appearance of a report. Designing a report entails arranging controls on a page so that a report user can analyze the information easily. A

report displays information in a combination of charts, footers, headers, paragraphs, subreports, and tabular lists. Figure G-61 shows a report layout.

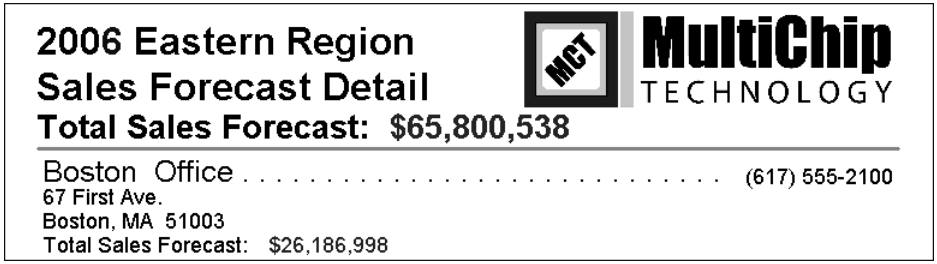


Figure G-61 Report layout

Related terms
chart, control, footer, header, page, report, subreport

layout editor

A tool in a report designer in which a report developer arranges, formats, and sizes controls. Figure G-62 shows the layout editor in e.Report Designer Professional.

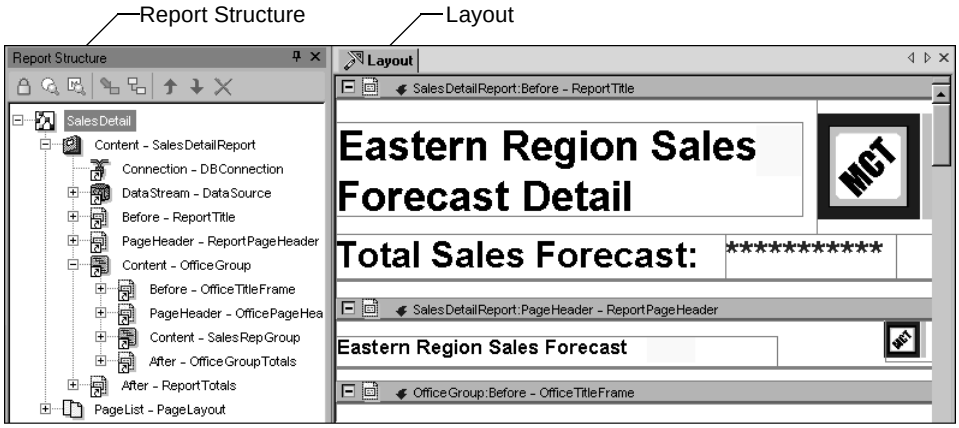


Figure G-62 Layout editor

Related terms
control, design, format, report

Layout window

See layout editor.

left outer join

See outer join.

libraries view

See libraries viewer.

libraries viewer

A tool that lists the components contained in a library. Figure G-63 shows the libraries viewer in e.Report Designer Professional.

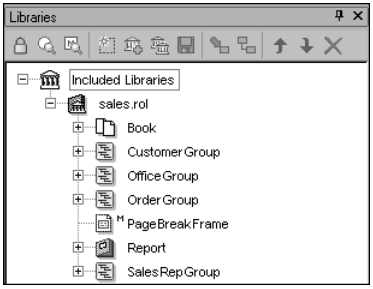


Figure G-63 Libraries viewer

Related terms
component, library

Libraries window

See libraries viewer.

library

1 A file used when creating or running a program. For example, Windows library files are dynamic link libraries. UNIX library files are shared libraries.

2 A file that contains published components or Actuate Basic functions or global variables.

Related terms
Actuate Basic, component, dynamic link library (DLL), function, publish, variable

Contrast with
component library, file types

library file

See report object library (.rol) file.

Library Organizer

A tool that a report developer uses to include or remove a report object library (.rol) file or an Actuate Basic source (.bas) file from a report design, and to create a new library. Figure G-64 shows Library Organizer.

Related terms
Basic source (.bas) file, design, library, report, report object library (.rol) file

Contrast with
file types

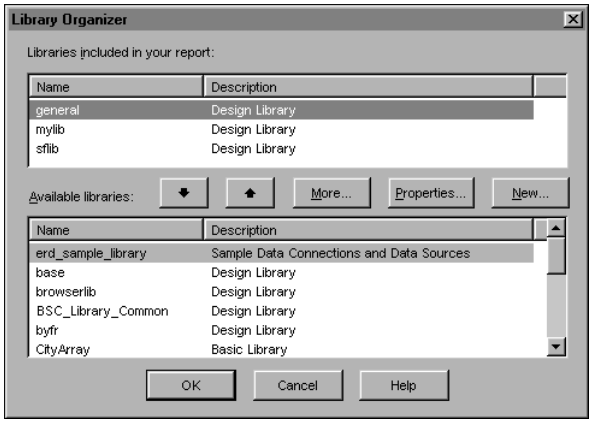


Figure G-64 Library files displayed in Library Organizer

line control

A component that draws a line in a report design. Line controls are contained in a flow, frame, or page. The relevant Actuate Foundation Class is AcLineControl.

Related terms
Actuate Foundation Class (AFC), component, design, flow, frame, page, report, report object design

link

See hyperlink.

listing report

A report that provides a simple view of data. Typically, a listing report displays a single line for each data row. Figure G-65 shows a listing report.

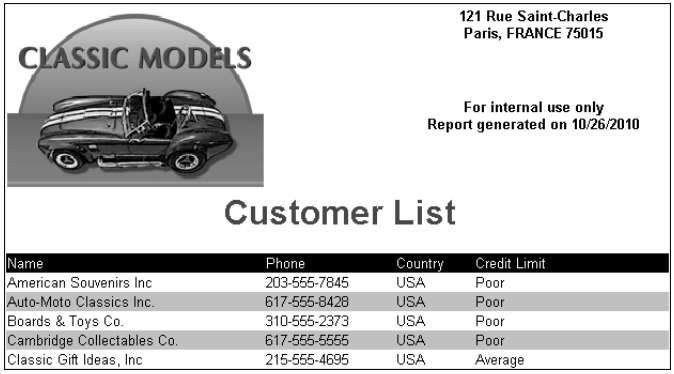


Figure G-65 Listing report

Related terms
data, data row, report

local variable

A variable that is available only at the current level in an application. A local variable stays in memory in the scope of an executing procedure until the procedure terminates. When the procedure finishes, the run-time system destroys the variable and returns the memory to the system.

Related terms
application, procedure, run time, scope, variable

locale

A location and the currency format, date format, language, sorting sequence, time format, and other characteristics associated with that location. The location is not always identical to the country. There can be multiple languages and locales within one country. For example, China has two locales: Beijing and Hong Kong. Canada has two language-based locales: French and English.

Related term
format

Contrast with
locale map, localization

locale map

For Actuate Basic reports, an Extensible Markup Language (XML) file that specifies the currency, date, number, and time formats for supported locales. For Configuration Console, Information Console and Management Console, an XML file that specifies the date, number, and time formats for supported locales. For example, some of the supported locales include Arabic (Bahrain), Chinese (Hong Kong SAR), Chinese (PRC), Chinese (Taiwan), French (France), German (Germany), Hebrew, Japanese, Korean, Spanish (Spain), and Thai.

Related terms
Actuate Basic report, Configuration Console, Extensible Markup Language (XML), Information Console, locale, Management Console

localization

The process of translating database content, printed documents, and software programs into another language. Report developers localize static text in a report so that the report displays text in another language that is appropriate to the locale the user chooses when he logs in to Information Console.

Related terms
database, Information Console, locale, report

Contrast with
internationalization

Long data type

1 An Actuate Basic data type that stores 4-byte integers, ranging in value from -2,147,483,648 to 2,147,483,647.

2 An Extensible Markup Language (XML) data type having a range from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.

Related terms
Actuate Basic, data type, Extensible Markup Language (XML), range, value

Contrast with
Double data type, Integer data type

lookahead aggregate

An expression requiring two passes through the data rows in a data source to calculate a value. For example, to calculate the percentage of a total sales amount that each product category represents requires one pass to calculate the total sales amount for all product categories. A second pass calculates the total sales amount for each product category then compares that value to the total for all categories to compute a percentage value.

Related terms
data row, data source, expression, value

Contrast with
running aggregate

Management Console

A set of web pages that provide volume management functions, such as creating security roles and users for Encyclopedia volumes.

Related terms
Encyclopedia volume, security role, web page

Contrast with
Configuration Console, Information Console

mailing label report

A report that formats name and address information to be printed on address forms such as Avery® labels.

Related term
report

Contrast with
multicolumn report

map See data source map (.sma) file or locale map.

matrix See cross tab.

member A method or variable defined in a class. A member provides or uses information about the state of a single object.

Related terms
class, method, object, state, variable

Contrast with
global variable, instance variable, static variable

member variable

A declared variable within a class. The member variables for an object contain its data or state.

Related terms
class, data, declaration, object, state, variable

metadata Information about the structure of data enabling a program to process information. For example, a relational database stores metadata that describes the data type, name, and size of objects in a database, such as tables and columns.

Related terms
column, data, data type, database, object, table

method A routine that provides functionality to an object or a class.

Related terms
class, object

Contrast with
data, function, overloaded method

method editor

A tool used to write code for Actuate Basic methods, as shown in Figure G-66.

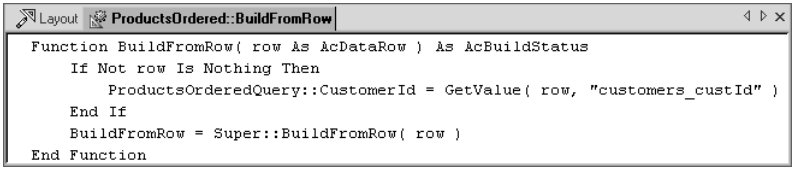


Figure G-66 Method editor

Related terms
Actuate Basic, method

Methods page

A page in the Properties window that displays the methods for a component, as shown in Figure G-67. From the Methods page, the user accesses the method editor to override methods or create new methods.

Related terms
component, method, method editor, page, Properties window

modal window

A window that retains focus until explicitly closed by the user. Typically, dialog boxes and message windows are modal. For example, an error message dialog box remains on the screen until the user responds.

Contrast with
modeless window

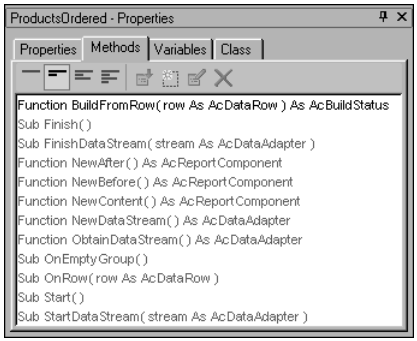


Figure G-67 Methods page

mode An operational state of a system. Mode implies that there are at least two possible states. Typically, there are many modes for both hardware and software.

modeless window

A window that solicits input but permits users to continue using the current application without closing the modeless window. For example, the Properties window is a modeless window.

Related terms
application, Properties window

Contrast with
modal window

module A file that contains class, type, or variable declarations for an application. A module is a unit of code that serves as a building block for a program.

Related terms
application, class, data type, declaration, variable

multicolumn report

A report that displays data in more than one vertical flow. A mailing label report is one type of multicolumn report. Figure G-68 shows a multicolumn report.

Figure G-68 A multicolumn report

Related terms
data, flow, mailing label report, report

multidimensional data

Any set of records that you can break down or filter according to the contents of individual fields or dimensions, such as location, product, or time interval.

Related terms
data, field, filter

Contrast with
analytics

multithreaded application

An application that handles multiple simultaneous sessions and users.

Related term
application

name space

See global name space.

navigation bar

In the DHTML Viewer, a tool that provides download, navigation, print, search, and table of contents.

Related terms
DHTML Viewer, search, table of contents

nested report

See subreport.

node

An individual part of an Extensible Markup Language (XML) document, such as an element, comment, or text string. A node tree is a hierarchical representation of the entire XML document.

Related terms
element, Extensible Markup Language (XML), hierarchy, string

Contrast with
dot notation

null

A value indicating that a variable or field contains no data.

Related terms
data, field, value, variable

numeric expression

A numeric constant, a simple numeric variable, a scalar reference to a numeric array, a numeric-valued function reference, or a sequence of these items, separated by numeric operators and parentheses. In Actuate Basic, for example:

```
Val(Dates) + 1
```

Related terms
Actuate Basic, array, constant, function, operator, variable

Contrast with
Boolean expression

object An instance of a particular class, including its characteristics. An object has properties and methods. For example, when a user runs an Actuate Basic report, the Factory process instantiates multiple persistent customer frame objects, one for each customer row retrieved from the data source.

Related terms
Actuate Basic report, class, data source, frame, method, persistent object, property, row

Contrast with
information object, transient object

object reference variable

A variable that contains a reference to an object. References can be passed to functions and subroutines as parameters.

Related terms
function, object, parameter, subroutine, variable

object-oriented programming

A paradigm for writing applications using classes, not algorithms, as the fundamental building blocks. The design methodology uses four main concepts: abstraction, encapsulation, inheritance, and polymorphism.

Related terms
application, class, encapsulation, inheritance

Contrast with
object

ODA See open data access (ODA).

ODBC See open database connectivity (ODBC).

open data access (ODA)

A technology that handles communication between a data source and an application. ODA provides interfaces for creating data drivers to establish connections, access metadata, and execute queries to retrieve data. ODA also provides interfaces to integrate query builder tools within an application designer tool. e.Report Designer Professional uses Actuate Basic technology to implement ODA.

Related terms
Actuate Basic technology, Actuate e.Report Designer Professional, application, connection, data, data source, interface, metadata, open data access (ODA) driver, query

open data access (ODA) driver

An ODA driver communicates between a data source and an application. An ODA driver establishes a connection to a data source, accesses metadata about the data, and executes queries on the data source. e.Report Designer Professional uses Actuate Basic technology to implement ODA drivers.

Related terms
Actuate Basic technology, Actuate e.Report Designer Professional, application, data, data source, driver, metadata, open data access (ODA), query

open database connectivity (ODBC)

A standard protocol used by software products as a database management system (DBMS) interface to connect applications and reports to databases.

Related terms
application, database, database management system (DBMS), interface, protocol, report

Contrast with
Connection, data source, Java Database Connectivity (JDBC)

operator

A symbol or keyword that performs an operation in an expression.

Related terms
expression, keyword

Order By page

A page in the query editor that displays sort keys, as shown in Figure G-69. You can use the Order By page instead of or in addition to the default behavior of group sections to specify how to sort data.

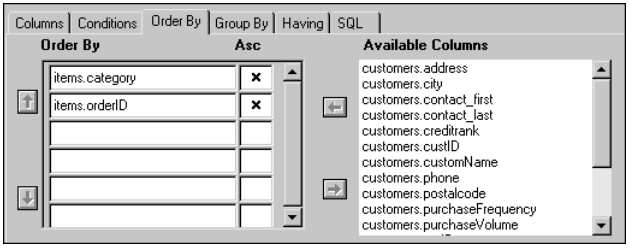


Figure G-69 Order By page

Related terms
data, group section, page, query editor, sort, sort key

outer join

A type of join that returns records from one table even when no matching values exist in the other table. The three types of outer join are left, right, and full outer join. A left outer join returns all records from the table on the left side of the join expression, even if no matching values exist in the table on the right side. A right outer join returns all records from the table on the right side of the join expression, even if no matching values exist in the table on the left side. For example, joining

customers and orders tables on customerID with the customers table on the left side of the expression returns a result set that contains all customer records, including customers who have no orders. A full outer join is the union of the result sets of both left and right outer joins.

Related terms
join, result set, table, value

Contrast with
inner join

output format

A format to which e.reports can be downloaded, including e.Analysis, Microsoft Excel, Microsoft PowerPoint, PDF, and RTF.

Related terms
Actuate e.Analysis, e.report, format

overlay chart

See dual y-axis chart.

overloaded method

In a single class, a method in which different sets of arguments can be specified.

Related terms
argument, class, method

override

To write new code that replaces the default code of an inherited method.

Related terms
inheritance, method

package

1 A set of functionally related Java classes organized in one directory.

2 A complete application, including all configuration files and programs.

Related terms
application, class, configuration file, Java

page

1 An area in a window that arranges and displays related information. A window can contain several pages, each of which is accessed by a tab. Figure G-70 shows an example of pages in the Properties window.

2 A component that specifies the design or layout of a page in the report. A page can contain controls, flows, frames, and graphic components. The relevant Actuate Foundation Class is AcPage.

Related terms
Actuate Foundation Class (AFC), component, control, design, flow, frame, layout, Properties window, report, tab

Contrast with
JavaServer Page (JSP), Page List, PageStyle, web page

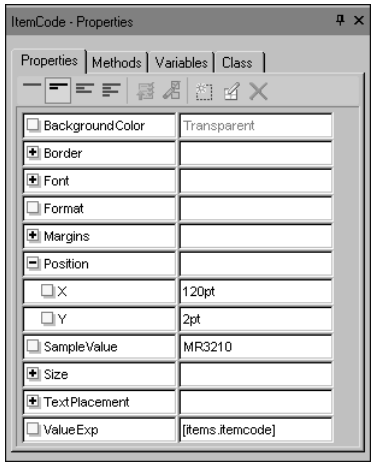


Figure G-70 Pages in the Properties window

page footer

Content that appears at the bottom of each page. For example, a page footer can display a date and a page number.

Contrast with
page header, PageFooter, PageHeader

page header

Content that appears at the top of each page. For example, a page header can display a document title and an illustration.

Contrast with
page footer, PageFooter, PageHeader

page-level security

See Actuate e.Reports Page Level Security option.

Page List

A component that maintains a list of page layouts. The relevant Actuate Foundation Classes include AcCustomPageList, AcLeftRightPageList, AcSimplePageList, and AcTitleBodyPageList.

Related terms
Actuate Foundation Class (AFC), component, layout, page

Contrast with
PageList

page number control

A component that adds page numbers to a report. A page number control must be in a flow, frame, or page. The relevant Actuate Foundation Class is

AcPageNumberControl. Figure G-71 shows a page number control placed in a page outside the flow.

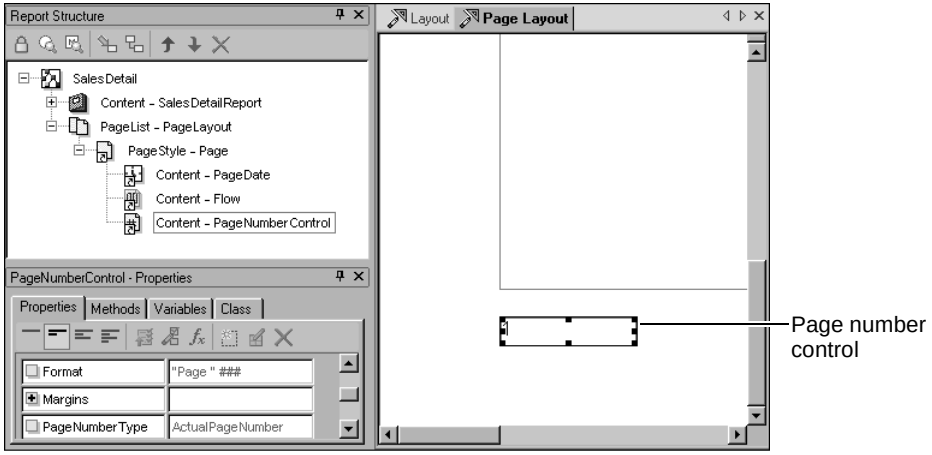


Figure G-71 Page number control placed in a page

Related terms

Actuate Foundation Class (AFC), component, control, flow, frame, page, report

page security

See Actuate e.Reports Page Level Security option.

PageFooter

A component reference property that identifies a component displayed or printed at the bottom of each page except, optionally, the last page. For example, PageFooter can contain a running total. PageFooter is a slot in the layout editor and the report structure. Figure G-72 shows an empty PageFooter slot in the Report Structure window.

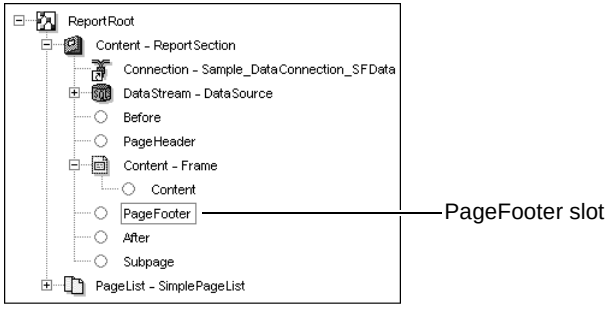


Figure G-72 PageFooter slot

Related terms
component, component reference property, layout editor, page, Report Structure window, slot

Contrast with
page footer, page header, PageHeader

PageHeader

A component reference property that identifies a component displayed or printed at the top of each page except, optionally, the first page. For example, PageHeader can display the report title in a different format from the report title format in the report's Before frame on the first page. PageHeader is a slot in the layout editor and the report structure. Figure G-73 shows a PageHeader slot in the Report Structure window.

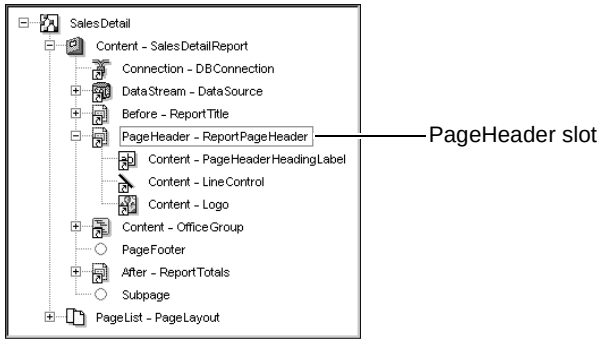


Figure G-73 PageHeader slot containing a frame

Related terms
component, component reference property, frame, layout editor, page, report, Report Structure window, slot

Contrast with
page footer, page header, PageFooter

PageList A component reference property that identifies the overall page style for the report, such as left-and-right page list, simple page list, and title-and-body page list. PageList is a slot in the Report Structure window, as shown in Figure G-74.

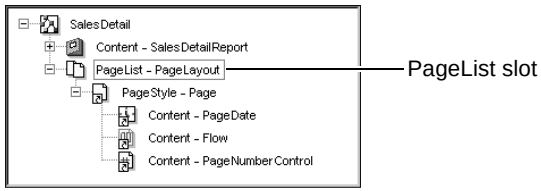


Figure G-74 PageList slot containing a page

Related terms
component reference property, page, report, Report Structure window, slot

Contrast with
Page List, PageStyle

PageStyle A component reference property that identifies a page component. A page component can contain visual elements such as page numbers. PageStyle is a slot in the Report Structure window, as shown in Figure G-75.

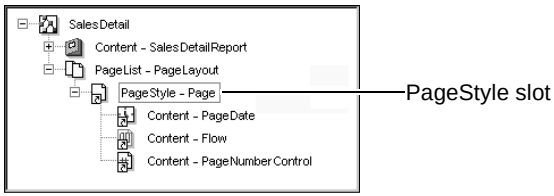


Figure G-75 PageStyle slot

Related terms
component, component reference property, page, Report Structure window, slot

Contrast with
PageList

palette See toolbox.

palette file See class representation (.apr) file.

palette representation (.apr) file

See class representation (.apr) file.

pane An area in a user interface. For example, Figure G-76 shows the navigation pane in Information Console.

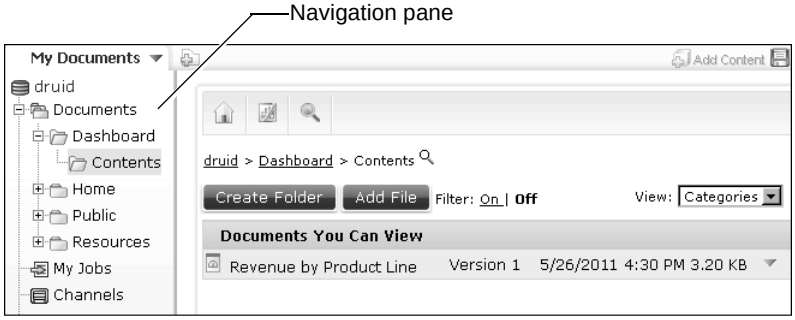


Figure G-76 Pane

Related terms
Information Console, interface

parallel section

A component that contains two or more subreports displayed or printed on the same page. For example, the left side of the page contains a report that displays or

prints employee addresses, and the right side of the page contains a report that displays or prints employee salary histories. The relevant Actuate Foundation Class is AcParallelSection. Figure G-77 shows a parallel section that contains two subreports in the Report Structure window.

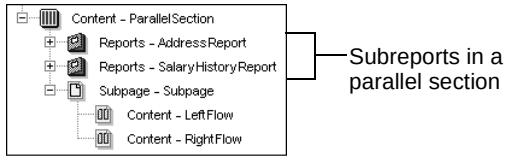


Figure G-77 Parallel section

Related terms
Actuate Foundation Class (AFC), component, page, report, Report Structure window, section, subreport

parameter **1** A report element or variable that provides input to the execution of the report. A user types or selects values for parameters on the Parameters page, as shown in Figure G-78.

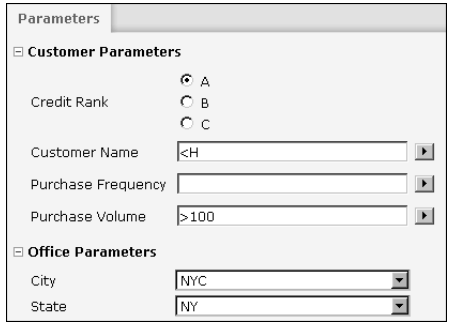


Figure G-78 Parameters page

2 In Actuate SQL, a variable that appears in a query, for example, in a WHERE clause.

3 The definition of an argument to a procedure.

Related terms
Actuate SQL, argument, Parameters page, procedure, query, report, variable

Contrast with
ad hoc parameter, function, report parameter, subroutine

parameter editor

A tool that displays the parameters that a report design contains. A parameter editor supports adding a parameter to a report design, modifying how a parameter appears to a report user, and deleting a parameter from a report design. Figure G-79 shows the parameter editor in e.Report Designer Professional.

Related terms
design, parameter, report

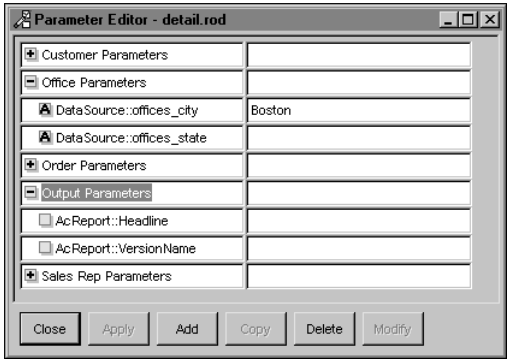


Figure G-79 Parameter editor

Contrast with
Parameters page

Parameters page

In Information Console and Management Console, a page that prompts a user to enter values for parameters when generating a report. Figure G-78 shows the Parameters page.

Related terms
Information Console, Management Console, page, parameter, report, value

parent class

See superclass.

password An optional code that restricts user name access to a resource on a computer system. For example, in an Encyclopedia volume, passwords can be up to 256 characters in length and can contain any characters except control characters or spaces. Passwords are case-sensitive. Passwords can use a mixture of case, alphabetical, and numeric characters to increase security.

Related terms
case sensitivity, Encyclopedia volume, user name

Contrast with
security role

pattern A template or model for implementing a solution to a common problem in object-oriented programming or design. For example, the singleton design pattern restricts the instantiation of a class to only one object. The use of the singleton pattern prevents the proliferation of identical objects in a run-time environment and requires a programmer to manage access to the object in a multithreaded application.

Related terms
class, design, instantiation, multithreaded application, object, object-oriented programming, run time, template

pattern matching

See search expression.

PDF converter

A tool that converts a report to a PDF file.

Related terms
converter, report

persistent object

An object that is stored in a file. For example, most objects created by running a report object executable (.rox) file, including data controls, graphical controls, pages, and sections, are persistent. A report object instance (.roi) file contains only persistent objects.

Related terms
control, data control, file types, object, page, report object executable (.rox) file, report object instance (.roi) file, run, section

Contrast with
transient object

Persistent Object Storage Manager (POSM)

A mechanism used for reading and writing a stored object that exists until the report object instance (.roi) file is deleted.

Related terms
object, report object instance (.roi) file

personal folder

See home folder.

platform The software and hardware environment in which a program runs. Linux, MacOS, Microsoft Windows, Solaris OS, and UNIX are examples of software systems that run on hardware processors made by vendors such as AMD, Apple, Hewlett-Packard, IBM, Intel, Motorola, and Oracle.

PowerPoint converter

A tool that converts a report to a Microsoft PowerPoint (.ppt) file.

Related terms
converter, report

previewer A tool that supports displaying data or a report. A data previewer enables the report developer to review the data rows returned by a query before designing

the report layout. A report previewer enables the report developer to review and improve the report layout before delivery to the user.

Related terms
data, data row, design, layout, query, report

privilege A level of access to an item in an Encyclopedia volume. Users have privileges either directly or through security roles. The privileges include, for example, the ability to delete, execute, read, and write to an object. The user who develops an item and places it in the Encyclopedia volume and the administrator both have all privileges for that item. Table G-2 lists the privileges a user can be granted for each item in an Encyclopedia volume.

Table G-2 Privileges

Privilege	Enabled functionality in an Encyclopedia volume
Delete	Remove items from an Encyclopedia volume.
Execute	Run items from the Encyclopedia volume.
Grant	Extend privileges for a specific item in the volume to other users.
Read	Open, work with, print, and download an item in the volume.
Secure read	Read secure parts of a report in the Encyclopedia volume. To use secure read, the user must have e.Reports Page Level Security option available and assigned.
Trusted execute	Execute an information object without having execute privilege for the information object's underlying data sources. This privilege applies only to information object (.iob) files and map (.sma) files.
Visible	View items in the Encyclopedia volume.
Write	Place and modify an item in an Encyclopedia volume.

Related terms
Actuate e.Reports Page Level Security option, administrator, data source map (.sma) file, Encyclopedia volume, information object, information object (.iob) file, object, report, run, security role

procedure A set of commands, input data, and statements that perform a specific set of operations. For example, functions, methods, and subroutines are all procedures.

Related terms
data, function, method, statement, subroutine

progressive viewing

A mechanism that displays the first page of a report in a web browser as soon as it generates.

Related terms
report

Project browser

A tool that lists the components included in a report design, as shown in Figure G-80. The list is organized alphabetically by file name and by symbol name within each file.

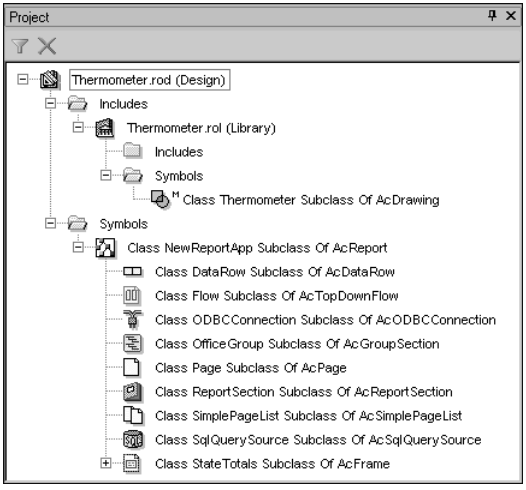


Figure G-80 Project browser

Related terms
component, design, report

Properties page

A page in the Properties window that lists the properties of a component, such as ValueExp. Use the Properties page to specify values for properties. Figure G-81 shows a Properties page.

Related terms
component, page, Properties window, property, value

Contrast with
Class page, Methods page, Variables page

Properties window

A tool used to define, manipulate, and view methods, properties, and variables of a component. The Properties window also identifies the class from which the component derives. The Properties window consists of the Properties page, Methods page, Variables page, and Class page, as shown in Figure G-82.

Related terms
class, Class page, component, method, Methods page, Properties page, property, variable, Variables page

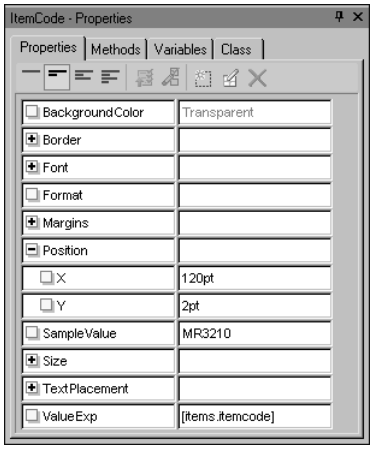


Figure G-81 Properties page

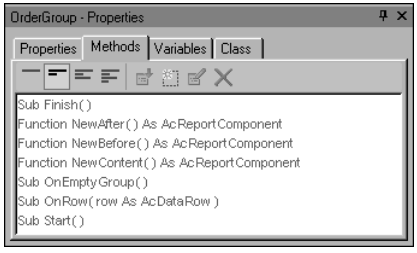


Figure G-82 Properties window

property A characteristic of an item that controls its appearance and behavior. For example, use the Properties page in the Properties window or use code to specify values for properties. For example, a report developer can specify a font size for a label control or the user name and password for a database connection.

Related terms
database connection, font, label control, Properties page, Properties window, user name, value

Contrast with
method

property list

See Properties page.

protocol A communication standard for the exchange of information. For example, in TCP/IP, the internet protocol (IP) is the syntax and order through which messages are received and sent.

Related term
syntax

publish

1 To upload files to an Encyclopedia volume to make them available to users. Users can run published report executable files.

2 A command to place a component in a library for reuse. When the component is reused, the component in the design is a reference to the component in the library.

Related terms
component, design, Encyclopedia volume, library, reference, report, report executable file

query

A statement specifying the data rows to retrieve from a data source. For example, a query that retrieves data from a database typically is a SQL SELECT statement.

Related terms
data, data row, data source, database, SQL SELECT statement

Query by Example (QBE)

A syntax used to write expressions that specify data to retrieve from a data source. For example, you can use a QBE expression to specify an ad hoc parameter value. The query is modified based on the QBE expression. Figure G-83 shows QBE expressions.

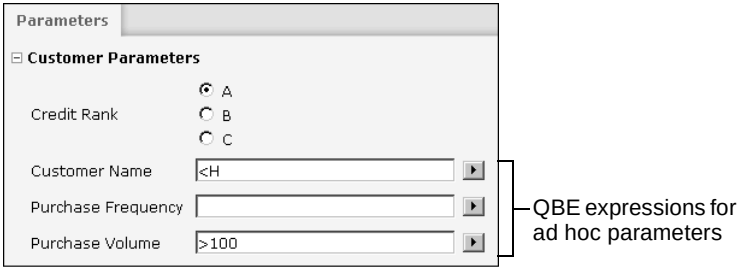


Figure G-83 Query by Example expressions

Related terms
ad hoc parameter, data, data source, expression, query, syntax, value

query data stream

A data stream that obtains data from a relational database using a SQL (Structured Query Language) query.

Related terms
data, data stream, database, query, Structured Query Language (SQL)

query editor

A tool used to write a statement that retrieves data from a data source. Figure G-84 shows the graphical query editor in e.Report Designer Professional. The upper pane supports selecting tables and specifying the joins between those tables. The lower pane displays the query text.

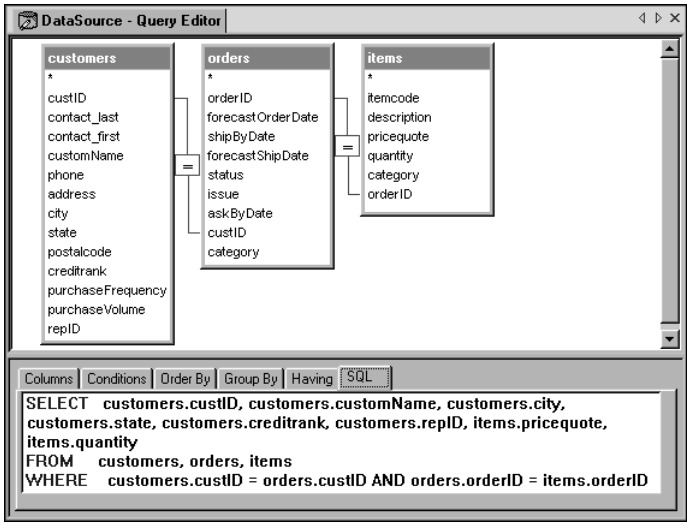


Figure G-84 Query editor

Related terms
Actuate e.Report Designer Professional, data, data source, join, query, statement, table

Contrast with
SQL SELECT statement, textual query editor (TQE)

query parameter

See parameter.

query synchronization

The process of updating the query so that it is consistent with a modified database schema.

Related terms
database, query, schema

range

1 A continuous set of values of any data type. For example, 1–31 is a numeric range.

2 The distance between the start and end values of the *x*-axis in a chart.

Related terms
chart, data type, value

Contrast with
tick interval

read privilege

See privilege.

rectangle control

A control used to add box shapes to a report design. The relevant Actuate Foundation Class is AcRectangleControl. Figure G-85 shows two rectangle controls in the layout editor.

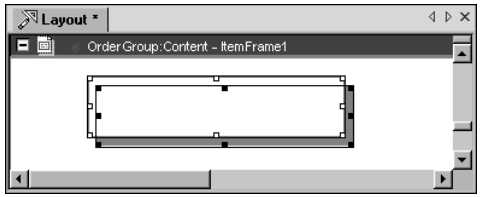


Figure G-85 Rectangle controls in the layout editor

Related terms
Actuate Foundation Class (AFC), control, design, layout editor, report

Contrast with
drawing control, line control

recursion A process in which a function or subroutine directly or indirectly calls itself to accomplish a task. The recursive function or subroutine continues until an external event interrupts it or until a specified condition occurs.

Related terms
event, function, subroutine

reference The pointer to an existing class in a report object design or a component library. To modify the properties of a class that is a reference, modify the existing class to which it is a reference or create a subclass of the class to which it is a reference.

Related terms
class, component, library, property, report object design, subclass

report A category of documents that presents formatted and structured content from one or more data sources, such as a database or text file. A sample report appears in Figure G-86.

Related terms
data source, database, format, image, structured content

report bursting

A report design technique used to break a report into several smaller reports. Using this technique, the single report object executable (.rox) file generates multiple report object instance (.roi) files. Only scheduled reports support bursting.

Related terms
design, report, report object executable (.rox) file, report object instance (.roi) file

Contrast with
report object, Reportlet

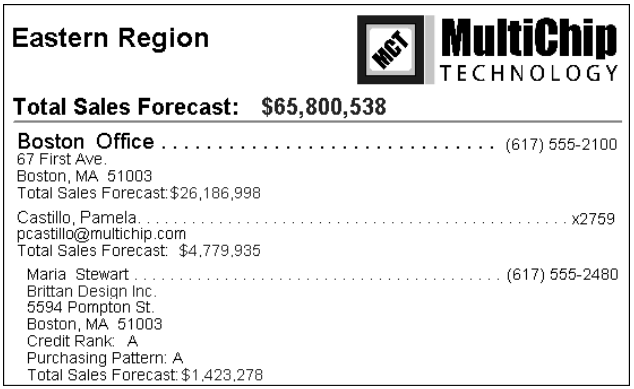


Figure G-86 Report showing formatted text and an image

report chart

See chart.

report document

A file in an Encyclopedia volume or file system that contains a report.

Related terms
Encyclopedia volume, report

Contrast with
file types

Report Encyclopedia volume

See Encyclopedia volume.

report executable file

A file that contains instructions for generating a report document.

Related term
report document

Contrast with
file types, report object executable (.rox) file

report, nested

See subreport.

report object

An object that contains all other objects in a report. The report object is the root of the report structure. The relevant Actuate Foundation Class is AcReport.

Related terms
Actuate Foundation Class (AFC), object, report

report object design

A specification that describes how to produce a report object instance.

Related terms
report object instance

Contrast with
report object design (.rod) file, report object instance (.roi) file

report object design (.rod) file

A binary file type that contains a report object design. Saving a report object design creates a design file.

Related terms
design, file types, report object design

report object executable (.rox) file

A binary file type that contains the instructions for generating a report document. Building a report creates a report object executable file from a report object design and associated Basic source files and component libraries.

Related terms
Basic source (.bas) file, component library, design, file types, report, report document, report object design

Contrast with
executable file

report object instance

An Actuate Basic report document.

Related terms
Actuate Basic technology, report document

Contrast with
report object instance (.roi) file

report object instance (.roi) file

A binary file type that contains the viewable report.

Related terms
file types, report

report object library

A collection of reusable and shareable, published components that provide consistent behavior and appearance across a suite of reports. Modifying a component in a library updates the component in all report designs that use that library.

Related terms
component, design, publish, report

Contrast with
report object library (.rol) file

report object library (.rol) file

A binary file containing reusable and shareable, published components that a report developer uses in a report design.

Related terms
component, design, file types, publish, report

Contrast with
report object library

report object parameter (.rop) file

A transient file containing a list of parameters that appear in Requester.

Related terms
file types, parameter, requester

Contrast with
report

report object value (.rov) file

A file that contains parameter values that the Factory uses to generate a report. Running a report creates an ROV file.

Related terms
factory, file types, parameter, report, run, value

report output

See output format.

report parameter

A report element that enables a user to provide a value as input to the execution of the report. Using a parameter to customize a report provides more focused information to meet specific needs. For example, parameters support selecting sales information by country and city.

Related terms
parameter, report, value

report section

A section that contains components that retrieve, organize, and display information. The relevant Actuate Foundation Class is AcReportSection. Figure G-87 shows a report section in the Report Structure window.

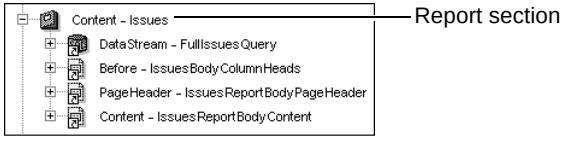


Figure G-87 Report section containing references to a data source and frames

Related terms
Actuate Foundation Class (AFC), component, data source component, frame, reference, report, Report Structure window, section

report server

See Actuate BIRT iServer.

Report Structure window

A tool that displays an outline of report components showing their functional relationships to one another, as shown in Figure G-88.

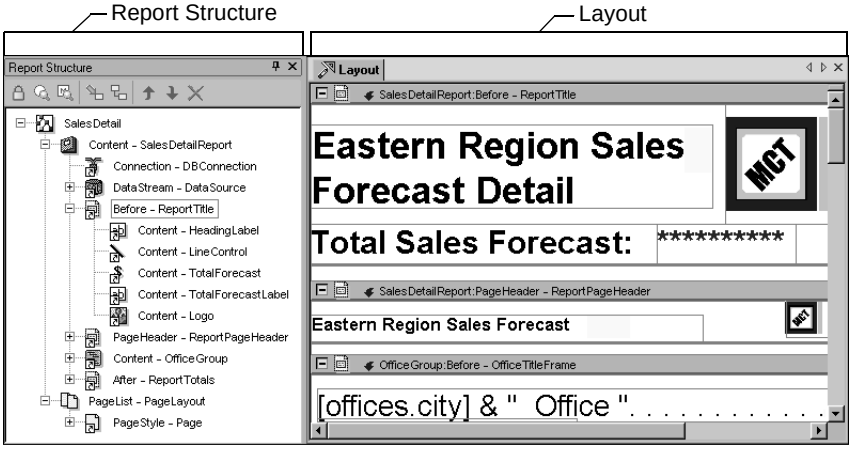


Figure G-88 Report Structure window

Related terms
component, report

Contrast with
layout editor

report template

See template.

reporting system

See Actuate BIRT iServer System.

Reportlet A portion of a report that can be embedded in a web page.

Related terms
report, web page

repository A location for rich information storage. In Actuate BIRT iServer System, the repository is an Encyclopedia volume.

Related terms
Actuate BIRT iServer System, Encyclopedia volume

request A message that an application sends to perform an action. For example, Information Console sends a request to Actuate BIRT iServer to provide the list of items in a folder.

Related terms
Actuate BIRT iServer, application, Information Console

requester A tool that provides input or modifies parameters. Generating a report from a report executable file uses the values of these parameters.

Related terms
parameter, report, report executable file

Contrast with
Parameters page

reserved word

See keyword.

resource An application file, such as a configuration file, image, library, or template.

Related terms
application, configuration file, image, library, template

result set Data rows from an external data source. For example, the data rows that are returned by a SQL SELECT statement issued to a relational database are a result set. A stored procedure can return one or more result sets.

Related terms
data row, data source, database, SQL SELECT statement, stored procedure

rich information application (RIA)

A web application that has many of the characteristics of desktop application software, typically delivered by way of a browser, a browser plug-in, extensive use of JavaScript, or a virtual machine.

Related term
application

right outer join

See outer join.

ROD See report object design (.rod) file.

ROI See report object instance (.roi) file.

ROL See report object library (.rol) file.

role See security role.

ROP See report object parameter (.rop) file.

ROS See search definition (.ros) file.

ROV See report object value (.rov) file.

row **1** A record in a table.

2 A horizontal sequence of cells in a cross tab.

Related terms
cell, cross tab, table

Contrast with
data row

row key An expression used to collect data rows into row groups and subgroups in a cross-tab control.

Related terms
cross tab, data row, group, expression

Contrast with
column key

ROX See report object executable (.rox) file.

run **1** To execute a program, utility, or other machine function.

2 To request current data in a new instance of a report. For example, run a report object executable (.rox) file to generate a new report.

Related terms
data, report, report object executable (.rox) file

Contrast with
report object instance (.roi) file

run time The period of time in which a computer program executes. For example, a report executable generates a report during run time.

Related terms
report, report executable file

Contrast with
design time, view time

running aggregate

An expression that calculates a value in one pass through the data rows in a data source. For example, an expression that calculates the sum of all orders in a group of data rows is a running aggregate.

Related terms
data row, data source, expression, value

Contrast with
lookahead aggregate

schema **1** A database schema specifies the structure of database entities such as tables and the relationships between these entities.

2 An Extensible Markup Language (XML) schema defines the structure of an XML document. An XML schema consists of element declarations and type definitions that describe a model for the information that a well-formed XML document must contain. The XML schema provides a common vocabulary and grammar for XML documents that support exchanging data among applications.

Related terms
application, data, database, declaration, element, Extensible Markup Language (XML), table, well-formed XML

scope The parts of a program in which a symbol or object exists or is visible. The location of an item's declaration determines its scope. For example, the global scope in an Actuate Basic program holds all global variable, function, and class names. Scopes can be nested. For example, a method introduces a new scope for its parameters and local variables. A class introduces a scope for its member variables, member functions, and nested classes. Code in a method in one scope has visibility to other symbols in that same scope and, with certain exceptions, to symbols in outer scopes.

Related terms
Actuate Basic, class, declaration, function, member, method, object, parameter, variable

Contrast with
scope resolution operator (::)

scope resolution operator (::)

In Actuate Basic programming language, an operator that refers to symbols in a specific scope. A leading :: refers to the global scope. For example, CustomerFrame::CustId refers to the control CustId within the scope of the frame, CustomerFrame. This control is different from TotalsFrame::CustId, which is scoped to TotalsFrame. Actuate Basic code accesses static variables in a class by using this operator, not dot notation.

Related terms
Actuate Basic, class, control, dot notation, frame, operator, scope, static variable

Scratch Pad

A tool that stores components outside the report design structure. The Scratch Pad is useful for restructuring a report design. Figure G-89 shows the Scratch Pad.

Related terms
component, design, report

search **1** To find a string in a document.

2 A mechanism that uses a set of search conditions to find data in a report.

Related terms
data, report, search conditions, string

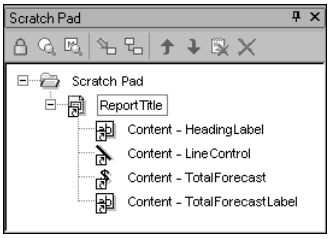


Figure G-89 Scratch Pad

search conditions

A set of search expressions to apply to a search. For example:

```
Order Total > 1000000
```

Related terms
search, search expression

search definition (.ros) file

A file that contains search conditions, data fields to display, and report document structure information.

Related terms
data, field, file types, report document, search conditions

search expression

A condition used in a search, for example:

```
Order Total > 1000000
```

Related term
search

Contrast with
wildcard

search indexing

A technique used to identify frequently searched controls to support faster searching.

Related terms
control, search

section

A component that determines the logical structure of information in a report design. Sections process data rows to generate output. There are five types of sections: conditional, group, parallel, report, and sequential sections.

Related terms
component, conditional section, data, data row, design, group section, parallel section, report, report section, sequential section

secure read privilege

See privilege.

security ID

An identifier such as an assigned name, a security role, or a user name specified in an Actuate Basic report by the report designer to restrict or support access to report components using an access control list (ACL).

Related terms
access control list (ACL), Actuate Basic report, component, report, security role, user name

security role

A name for a set of privilege levels. Assigning security roles to a user defines the user's privileges.

Related term
privilege

Contrast with
security ID, user name

SELECT See SQL SELECT statement.

select To highlight one or more items in a user interface component, such as a dialog box or a layout editor. Figure G-90 shows selected items in a report design in the layout editor.

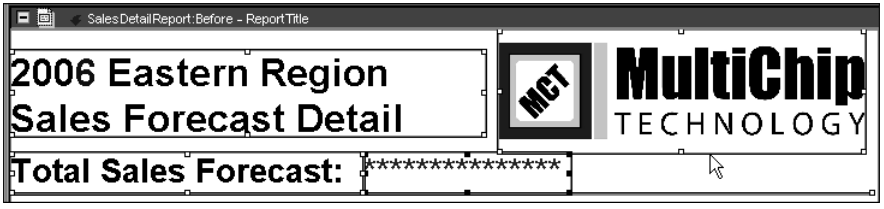


Figure G-90 Selected items in layout editor

Related terms
design, interface, layout editor

semantic type

An attribute that describes a data type. An application uses the semantic type to determine the use and purpose of a data type. For example, Actuate Basic code can use the semantic type, AC_SEMANTIC_TYPE_OBJECT_ID, to identify a data type, such as AcProperty::STRING.

Related terms
Actuate Basic, application, data type

sequential section

In an Actuate Basic report design, a section that contains several frames, subreports, or other sections that display or print in a specified order. The relevant Actuate Foundation Class is AcSequentialSection.

Related terms
Actuate Basic report, Actuate Foundation Class (AFC), design, frame, section, subreport

series

A sequence of related values. In a chart, for example, a series is a set of related points. Figure G-91 shows a bar chart that displays a series of quarterly sales revenue figures over four years.

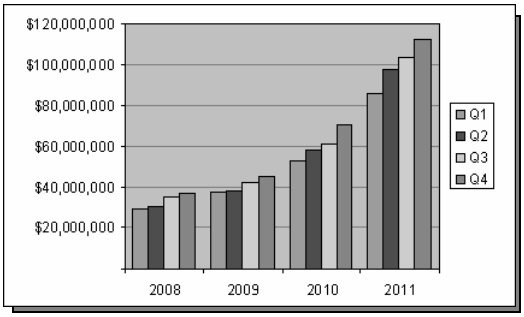


Figure G-91 Series in a chart

Related terms
chart, value

Contrast with
category

server

See Actuate BIRT iServer and web server.

shared library

A library of routines loaded and linked into an application at run time in the UNIX and Linux environments. These routines can be called from an Actuate Basic report.

Related terms
Actuate Basic report, Actuate BIRT iServer System, application, driver, library, run time

Contrast with
dynamic link library (DLL)

Single data type

An Actuate Basic data type that stores single-precision floating point numbers, ranging in value from -1.797693134862315E308 to -2.23E-308 for negative values and from 2.23E-308 to 1.797693134862315E308 for positive values and zero.

Related terms
Actuate Basic, data type, value

Contrast with
Double data type

slot A visual target in Report Structure that represents a component reference property. A slot indicates where a report designer can place a component. Figure G-92 shows empty and filled slots.

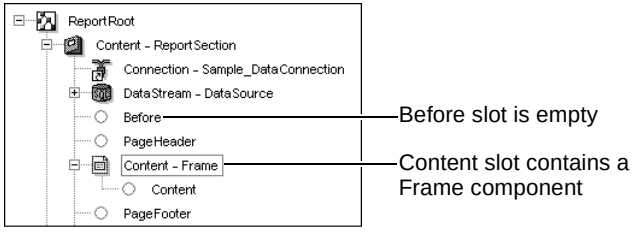


Figure G-92 Empty and filled slots in a report structure

Related terms
component, component reference property, design, frame, property, report, Report Structure window

SMA See data source map (.sma) file.

SmartSearch

See search.

sort To specify the order in which data is processed or displayed. For example, customer names can be sorted in alphabetical order.

Related term
data

Contrast with
sort key

sort filter A type of data filter that sorts data rows for another data adapter.

Related terms
data adapter, data filter, data row

sort key An expression used to sort data. For example, if you sort data by country, the country field is a sort key. You can sort data using one or more sort keys.

Related terms
data, expression, field, sort

SQL (Structured Query Language)

A language used to access and process data in a relational database.

Related terms
data, database

Contrast with
query, SQL page, SQL SELECT statement

SQL page A page in the graphical query editor that displays the SQL SELECT statement, as shown in Figure G-93.

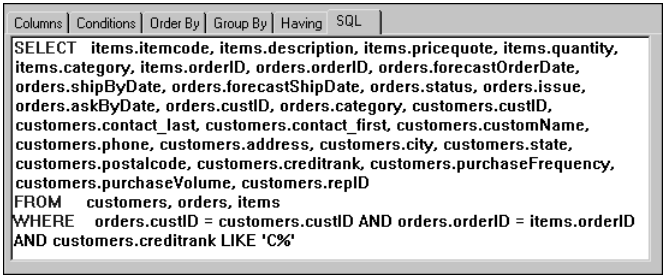


Figure G-93 SQL page in the graphical query editor

Related terms
page, query editor, SQL SELECT statement

Contrast with
Structured Query Language (SQL)

SQL SELECT statement

A query statement in SQL (Structured Query Language) that provides instructions about the data to retrieve from a database. For example, the following SQL query accesses a database's customers table and retrieves the customer name and credit rank values where the credit rank is less than or equal to B. The SQL query then sorts the values by credit rank and customer name.

```
SELECT   customers.creditrank, customers.customName
FROM     customers
WHERE    customers.creditrank <= 'B'
ORDER BY customers.creditrank, customers.customName
```

Related terms
data, database, query, sort, SQL (Structured Query Language), statement, table

Contrast with
query editor, SQL page

state The set of information about a client, database, or object that is relevant to an application. For example, in web applications, the client state can include information such as the pages the client has visited, any identification that the user has provided, the items the user has selected on earlier forms, and various preferences that the user expressed. Storing and accessing the client state supports continuity in web applications when users move between pages.

Related terms
application, database, object, page

Contrast with
instance variable

statement **1** A syntactically complete unit in a programming language that expresses one action, declaration, or definition. To continue a statement on another line, Actuate Basic uses a plus sign (+) as the first character of the next line.

2 An object that provides a way to execute SQL (Structured Query Language) statements. The relevant Actuate Foundation Class is AcDBStatement.

Related terms
Actuate Basic, Actuate Foundation Class (AFC), character, declaration, object, SQL (Structured Query Language)

Contrast with
SQL SELECT statement

static control

A component that does not display data from a data source. For example, images, labels, and lines are static controls. For these controls, the relevant Actuate Foundation Classes are AcImageControl, AcLabelControl, and AcLineControl. Figure G-94 shows the icons for the image, label, and line static controls.

Figure G-94 Static controls

Related terms
Actuate Foundation Class (AFC), component, control, data, data source, image, image control, label control, line control

Contrast with
data control

static hyperlink

A link to a location on a web page or in a file.

Contrast with
dynamic hyperlink, hyperlink

static parameter

A variable for which a report user can set a value when generating a report. The parameter value affects the report output. For example, a report design can use a static parameter in a query to filter the data rows returned by a data source.

Related terms
data row, data source, design, filter, parameter, query, report, value, variable

static variable

In Actuate Basic, a variable shared by all instances of a class and its descendant classes. In Java, a static variable is known as a class variable. The compiler specifies the memory allocation for a static variable. The program receives the memory allocation for a static variable as the program loads.

Related terms
Actuate Basic, class, class variable, descendant class, Java, variable

stored procedure

A named set of one or more SQL (Structured Query Language) queries that is stored in a database and can be called from an application.

Related terms
application, database, query, SQL (Structured Query Language)

stored procedure browser

A window that lists the names of available stored procedures. You use the Stored Procedure Data Source Builder to access the stored procedure browser.

Related terms
stored procedure, Stored Procedure Data Source Builder

Stored Procedure Data Source Builder

A tool that builds a data source using a stored procedure from a database.

Related terms
data source, database, stored procedure

Contrast with
stored procedure browser

string

A sequence of characters, for example 'Hello world'.

Related term
character

String data type

A data type that consists of a sequence of contiguous characters including letters, numerals, punctuation marks, and spaces. In Actuate Basic, variable-length strings can contain up to approximately 2,147,283,647 characters or up to your computer's memory limit.

Related terms
Actuate Basic, character, data type, string

Contrast with
string expression

string expression

An expression that evaluates to a series of contiguous characters. Parts of the expression can include a function that returns a string, a string constant, a string literal, a string operator, or a string variable. For example, in 'abc'+'def' is a string expression that evaluates to 'abcdef'.

Related terms
character, constant, expression, function, operator, string, variable

Contrast with
String data type

structural component

A component that controls how the report is organized. A structural component is visible in Report Structure but not in the layout editor or in the final report. An example of a structural component is a sequential section.

Related terms
component, layout editor, report, Report Structure window, sequential section

Contrast with
visual component

structural relationship

A relationship in which one component references another component. Report Structure displays structural relationships. For example, a frame references the controls contained within it.

Related terms
component, control, frame, Report Structure window

structured content

A formatted document that displays information from one or more data sources.

Related terms
data source, format

Contrast with
report

Structured Query Language (SQL)

See SQL (Structured Query Language).

study chart

The part of a chart that appears below the base chart. A study chart uses a different set of axes from the base chart and typically displays different types of data. In Figure G-95, a study chart shows the trading volume for each date and a base chart shows the stock price trading range.

Related terms
base chart, chart, data

Contrast with
dual y-axis chart

style

A named set of formatting characteristics, such as alignment, borders, color, and font that report developers apply to a component or report item to control its appearance.

Related terms
component, font, format, report

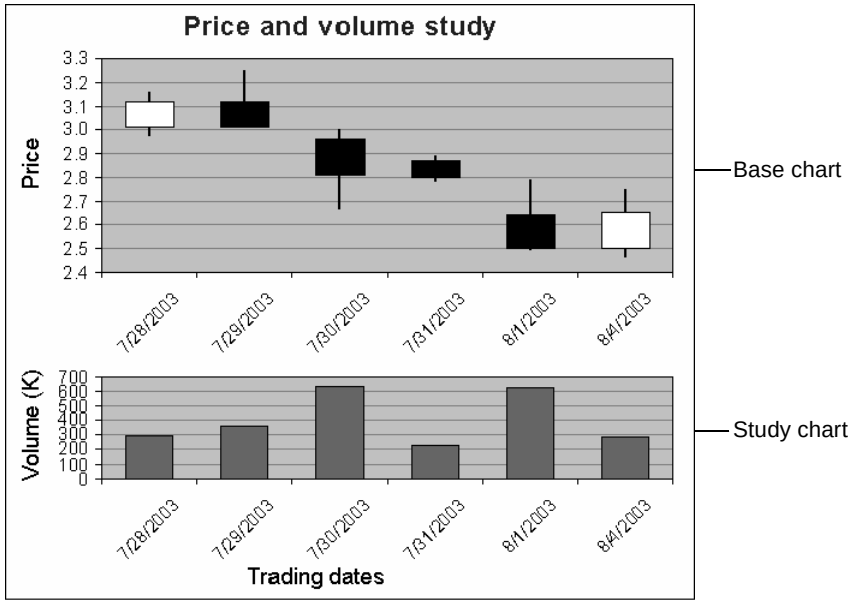


Figure G-95 Study chart and a base chart

subclass A class that is the immediate descendant class of another class. When you change the original class, the unmodified properties of the subclass also change. A change to the subclass does not affect the original class. The Project browser window displays the class relationships in a report design, as shown in Figure G-96.

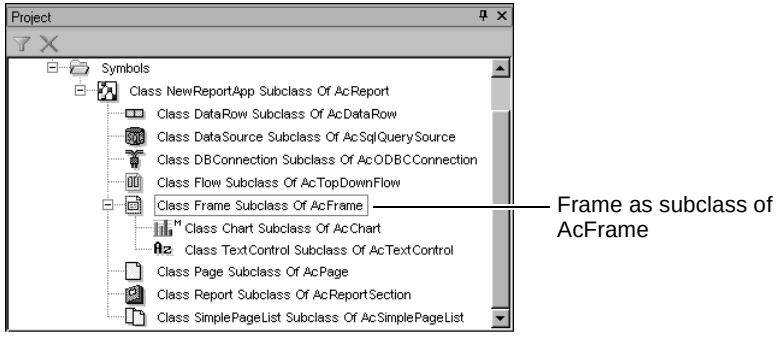


Figure G-96 Subclass

Related terms
class, descendant class, Project browser, property, subclass

Contrast with
ancestor class, base class, reference, superclass

Subpage A component reference property that identifies the components of a page associated with a section. Subpage is a slot in Report Structure and the layout editor. The relevant Actuate Foundation Class is AcSubPage. Figure G-97 shows a Subpage slot in Report Structure.

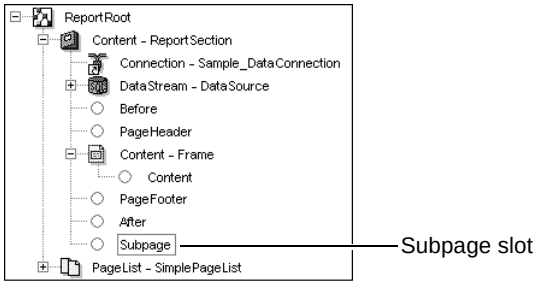


Figure G-97 Subpage slot

Related terms
Actuate Foundation Class (AFC), component, component reference property, layout editor, page, Report Structure window, section, slot

subpage A component specifying the layout of a page associated with a section. The relevant Actuate Foundation Class is AcSubPage. Figure G-98 shows a subpage component in Report Structure.

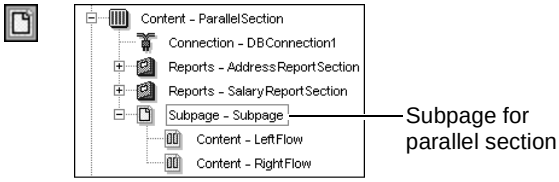


Figure G-98 Subpage component

Related terms
Actuate Foundation Class (AFC), component, page, Report Structure window, section

subreport An item using data from a different data set or data source from other items in a report design. An outer report can contain multiple subreports, also called nested reports. In this case, the subreports typically use data values from the outer report to filter data rows for display. Alternatively, multiple independent subreports exist at the same level in the report design. A subreport is a report section placed inside another section. The relevant Actuate Foundation Class is AcReportSection. For example, Figure G-99 shows subreports in a parallel section.

Figure G-99 Subreports in a parallel section

Related terms
Actuate Foundation Class (AFC), data, data source, design, filter, parallel section, report, report section, section, value

subroutine

A sequence of instructions defined as a separate unit within a program, allowing the unit to be invoked anywhere in the program simply by including its name as one of the instructions. Subroutines are similar to functions, except they have no return value.

Related terms
function, value

Super A keyword that accesses a method in a superclass. Super searches progressively until it finds the method.

Related terms
keyword, method, superclass

Contrast with
class

superclass

The immediate ancestor class. The Project browser, shown in Figure G-100, shows the superclass of each class in a report design.

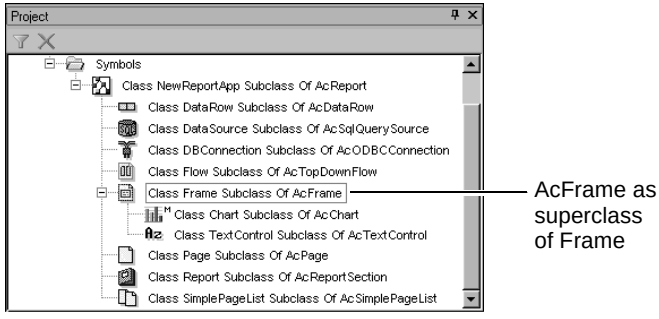


Figure G-100 Superclass

Related terms
ancestor class, class, design, Project browser, report

Contrast with
descendant class, subclass

symbol A name that represents an entity. A symbol is unique in a scope. The Project browser shows the symbols used in a report design. For example, a symbol can be the name of a class or variable.

Related terms
class, design, Project browser, report, scope, variable

syntax The rules that govern the structure of a programming language.

tab The label above or below a page in a window that contains multiple pages. Figure G-101 shows tabs that access different pages in the Properties window.

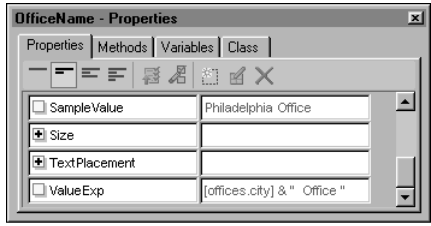


Figure G-101 Tabs in the Properties window

Related terms
page, Properties window

Contrast with
label control

table A named set of columns in a relational database. Figure G-102 shows tables in the graphical query editor.

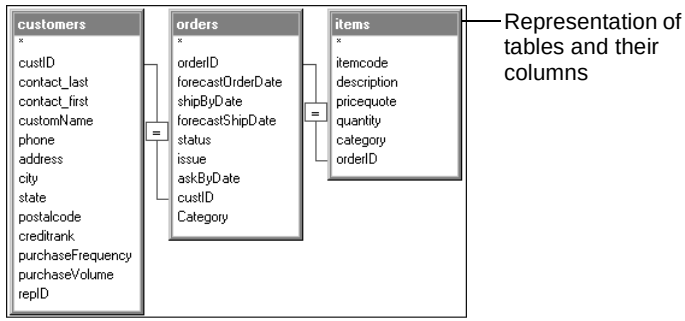


Figure G-102 Tables in the graphical query editor

Related terms
column, database, query editor

table of contents

A hyperlinked outline of report contents.

Related terms
hyperlink, report

tag An element in a markup language that identifies how to process a part of a document.

Related term
element

Contrast with
Extensible Markup Language (XML)

template A predefined structure for a report design. A template must be a report object design (.rod) file.

Related terms
design, report, report object design (.rod) file

text control

A component that displays textual data in a report. For example, a text control can display customer names. A text control must be in a flow, frame, or page. The relevant Actuate Foundation Class is AcTextControl. Figure G-103 shows text controls in the layout editor and Report Structure.

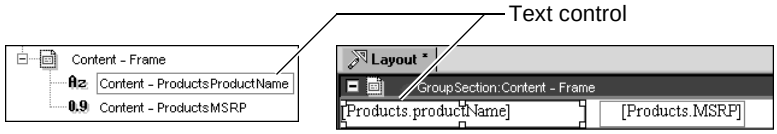


Figure G-103 Text control

Related terms
Actuate Foundation Class (AFC), component, control, data, flow, frame, layout editor, page, report, Report Structure window

Contrast with
data control, label control

textual query editor (TQE)

A textual tool used to write a SQL SELECT statement.

Related term
SQL SELECT statement

Contrast with
query editor

tick A marker that occurs at regular intervals along the *x*- or *y*-axis of a chart. Typically, the value of each tick appears on the axis.

Related terms
chart, value

Contrast with
grid, tick interval

tick interval

The distance between ticks on an axis. Figure G-104 shows a chart with a one-hour tick interval on the x-axis.

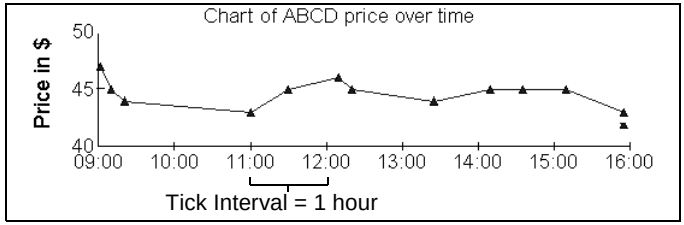


Figure G-104 Chart with one-hour tick interval on the x-axis

Related terms
chart, tick

toolbar A user interface component that provides access to common tasks. Different toolbars are available for different kinds of tasks. Figure G-105 shows the Viewer toolbar.

Figure G-105 Viewer toolbar

Related term
interface

toolbox A graphical representation of a set of components that represent specific classes available to create a report object design. The toolbox groups classes by functionality: Controls, Data, Drawing and Graphics, Pages, and Structure. Users can add tools and modify the toolbox. Figure G-106 shows the toolbox and the tools that it contains.

Related terms
class, component, report object design

transient object

An object that performs specialized tasks during report generation. Transient objects exist while an Actuate Basic report generates, but are not saved to the report file and do not appear when the report is viewed or printed. For example, connections and data rows are transient objects.

Related terms
Actuate Basic report, connection, data row, object, report

Contrast with
persistent object, report object instance (.roi) file

trusted execute privilege

See privilege.

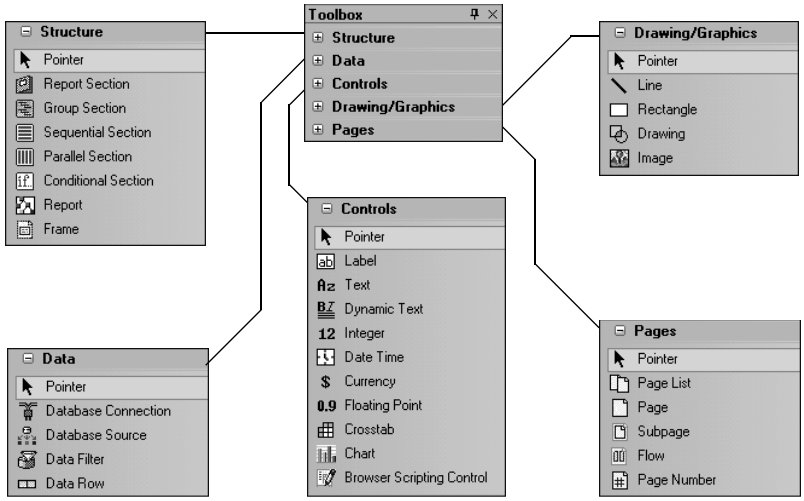


Figure G-106 Toolbox

twip A unit of screen measurement equal to 1/20 of a printer's point. There are approximately 1,440 twips in an inch or 567 twips in a centimeter.

type See data type and semantic type.

type declaration character

A character appended to a variable name to specify the variable's data type. Table G-3 lists the type declaration characters in Actuate Basic.

Table G-3 Type declaration characters

Character	Data type
%	Integer
&	Long
!	Single
#	Double
@	Currency
$	String

Related terms
Actuate Basic, character, Currency data type, data type, Double data type, Integer data type, Long data type, Single data type, string, String data type, variable

Contrast with
semantic type

Unicode A living language standard managed by the Technical Committee of the Unicode Consortium. The Unicode standard provides code points for more than 65,000 characters. Unicode encoding has no dependency on a platform or software program and thus provides a basis for software internationalization.

Related terms
character, code point, internationalization

uniform resource identifier (URI)

A set of names and addresses in the form of short strings that identify resources on the web. Resources are items such as documents, downloadable files, and images.

Related term
image

Contrast with
uniform resource locator (URL), string

uniform resource locator (URL)

A character string that identifies the location and type of a piece of information that is accessible over the web. http:// is the indicator that an item is accessible over the web. The URL typically includes the domain name, type of organization, and a precise location within the directory structure where the item is located.

Related terms
character, HyperText Transfer Protocol (HTTP), string

Contrast with
uniform resource identifier (URI)

unique ID An identifier for a persistent object that distinguishes it from all other objects.

Related terms
object, persistent object

Contrast with
transient object

universal hyperlink

See hyperlink.

URI See uniform resource identifier (URI).

URL See uniform resource locator (URL).

user name

A name that identifies a user of a resource on a computer system. For example, a user must provide a user name and password in order to log in to Information Console.

Related term
Information Console

user-defined data type

A data type defined by a programmer, not built in to the programming language. User-defined types are classes, structures, and type definitions assembled from built-in data types, such as numbers and characters.

Related terms
character, class, data type

Contrast with
semantic type

value

1. The content of a constant, parameter, symbol, or variable.
2. A specific occurrence of an attribute. For example, blue is a possible value for an attribute color.

Related terms
attribute, constant, parameter, symbol, variable

Varchar data type

An Actuate SQL data type used for string calculations. The Varchar data type stores a sequence of Unicode characters. The Varchar data type supports specifying a maximum character length for the string. For example, VARCHAR (30) represents strings with a maximum length of 30 Unicode characters.

Related terms
Actuate SQL, character, data type, string, Unicode

variable

A named storage location for data that a program can modify. Each variable has a unique name that identifies it within its scope and contains a certain type of data.

Related terms
data, data type, scope

Contrast with
class variable, Fields, global variable, instance variable, local variable, member variable, object reference variable, static variable

Variables page

A page in the Properties window that lists the variables of a component. This page supports adding new variables and creating parameters. Figure G-107 shows a Variables page.

Related terms
component, page, parameter, Properties window, property, variable

Contrast with
Class page, Methods page, Properties page

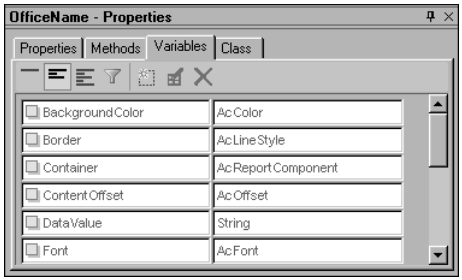


Figure G-107 Variables page

Variant data type

An Actuate Basic data type that holds data of any other type. Variants contain two pieces of information, a value and code indicating the value's type. Variant types can change dynamically when a report is running.

Related terms
Actuate Basic, data, data type, report, value

view A predefined query that retrieves data from one or more tables in a relational database. Unlike a table, a view does not store data.

Related terms
data, database, query, table

view time The period of time in which a user examines a report.

Related term
report

Contrast with
design time, run time

viewer See DHTML Viewer.

visible privilege

See privilege.

Visual Basic

A programming language developed by Microsoft. Actuate Basic is compatible with Visual Basic 3.0 and has object-oriented extensions.

Related terms
Actuate Basic, object-oriented programming

visual component

A component, such as a control or a frame, that displays data or static graphic items in a report. A visual component appears in both Report Structure and the layout editor.

Related terms
component, control, data, frame, layout editor, report, Report Structure window

Contrast with
structural component

volume See Encyclopedia volume.

Volume Administration console

See Management Console.

web directive

A command used to generate content for a web page. The web page contains the directive. In some situations, a web directive can be a parameter used by a command. The web server executes the command when a user views the web page. For example, a command embedded in a uniform resource locator (URL) can be passed by Information Console to Actuate BIRT iServer System. An example of such a command for Information Console:

```
http://localhost:8900/iportal/getjobdetails.do?jobID=1
   &serverURL=http://maximus:8000&volume=maximus
```

Related terms
Actuate BIRT iServer System, Information Console, parameter, uniform resource locator (URL), web page, web server

web page A HyperText Markup Language (HTML) page containing tags that a web browser interprets and displays.

Related terms
HyperText Markup Language (HTML), tag

web server

A computer or a program that provides web services on the internet. A web server accepts requests based on the HyperText Transfer Protocol (HTTP). A web server also executes server-side scripts, such as Active Server Pages (ASP) and JavaServer Pages (JSP).

Related terms
HyperText Transfer Protocol (HTTP), JavaServer Page (JSP), request

well-formed XML

An Extensible Markup Language (XML) document that follows syntax rules established in the XML 1.0 recommendation. Well-formed means that a document must contain one or more elements and that the root element must contain all the other elements. Each element must nest inside any enclosing elements, following the syntax rules.

Related terms
element, Extensible Markup Language (XML), syntax

wildcard A character used in a search or Boolean expression that matches one or more literal characters.

Related terms
Boolean expression, character, search expression

write privilege

See privilege.

XML (Extensible Markup Language)

See Extensible Markup Language (XML).

XML converter

A tool that converts a report to an Extensible Markup Language (XML) file.

Related terms
converter, Extensible Markup Language (XML), report

XML data format

A format that displays an entire Actuate Basic report or selected elements of the report. A report converted to XML retains its original design and graphics. XML data format ignores page components such as pages, flows, page headers, and page footers.

Related terms
Actuate Basic report, data, Extensible Markup Language (XML), report

Contrast with
XML display format

XML data stream

1. Output generated by an Extensible Markup Language (XML) converter.
2. Input from an XML data source.

Related terms
converter, data source, Extensible Markup Language (XML), XML converter

Contrast with
data stream

XML display format

An Actuate Basic report format that encapsulates Extensible Markup Language (XML) data and XML display formatting.

Related terms
Actuate Basic report, data, Extensible Markup Language (XML), format

XML element

See element.

XML report

A report in Extensible Markup Language (XML) display format generated from a report object instance (.roi) file.

Related terms
Extensible Markup Language (XML), report, report object instance (.roi) file, XML display format

Contrast with
DHTML report

XML schema

See schema.

Z-order

The priority assigned to overlapping parts of a report. Parts having a higher Z-order hide parts having a lower Z-order.

1 The order in which a combination, multiple *y*-axis, or three-dimensional chart control displays the chart series. The example in Figure G-108 shows a bar series having a Z-order of 1 and an area series having a Z-order of 0.

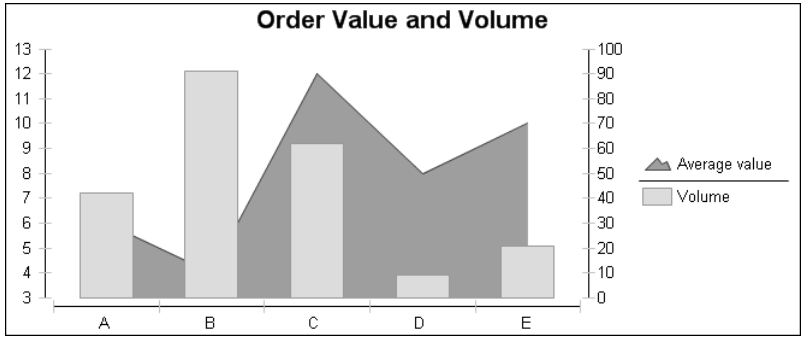


Figure G-108 Multiple *y*-axis chart having bar series in front of area series

2 The order in which an e.report stacks overlaid controls in the layout. For example, a label control having a higher Z-order than a rectangle control placed slightly below and to the right produces a shadow effect, as shown in Figure G-109.

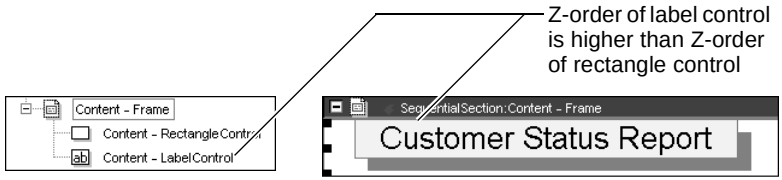


Figure G-109 Overlapping controls in a frame

Related terms
chart control, e.report, frame, label control, layout, rectangle control, series

Index

Symbols

, (comma) character
  finding 707
  search expressions and 705
, operator 705
; (semicolon) character
  parameter declarations and 460
  search expressions and 458
:: operator 838
! (exclamation point) character
  as literal value 707
  HTML tags and 742
! operator 705
? (question mark) character
  class names and 25
  parameter declarations and 460
? (question mark) character tag 743
? wildcard character 705
?? icon 129
. (dot) operator 783
. (period) character in class names 25
… (ellipsis) character 145
… button 786
' (single quotation mark) character 739
" (double quotation mark) character
  as literal value 454, 739
  link expressions and 454, 458
  search expressions and 459
" (double quotation mark) character tag 742
[ ] (brackets) characters
  search expressions and 706
{} (brace) characters 739
* (asterisk) characters in cells 326
* (asterisk) symbol in Requester 504
* operator 757
* wildcard character 705, 756
\ (backslash) character
  as escape character 708
  as literal value 739
  control names and 25
  finding 707, 708
  RTF tags and 736
  search expressions and 706, 707, 708
& (ampersand) character 454
& (ampersand) character tag 742
# (pound sign) character 459
# wildcard character 706
+ (plus sign) icon 45
< (less than) character tag 742
< operator 705
<> (angle brackets) characters
  finding 707
<= operator 705
= (equal sign) character
  searching for 707
= operator 705
> (greater than) character tag 742
> operator 705
>= operator 705
~ (tilde) character 456
- (hyphen) character
  finding 707
  non-breaking 739
  search expressions and 705
- (hyphen) character tag 742
- (minus sign) icon 45
- operator 705

A

absolute font size 742
absolute formatting 155
absolute paths 456, 590, 630
absolute URLs 640
abstract base class 749
AC_MASTER_FONT_FULLPATH_NAME
  variable 479
AC_USE_DEFAULT_FONT_FILE
  variable 479
accelerator keys. *See* keyboard shortcuts
access control lists 749
  See also security IDs
  creating security IDs for 531, 533
  customizing 543–546
  displaying for current page 535
  generating 531, 544, 545

access control lists *(continued)*
implementing page-level security and 530
nesting sections and 532, 544
access permissions. *See* privileges
access restrictions 530, 543
access types 749
accessing
blank report designs 73
color palettes 147
component libraries 572
context menus 608
data 30, 73
data perspective 21
data sources 4, 73, 410, 417
databases 4, 12, 30, 504
Encyclopedia volumes 620
external functions 572
function libraries 573
multiple view windows 693
report templates 71, 628
report wizards 10, 69
reports 594, 686
resources 588
sample configurations 5
sample reports 686
specific parts of a report 688
table of contents 689
ToolTips 36
AcChart class 196
AcChart6 class 196
accounts 594
AcCrosstab class 298
ACL variable 544, 545
ACLs. *See* access control lists
AcReport class 549, 666, 667
AcReportComponent class 667
Acrobat Reader 480
See also PDF Viewer
ActionScript 649
active perspective 619
Active Perspective Only command 44
actual page numbers 181, 538
ActualPageCount value 538
ActualPageNumber value 534
Actuate Basic 749
See also source code; source files
caution for fatal errors and 490
customizing reports with 61
displaying source code for 493
generating source code for 58, 486, 494
mapping security roles with 545
setting source encoding for 621
tracking methods calls and 489
Actuate Basic reports 750
See also e.reports; reports
Actuate Basic technology 750
Actuate Foundation Class Library 753
Actuate Foundation Classes 55, 61, 196, 495, 752
See also classes
Actuate Palette Representation files. *See* apr files
Actuate server. *See* iServer
Actuate SQL 753
See also queries; SQL statements
ad hoc parameter builder 679
ad hoc parameters 405, 430, 507, 512, 753
ad hoc reporting 504
Add Component command 621
Add Component dialog 622
Add Enum button 570
Add Group command 622
Add missing groups setting 234, 319, 323
Add overlay setting 219
Add Structure button 567
Add study setting 219
adding
aggregate data 76, 114–118
background colors 147–148, 183, 240, 241
balloon help 440
borders to chart legends 242
borders to chart titles 239
borders to components 147
borders to page layouts 183
browser scripting controls 645, 647
charts 196, 198, 206, 218–219
classes to libraries 624
column headings 96, 106–109, 306
columns to Query Editor 86
component libraries 588
components
to designs 44, 46, 584
to libraries 576
to report sections 404–405

to toolbox 621, 622, 623
computed columns 75, 100, 102, 431
controls 45, 104, 365, 376
cross tabs 299–302, 304
currency controls 129
data groups 83, 86, 93
date and time groups 236
detail reports 548, 549
directories to search paths 617
drop shadows 148
dynamic text controls 338, 341–342, 351, 366
encoding options 621
error-checking routines 526
fonts to designs 474, 478, 480
graphical elements 150–151
group headings 109–113
hyperlinks to charts 269–270
hyperlinks to reports 214, 454, 460
images 97, 151–154, 184
Information Console headlines 507
items to drop-down lists 512
labels to charts 247, 249, 252
legends to charts 237, 241, 243
libraries to designs 573, 579–584
link fields 426
missing groups to charts 232, 234
multi-line text 338
page breaks 420
page headers and footers 27, 68, 179–180
page numbers 181–182, 534
parameters to designs 507
property groups 561
row headings 306
section components 401, 402
security IDs 531, 533
server profiles 594–596, 602
static group to charts 235–236
subpages 407
summary data 16, 76, 97
table of contents 466–469
tables to Query Editor 84
titles to reports 18, 76
trendlines 260, 263
URLs 454, 455, 465
watermarks 184
address labels 338
adjusting row spacing 146
adjusting spacing around controls 348
administrators 753
Adobe Acrobat Reader. *See* Acrobat Reader
Adobe Flash Player. *See* Flash Player
Advanced Chart Options page (Chart Builder) 237, 243, 250
Advanced Grouping page (Chart Builder) 234
Advanced Grouping page (Crosstab Builder) 323
Advanced page (Document Properties) 715
Advanced setting (New Property) 561
AFC. *See* Actuate Foundation Classes
After property 754
After slot 97, 351, 754
See also group footers
aggregate data
See also aggregate values; summary data
adding 76, 114–118
building cross tabs and 299, 310
building listing reports and 297
grouping 308
aggregate expressions 754
charting data and 206, 210, 213
restrictions for 221, 223
comparing values and 118, 120
customizing properties and 564
defined 754
filtering data and 118, 119
aggregate functions 755
calculating values and 114, 118
creating cross tabs and 297
defined 755
restrictions for 120, 406
returning null values and 114
aggregate information. *See* aggregate data
aggregate rows 755
aggregate values 118, 297, 310, 755
See also aggregate data
defined 755
Alias attribute 590, 630
aliases 566, 569, 755
Align command 137
ALIGN parameter 741
aligning
browser code output 646

aligning *(continued)*
components 134, 137–138
controls 348
labels and data points 255, 256
text 117, 141, 737, 741
All Perspectives command 44
Allow user to type a value setting 512
alphanumeric searches 705, 706
alphanumeric values 90
alternate names. *See* aliases; display names
AlternateText property
building browser scripting controls and 646
displaying sample data and 645
generating PDF reports and 653
Always prompt for values setting 507
ampersand (&) character 454
ampersand (&) character tag 742
analyzing data 68, 297
ancestor classes 756
See also superclasses
And operations 701
AND operator 48, 541
angle brackets (<>) character tags 742
angle brackets (<>) characters
finding 707
animation 649
annual sales reports 210
Appearance page (Format) 132
applets 642
application programming interfaces (APIs) 756
Application server field (iServer Profile) 595
application servers 595
See also web servers
application window
See also e.Report Designer Professional
accessing elements of 36, 37
changing fonts for 613
changing toolbox in 621, 623
customizing 608
personalizing 41–44, 693
viewing status messages in 27
applications 756
See also multithreaded applications
building interactive 649
building page layouts for 167
defined 756
designing multi-platform 472–473, 479
generating browser code and 651
XML data formats and 662, 664
.apr files 624, 625, 626
.apr files. *See* class representation files
archiving rules 594, 598
area charts
See also charts
aligning components in 255, 256
as study chart layer 206
changing series labels for 250
creating 199
displaying data in 250
generating sample data for 270
labeling data points in 253
plotting empty values for 256
selecting data for 207, 220
arguments 756
See also functions; parameters
arranging items in lists 513
arrays 489, 491, 499, 756
defined 756
Asc setting 17
ascendant classes. *See* ancestor classes
ascending sort order 17, 75, 124, 125
ASCII characters 707, 708
ASCII control codes 51, 339, 341
ASCII formats 342
assignment statements 756
asterisk (*) character 756
asterisk (*) symbol in Requester 504
asterisks (*) characters in cells 326
attribute 757
attributes
See also properties
adding to XML elements 668
applying text 338, 738, 741
changing font 611, 612
mapping to controls 666
specifying page 168, 169, 170, 172
Auto value (CSS attribute) 645
autoarchive rules 594, 598
Autoinclude attribute 590
AutoScrollbar value 645
auto-split options 358
AutoSplitVertical property 358, 359, 360

Ave function 114
averages 76, 114, 310
Axes page (Chart Options) 244, 245, 246
axes values
  See also x-axis; y-axis values
  changing line styles for 246
  chart layers and 204, 205, 213
  flipping 199, 219
  formatting 205, 237, 243, 244
  labeling. *See* axis labels; axis titles
  overview 243
  plotting dual 202
  plotting empty values for 257
  positioning intersecting 245
  scaling 246–247
  setting minimum/maximum 247
  setting minor ticks for 247
axis labels
  adding 247–250
  arranging 248
  customizing 249
  defined 236
  formatting 248
  setting default font for 611
axis titles 611, 613

B

–b option (erdpro) 747
Back button 690
background colors
  browser scripting controls and 653
  chart areas and 240, 241
  chart legends and 242
  chart titles and 239
  components and 147
  cross tabs and 330
  customizing 240, 241
  dynamic text controls and 362
  page layouts and 183
background fill 139
background images 166, 184
background patterns 183
BackgroundColor property
  background images 184
  borders 148
  cross tabs 330
  page layouts 183
backgrounds, changing chart 237, 240, 241
backslash (\) character
  as escape character 708
  as literal value 739
  control names and 25
  finding 707, 708
  RTF tags and 736
  search expressions and 706, 707, 708
backslash (\) character tag 739
back-up files 746
backward compatibility 196, 746
balloon help 440, 473, 757
BalloonHelp control 473
BalloonHelp method 440
BalloonHelp property 440
bar charts
  See also charts
  adding bars to 265
  as study chart layer 206
  changing bar shapes for 266
  changing series labels for 250
  creating 199
  formatting bars in 266
  generating sample data for 270
  labeling data points in 253
  overlapping bars for 267
  plotting empty values for 256
  plotting missing values for 256
  positioning bars in 267
  selecting bar outline styles for 266
  selecting data for 207, 220
bar colors 204
bar outline styles 266
bar overlap (charts) 200
bar spacing options 265
bars in charts 198, 200, 266
.bas files. *See* source files
base charts 757
base classes 757
Base layer legend text setting 242
base layers (charts) 204, 205, 206
base unit (charts) 758
Basic. *See* Actuate Basic
Basic reports 602
  See also reports
Basic source files. *See* source files

batch files 746
Before frames 24
Before property 759
Before slot 96, 107, 351, 759
  See also group headers
bidirectional text 759
BIRT Information Object Designer. *See* IO Design perspective
BIRT iServer 750
BIRT iServer System 750
BIRT iServer System Options (licensing) 751
BIRT Studio 602
bitmapped image files 151
blank lines 347
blank report designs 72, 73, 100, 123
Blank Report option 72, 627
  See also blank report designs
Blank Report wizard 627
blank space 334, 338, 352, 354
  See also leading spaces; trailing spaces
BMP files 151
body page layouts 190–193
boilerplate text 338
bold text 738, 741
Boolean expressions 405, 759
Boolean types 518
Boolean values 141, 518
border attributes 239, 240, 241
Border dialog 239, 240, 241, 269
Border property 183
borders
  browser scripting controls and 653
  charts and 239, 240, 242, 266
  components and 147
  cross tabs and 326–329
  drop shadows and 148
  dynamic text controls and 362
  page layouts and 183
Bottom property 363
BR tag 740
brace {} character tags 739
brace {} characters 739
brackets ([ ]) characters
  search expressions and 706
Break command 487
Break line setting 256
break reports. *See* grouped reports
breakpoints 759
  clearing 487, 494
  pausing report generation and 494
  removing 487, 494
  setting 487, 494
browser button. *See* ellipsis (…) button
browser code
  See also browser scripting controls
  aligning output 646
  clipping output for 645–653
  converting to PDF formats 481, 653
  debugging 646
  displaying 647, 653
  example for Internet Explorer 654
  generating dynamically 650–653
  generating HTML tags for 644
  including in designs 642, 645, 648
  referencing 648
  setting scaling values in 652
  supported formats for 642
browser scripting controls 760
  adding to frames 645, 647
  adding to libraries 648, 654
  creating Flash objects with 649
  creating HTML forms with 648–649
  displaying values 646, 652
  getting execution context for 651
  overview 642
  replacing image controls with 480
  resizing 652
  specifying output alignment 646
browser source viewer 647
BrowserClipping property 645
BrowserCode method 650, 652
BrowserCode property 645, 646
browsers. *See* web browsers
bubble charts 200
  See also charts
Build and Run command 59, 369, 494, 526
Build command 59
Builder button. *See* ellipsis (…) button
BuildFromRow method 650, 652, 653
building interactive web pages 649
building reports 46, 58, 400, 686
  See also creating reports
built-in classes 61
bullet symbols 739, 742

bulleted lists 338, 357
Bundle Rox in Roi parameter 677
bundling reports 506
BurstGrouping sample report 549, 552
BurstGrouping.rod 549
bursting 548–549, 831
bursting examples 549–553
BurstSubreports sample report 549, 550
BurstSubreports.rod 549
buttons
  accessing help for toolbar 36
  creating radio 511, 512
  defining as hyperlinks 690
  generating values for 511, 513
  hiding toolbar 43
  setting parameter values and 680
  testing 524

C

calculated columns or fields. *See* computed columns; computed fields
calculated columns. *See* computed columns
calculations
  aggregating data and 116
  building expressions for 101, 102
  checking values for 103
  creating cross tabs and 312
  creating grouped reports and 114
  creating listing reports and 298
  debugging 485
  summarizing data and 76
call stack 487, 490
Call Stack command 487
Call Stack window 485, 489–490
candlestick charts
  See also stock charts
  adding bars 258
  creating 204
  selecting data for 212
CanIncreaseHeight property 351, 378, 391, 392
CanIncreaseWidth property 349, 350, 378, 391, 392
CanMoveLeft property 378
CanMoveUp property 348, 378
CanReduceHeight property 347, 348, 377, 378
CanReduceWidth property 378
carriage return characters 339, 708, 739
CascadeSecurity property 532, 543
cascading security 532, 543, 544
cascading style sheets 645
case sensitivity 760
case sensitivity (class names) 416
case sensitivity (declarations) 460
categories. *See* data categories
category 760
category axes. *See* x-axis values
Category gap settings 265
category keys 221
Category keyword 215
category labels 197, 220
category values 760
CategoryLabelExp property 215
cell (defined) 760
cells
  See also cross tabs; spreadsheets
  applying colors to 330
  changing data alignment in 310, 311
  defined 760
  displaying multiple aggregate values in 310
  formatting values in 329
  inserting summary data in 76
  selecting 298
Cent currency symbols 742
Center command 138
CENTER tag 740
centering
  components 138
  controls 388, 389
  text 737, 740
centimeters 614, 615
Change the report design setting 18
changes
  previewing 59
  undoing 298
changing
  charts 237–238
  component images 623
  component libraries 585
  component names 25, 26, 576

changing *(continued)*
component properties 56
component references 47
components 584, 587
cross tabs 325
custom properties 558, 563
data alignment in cells 310, 311
data types 105, 106, 499
date formats 183, 312
default formats 25, 76, 155
default layouts 123, 167
floor colors 241
font attributes 610, 611, 612
hyperlinks 237
images 152
margins 76, 168, 177, 332
padding 353
page number formats 181
page orientation 168, 171
page size 168
printing options 716
registry keys 613
report designs 18, 21, 25, 28, 69
report parameters 504, 510
search conditions 704
sort order 125
source code 621
value expressions 220
wall colors in charts 241
web scaling factor 651
character codes 738, 742
character encoding 466, 739, 742
character entity reference 742
character format tags 738
character formats. *See* formats
character patterns
See also search expressions; wildcards
character sets 480, 667, 738, 742, 761
character strings 845, 855
character strings. *See* strings
characters
See also escape characters
adding to date and time values 143
adding to numbers 141
browser scripting controls and 643, 644
defined 761
encoding restrictions for 621
generating RTF output and 345, 737
grouping alphabetically 90
including in search expressions 458, 459
including in URLs 466
inserting overflow 144
matching specified 707
naming conventions and 25, 47, 77
search expressions and 839
searching for 706, 707, 710
type declarations and 853
viewing truncated data and 145
Chart background setting 240, 241
Chart border setting 240, 241
Chart Builder
adding drop lines with 259
adding high-low lines with 259
adding tick marks with 244
adding trendlines with 263
adding up or down bars with 258, 266, 267
arranging category markers from 255
arranging intersecting axes and 245
arranging labels with 248, 255
changing charts from 237
changing line styles from 246
connecting data points with 257
creating hyperlinks with 269
creating legends with 241, 243
creating titles with 238, 239
generating sample data with 270
grouping data with 230, 231, 232, 234, 236
labeling axis values with 249
labeling data points with 252, 253, 254
labeling data series with 250, 251
opening 218
overview 196, 218
plotting empty values with 256, 257
scaling axes values with 246
selecting chart type from 206, 219
selecting fields from 220, 222, 223, 225
selecting formatting options in 240, 241, 242, 248
setting pie sector attributes with 208, 267, 268, 269
setting series colors with 250, 251
sorting data with 229
Chart command 218

chart controls 214, 218, 237, 762
See also charts
Chart Fonts page (Options) 611
chart layers
adding drop lines to 260
adding high-low lines to 259
adding to charts 219
adding up or down bars for 259
changing data series for 251, 252
connecting data points for 258
creating legends for 243
formatting bars in 266
formatting data in 228, 229
formatting labels for 248
generating sample data for 270, 271
generating summary data for 227, 229
labeling axes values for 248, 249
labeling data points for 253, 255
overview 204
positioning data points for 255
reformatting legends for 243
restrictions for 205, 206
scaling values in 246
selecting bar spacing for 266
selecting data for 206, 213, 226, 228
setting number of minor ticks for 247
Chart Title dialog box 239
chart titles 239
chart types 198, 206, 218
charting engine 271
charts 761
See also specific type
adding legends to 237, 241, 243
adding titles to 238, 239
adding trendlines to 260, 263
changing 237–238
changing border attributes for 240, 241
changing fonts for 239, 242, 248, 254
changing properties for 237, 238
creating 218–219
defined 761
displaying 271
embedding fonts for 476
formatting 239–241
generating sample data for 237, 270–271
grouping categories for 221
grouping data for 230–232, 235–236
grouping data series for 222
labeling values in 247, 248, 249
linking to reports 214–215, 237, 269–270
overview 196, 197
plotting empty values for 256–257
plotting missing values for 232, 234, 256
rendering platform-specific 271
resizing 237
scaling values in 246–247
selecting data for 206–213, 219–229
setting fonts for 611–613
sorting data for 221, 229–230
check boxes 510, 518
checking user-entered values 526
checkpoints (report generation) 678
Choose Control Type dialog 103
Choose Database page (Report Wizard) 12
Choose Field dialog 111
Choose Layout Style page (Report Wizard) 17
circle controls 150
class association information 763
class declarations 762
class hierarchy 762
Class Hierarchy command 55
class hierarchy window 47, 55
class libraries 129, 616, 624
Class Name dialog 47
class names 25, 416, 586, 763
Class page 763
Class page (Properties) 106
class representation files 763
Class Representation files. *See* apr files
class variables 763
See also instance variables; variables
classes 762
See also Actuate Foundation Classes
adding to libraries 624
associating with toolbox 623–626
changing control types and 106
customizing properties for 559, 562
customizing structures and 568
defined 762
developing with 60, 62, 692
displaying 55
generating XML data and 665, 666
monitoring method calls for 492, 495

classes *(continued)*
referencing components and 584, 586
removing from libraries 129
searching for multiple instances of 444
searching over multiple 445
setting default fonts and 610
viewing methods associated with 492
viewing properties associated with 52
Classes page (Options) 623, 624, 625
Clear All Breakpoints command 487, 494
client file systems 572
Clip property 145, 338, 646
Clipboard 696, 763
See also Scratch Pad
clipping browser output 645, 653
clipping data 141, 145
ClipToControlSize value 646
Close All Panes command 43
Close Design command 59
close values, plotting 203, 212, 226
closing
confirmation messages 6
report designs 59
windows 43, 717
cluster environments 479
code
See also Actuate Basic; source files
changing fonts for 610, 611
controlling execution of 494–496
creating Flash objects and 649
customizing reports and 60, 61
debugging 39, 484, 493, 495
displaying 493
editing 621
generating 58, 486, 494
grouping data and 93
removing breakpoints from 487, 494
setting breakpoints for 494
setting encoding options for 621
stepping through 487, 489, 495, 498
testing and debugging 780
verifying input values with 526, 527
viewing browser scripting 647, 653
viewing errors in 487
viewing methods in 489
writing for web browsers 644, 650
code mapping 144
code points 764
collapsing
parts of a report 45, 95
table of contents 690
color chooser 764
color names 741
color palettes 147
COLOR parameter 741
Color property
borders 147
line styles 151
color selector 240, 241
color tables 736, 738
colors
See also background colors
applying to fonts 610, 611, 741
applying to shadows 150
building browser scripting controls and 653
building charts and 204, 239, 250
building cross tabs and 330
conditional formatting and 155
customizing 147
entering values for 147
laying out reports and 183
selecting default 251
setting dynamic text 362
setting transparent 184, 654
Colors page (Chart Builder) 267, 268
Column Editor 765
column headings
adding 96, 106–109, 306
compared with group headings 109
creating cross tabs and 296
displaying computed values in 312
displaying static text in 312–314
repeating for multiple pages 107
setting border properties for 327
column keys 765
See also cross-tab reports
column names 405
See also column headings
ColumnHeadingsBorder property 327
columns 764
See also fields
adding background colors to 330
adding borders to 327, 329

applying conditions to 74, 405
building cross tabs and 314, 331
building queries and 86
charting data and 206, 207, 208, 210
defined 764
grouping on 75, 83, 86
inserting summary data in 16, 76
referencing in link expressions 454
resizing 326
selecting 13, 74
sorting on 17, 124, 125
Columns page 765
combo boxes 648
comma (,) character
finding 707
search expressions and 705
comma-separated values (CSV) files 766
COMMENT tag 740
comments 740
common fields 520
Commonly used setting (New Property) 561
communications protocol 828
communications protocols 456
Compact command 138
compacting components 138
company logos 151, 184
comparisons
exploding pie sectors based on 268
listing reports and 298
plotting numeric 202
summary data 118, 120
compile 766
compiler 485
compile-time errors 484
compiling 27, 59
Component Drop dialog 587
component icons 45
component libraries 766
accessing 572
adding components to 576, 578
adding to configuration files 589
changing 584, 585, 587, 591
creating 574–578
deleting 585
described 572
displaying 574, 590
including in designs 573, 579–584, 588
limiting size of 575
naming 576, 577
protecting 591
publishing to 34, 68, 586–587
referencing components in 584, 587
restoring 585
restricting changes to 591
saving 576, 577
selecting 580, 581, 582
selecting components from 47, 584
setting as read-only 591
setting search paths for 590, 616
specifying default 583
subclassing components in 585–586
Component Library setting 576, 627
component names 9, 26, 45, 47, 608, 609
component palettes. *See* toolbox
component reference property 766
Component Search command 48
Component Search dialog 48
Component Search Results command 52
Component Search Results window 52
components 766
See also specific type; structural components; visual components
adding
to designs 44, 46, 584
to libraries 576
to report sections 404–405
to toolbox 621, 622, 623
adding borders to 147
aligning 134, 137–138
applying formats to 76, 155
arranging 134, 138, 377, 613
assigning group names to 622
associating classes with 623
associating images with 623
centering in frames 138
changing 584, 587
changing properties for 56, 586
compacting 138
controlling expansion of 378
copying 57
customizing 132
deselecting 135
determining relationships among 450
developing for 60, 61, 62

components *(continued)*
displaying 44
dynamically sizing 376
generating table of contents from 466
layering 139
linking to 457
mapping XML properties to 665, 666, 667, 669
missing in report structure 585
moving 135
naming 47, 48
organizing 45
overview 32, 33
prompting for names 609
providing balloon help for 440
publishing 34, 586–587
referencing 47, 584, 587
renaming 25, 26, 576
resizing 136, 613
reusing 34, 558, 572, 586
searching for 48, 50, 53, 442, 444, 445
selecting 134, 572, 575
setting background colors for 147–148
setting properties for 56, 60, 522, 565
spacing evenly 137, 139
storing temporary 127
tracing variables for 497
updating 47
viewing classes associated with 55
viewing properties for 52
computed columns
adding to reports 75, 100, 102, 104, 431
aggregating values for 116
creating group headings from 110, 111
verifying values in 128
Computed Field command 75, 102, 111
computed fields 312, 767
See also computed columns
defined 767
concatenating RTF tags 736
concatenation 312, 578
concatenation character 454
concrete base class 767
conditional data selection 74
conditional expressions 405
See also Boolean types; Boolean values
conditional formats 767
See also formats
conditional formatting 31, 154, 158
Conditional Formatting dialog 134
conditional section components 429
conditional section icon 402
conditional sections 767
adding components to 405
aggregating data and 120, 406
described 402
displaying properties for 406
nesting 428
placing in reports 427, 428
specifying conditions for 405–406
conditional subreports
creating 427, 428
example for 403
running 433
conditions
adding to conditional sections 405–406, 428
applying to formats 154–159
filtering data and 74
removing parameters for 522
setting grouping 93
setting search. *See* search conditions
Conditions page 768
Conditions page (Query Editor) 74, 522
Configuration Console 768
configuration files 768
accessing 588
accessing templates and 71, 628
adding libraries and 72, 573, 583
creating reports and 4, 5, 72
including in designs 588
localizing formats and 142
opening 5
setting search paths and 590, 617
specifying 5, 72
configurations
accessing data and 73
accessing sample 5
replacing default 72
saving toolbox 624, 625
confirmation messages 6
connection components 413, 587, 602, 769
See also connections

connection definition files 602
Connection objects 769
Connection property 768, 769
Connection slot 768
connections 769
  accessing sample database 12
  adding to parallel sections 413
  building 12, 73, 504
  customizing 12
  grouping with data streams 587
  publishing 587
  selecting database 12
  sharing 413
  testing 596
consistent report design 770
  See also report designs
constants 569, 570, 770
constructor code 770
containers 497, 770
  See also frames
containment 770
containment hierarchy 770
content. *See* structured content
Content property 771
Content slot 96, 351, 402
context block (scripting control) 643, 644
context menus 608
Context path field (iServer Profile) 595
context paths 595
Continue command 487, 496, 499
control codes 339, 341
controlling dynamic sizing and positioning 376
controlling page breaks 358, 361, 365
controlling program execution 494
controls 771
  See also visual components
  adding 45, 104, 365, 376
  aligning 348
  applying fonts to 473, 476, 610
  applying formats to 140, 147, 158–159
  centering 388, 389
  changing data type for 105, 106
  changing relative position of 371
  changing values in 105
  combining with cross tabs 331
  controlling dynamic sizing for 378, 382
  controlling position of 378
  copying 105
  creating type-specific 104
  customizing 623
  defining overflow characters for 141, 144
  described 97
  displaying properties for 106
  displaying sample data in 608
  enabling horizontal expansion for 380, 382
  enabling vertical expansion for 378, 380, 382
  generating dynamic content for 352
  generating output-specific 480, 481
  generating table of contents from 466
  grouping 98, 350, 362
  linking to 458
  maintaining spacing among 348
  mapping to XML attributes 666
  monitoring object reference variables for 496
  moving 348
  naming 25
  overlapping 473
  renaming 25
  repositioning 380, 382, 387–389
  resizing 26, 383–387
  searching for 447, 448, 449
  setting properties for 376, 564
  splitting across multiple pages 383
  stacking 480
<conversation> tag 663
conversation element 663
conversions
  scripting controls and character 643, 644
  values to strings 531, 578
  XML data formats and 662, 663, 666
converters 771
  See also Excel converter; XML converter
Copy button 697
Copy command 58
copying
  autoarchiving rules 598
  components 57
  controls 105
  data 696–697
  file properties 597
  from Scratch Pad 128

copying *(continued)*
images 696
privileges 594, 598
copyright character tag 743
copyright symbols 743
Count function 114
Create New Report dialog
creating blank report designs and 73
creating listing reports and 10
hiding items in 626, 627, 628
selecting options in 35
creating
ad hoc parameters 430
background colors 240, 241
back-up files 746
balloon help 440
browser-specific menus 648, 654, 656
charts 196, 198, 206
check boxes 510, 518
class libraries 624
component libraries 574–578
component references 584
computed columns 75, 100, 102, 431
cross tabs 303–325
custom font files 477, 479
data groups 75, 83–87
data structures 566–568
date and time groups 236
detail reports 425, 460, 548, 549
DHTML forms 654–656
drop-down lists 511, 512
error-checking routines 526
Flash objects 649
function libraries 578–579
group headers 96, 98
grouped reports 10, 84
HTML forms 648–649
hyperlinks 214, 269–270, 454, 460
Information Console headlines 507
joins 21
left and right page layouts 187–189
link fields 426
listing reports 10–18
master reports 400, 401
multiflow page layouts 408
page headers and footers 96, 98
properties 558, 561, 562
property groups 561
queries. *See* queries
radio buttons 511
report designs 30–31, 66, 67, 68, 72, 73, 100
report executable files 59
report object value files 524
report titles 76
Reportlets 636, 637, 638
reports
from templates 71
guidelines for 66
international considerations for 67
options for 69
platform-independent 662
with wizards 4, 69–70
without wizards 72–77
search expressions 704–708
security IDs 531, 533
server profiles 594–596, 602
source files 621
static groups 235, 236
subreports 400, 410, 417, 422, 427
summary data 16, 76
table of contents 466, 468
title pages 190–193
URLs 454, 455, 465
value expressions 342
virtual security IDs 541–543
criteria. *See* parameters; conditions
cross tabs 772
See also cross-tab controls
adding 299–302, 304
as dynamically sized components 348, 376, 377, 390
building queries for 302, 303, 315
changing border properties of 327, 328
changing data layouts for 310, 311
changing margins for 332
compared to listing reports 297, 299, 308
computing values for 312
creating 303–325
customizing 325–330
defining headings for 306, 312–314
defining multiple columns or rows for 309
developing 298
displaying 299, 310, 331, 333
eliminating blank space in 334–336

formatting numeric values in 329
generating 299, 307, 314, 331
grouping data for 308, 316, 318
limitations of 298, 315
overview 296, 331
placing controls with 331
printing 331–336
resizing 311
resizing columns or rows for 326
retrieving data for 315, 321, 331
retrieving missing groups for 318–325
setting color properties for 330
sorting data for 315
specifying fixed page size for 391
troubleshooting output for 396
truncated data in 326
undoing changes to 298
cross tabulation. *See* cross tabs; cross-tab controls; cross-tab reports
cross-platform applications
designing for 167, 472–473, 479
generating output and 352
rendering charts for 196, 271
XML data formats and 662
Crosstab Builder 303, 305, 319
Crosstab command 304
cross-tab controls 772
See also cross tabs
adding 299, 304
placing multiple 302
subclassing 298
viewing 325
cross-tab reports 331, 333, 334, 772
Crosstab Value Properties dialog 329
CSS attribute 645
CSS files. *See* cascading style sheets
CSS STYLE elements 647
CStr function 531
currency controls 773
adding 129
creating default 623
displaying properties for 26
formatting values 25, 26, 141, 142
Currency data type 773
Currency format type 141
currency symbols 737, 742
currency values. *See* currency controls
current date component 123
current scaling factor 652
current working directory 617
CurrentUserACL variable 536
cursor (hyperlinked fields) 691
cursor (searchable objects) 701, 702
custom browser code
See also browser scripting controls
aligning output 646
clipping output for 645–653
converting to PDF formats 481, 653
debugging 646
displaying 647, 653
example for Internet Explorer 654
generating dynamically 650–653
generating HTML tags for 644
including in designs 642, 645, 648
referencing 648
setting scaling values in 652
supported formats for 642
Custom Browser Code Editor 645, 658
Custom Categories Labels dialog 249
Custom Color button 240, 241
custom data sources. *See* open data access
CustomDHTMLFooter method 655
CustomDHTMLFooter property 649
CustomDHTMLHeader method 655
CustomDHTMLHeader property 648
Customer Orders sample report 10, 27, 84
CustomerOrders.rod 20, 86
customized_fonts.rod 477, 479
customizing
access control lists 543–546
axes labels 249
colors 147
components 132
controls 623
cross tabs 325–330
data 140
data structures 566, 568
database connections 12
fonts 476, 479, 609–611
formats 141, 143
measurement units 615
page layouts 31, 126, 406
page number formats 182
properties 558–565

customizing *(continued)*
report views 710
reports 60, 61
sort keys 93
toolbox 621–624
XML data formats 668

D

data
See also data controls; values
accessing 30, 73
adding to reports 75, 97, 166, 206
analyzing 68, 297
changing cell alignment for 310, 311
changing font attributes for 610, 611
charting 206–213
copying 696–697
creating static groups for 236
customizing appearance of 140
defined 773
displaying 75, 97
formatting 76, 140–145, 158
generating sample 100, 270–271, 608, 645
grouping
charts and 230–232, 234, 235
cross tabs and 308, 316, 318
reports and 10, 75, 81–94
highlighting 147, 150
keeping original order for 230
laying out 94–124
linking to 458, 459, 463
placing for table of contents entries 466
placing in slots 95, 97
plotting range of values for 203
plotting relationships to whole 201
retrieving
for cross tabs 315, 321, 331
from databases 74, 104, 507
from multiple tables 10, 422
from XML documents 662, 664
with queries 13, 31, 46, 84
returning specific values for 504, 506
searching for 701–704, 710
selecting 68, 74, 696
sorting 75, 124–126, 229, 315
specifying overflow characters for 144
summarizing 16, 76
testing output for 128, 129
truncating 141, 145, 338
updating 676
viewing current 59
viewing group keys for 15
viewing relationships 196
viewing trends in 260
zooming in on 695
data adapters 773
data categories
See also charts
assigning values to 221
building axis labels with 249
building chart titles from 238
changing attributes of 237
creating chart layers and 213
displaying markers for 255
generating legends with 241
generating sample data for 270
grouping 221, 230
grouping data points and 196
linking to 215
multiple value expressions as 220
plotting values for 207, 267
reformatting 250
renaming 220
setting distance between 267
sorting 221
spacing options for 265
Data command 21, 37, 74
data connection definition files 774
data controls 97, 774
See also controls; data
data filter components 774
data filters 774, 791
See also sort filters
defined 791
defining parameters as 509
entering conditions for 74
removing parameters for 522
data labels 775
Data page (Chart Builder) 219, 220
data perspective 21, 39
data point markers 257
data points 775
See also charts

adding to line or scatter charts 257
adding to pie charts 254
creating hyperlinks for 269
defined 775
formatting labels for 254
labeling 250, 252, 253
linking to 215
plotting 196, 207, 256
positioning 254, 255
selecting chart layers and 204
data ranges (cross tabs) 321–325
data row components 775
Data Row Editor 776
data row slot. *See* DataRow slot
data rows 775
See also rows
data series 841
See also charts
adding 221, 224, 227
building chart titles from 238
changing appearance of 250
changing color of 250
connecting data points for 257
creating chart layers and 227, 229
defined 841
described 196
empty values in 237, 256
generating legends with 241
generating sample data for 270, 271
grouping 222
labeling 220, 250, 251
linking to 215, 269
multiple value expressions as 220, 223, 250
plotting values for 199, 207, 210, 237
positioning data points for 255
renaming 220
selecting line styles for 252
selecting multiple columns for 210
sorting 221
data sets 296
Data Sorting page (Report Wizard) 17
data source components 776
See also data sources
data source map files 776
data sources 776
accessing 4, 410, 417
accessing data in 30, 73
building charts and 206
building cross tabs and 296, 312, 315
configuring 73
connecting to 73
defined 776
formatting data and 143
generating information objects for 602
laying out reports and 81
returning null values from 681
selecting 11
setting encoding options and 621
Data Sources page (Report Wizard) 11
data stream slot. *See* DataStream slot
data streams 777
See also query data streams
creating reports and 11, 435
grouping with connections 587
publishing 587
data structures
See also source code
creating 566–568
customizing properties and 561, 564
debugging 489, 491
data type aliases 569
data types 777, 855
See also type declaration characters
adding fields and 103, 104
adding static text and 314
associating with controls 104
changing 105, 106, 499
charting data and 230, 236, 250, 251, 252, 267
customizing properties and 558, 561, 562, 563, 564
defined 777
monitoring 491
specifying named values and 569
specifying single values and 566
summarizing data and 16
database 777
Database Browser 21
Database Connection command 73
database connections. *See* connections
Database Field command 99, 111
Database Login page (Report Wizard) 12
database management systems 778
database schemas 837

database statements 844
databases
  accessing 4, 12, 30, 504
  choosing predefined 12
  configuring 73
  connecting to 12, 504
  creating reports with 69, 70, 338
  installing client software for 30
  linking security roles to 544, 545
  placing data from 99
  referencing images in 154
  retrieving data from 13, 74, 104, 422, 507
  selecting columns from 13
  selecting tables from 10, 13
  setting up 81
  viewing directory paths for 21
  viewing fields in 75
  viewing tables in 21, 74
DataRow property 778
DataRow slot 778
DataSource components. *See* data sources
DataStream components. *See* data streams
DataStream property 778
DataStream slot 778
date and time groups 236
date controls 97
Date data type 779
date data types 250
date picker 680
date stamps 182
date time controls 779
Date$ function 102
dates
  charting 212, 226
  displaying 97
  excluding 234
  formatting 142–143, 183, 312
  generating missing values for 319, 324
  grouping values for 91, 93, 316, 318, 321
  printing 334
  selecting 510, 680
DBMS (defined) 778
.dcd files. *See* data connection definition files
Debug command 37
Debug menu 485, 486
debug perspective 39, 485
debugging 780
  defined 780
debugging browser code 646
debugging options 487
debugging reports 484, 692
debugging sessions
  monitoring variables and 496–500
  resuming 487, 495
  running successive 491
  saving parameters for 491
  starting 492–494
  stepping through code and 495, 498
  stopping 496
debugging tools 39, 484, 485, 496
DebugOption property 646
decimal values 141, 615
declarations 780
  See also implicit declarations
declarations section 780
decorative elements 123, 126, 183, 450
default chart titles 238, 239
default colors 251
default component images 623
default component names 47
default configurations 72
default directories 609
default encoding 669
default fonts 271, 610, 611, 613
default formats 25, 76, 140, 222
default layouts 123, 167
default locales 619
default margins 332, 362
default padding 352, 356
default page settings 167
default page styles 165
default parameter settings 676
default printer settings 715
default sort order 15, 124
default unit of measure 614, 615
default values
  page layouts and 31
  report parameters and 509, 510, 513
default zoom setting 695
defaults, restoring 41, 44
DefaultSplitting value 358, 359
degree symbol character tag 743
delays (refreshes) 678
delete privilege 780

deleting
component libraries 585
custom properties 558
directories from search paths 618
master reports 548
parameters 521–523
references to obsolete properties 488
sort fields 125
dependencies. *See* file dependencies
deploying reports 152, 594
derived classes. *See* descendant classes
descendant classes 780
descendant elements (structure parameters) 681
descending sort order 75, 125
Description attribute 630
deselecting components 135
Design command 23, 37
Design Editor 780
Design Editor page (Options) 5, 8, 608
Design element 588
design environment 6, 34, 36, 608
design files 833
design files. *See* report object design files
design options 30, 35, 69
design perspective
customizing 608
overview 38
publishing from 595
viewing libraries in 573, 590
viewing report structure and 6, 23
Design Properties dialog 525, 526, 617
design search path 617
Design Search Path page (Design Properties) 617
design tasks 30, 46, 78
design time 781
design tools 44
design window 5
See also application window
designer options 608, 619
designing interactive web pages 649
designing reports
See also designs
common operations for 30, 46, 78
cross-platform guidelines for 472–473, 479
cross-tab controls and 299–302, 331, 377
dynamic text controls and 338, 365
dynamically sized components and 378, 380
locale-specific environments and 67
output considerations for 173–178
overview 4, 30, 32
page-level security and 532–538
prerequisites for 4, 5
process described 30
programming basics for 60, 575
sample resources for 72
search strategies for 448, 449, 450, 453
XML output and 662, 664, 665
designing subpages 407
designing subreports 400, 401, 403, 433
designs 770, 780, 833
See also page layouts
accessing configurations for 5, 588
adding parameters to 507
adding report components to 6, 44, 46, 584
adding section components to 411, 418, 428, 466
applying fonts to 474, 478, 480
arranging components in 134–140, 613
backward compatibility for 60
building multiple reports and 400, 401, 402
building templates for 71, 73, 628
changing 18, 21, 25, 28, 69
closing 59
creating 66, 67, 72, 73, 100
defined 780
developing 60, 692
displaying 44
embedding browser code in 642, 645, 648
embedding tags in 341
enabling page-level security in 532, 535–538
generating information objects for 602
grouping components in 350
hiding empty slots in 95
including libraries in 573, 575, 579–584, 588
missing components in 585
organizing data in 45, 75
overview 30, 32, 34
placing data in 66, 75, 95, 97, 123

designs *(continued)*
preserving previous versions 746
previewing 18, 20
prototyping 68, 163
referencing components in 584, 586, 587
renaming components in 25, 26
report bursting and 548, 549
resizing components in 136
reusing components in 34
saving 9, 20, 58, 127
selecting components in 135
selecting data sources for 12
selecting database columns for 13
setting default directories for 609
setting search paths for 616
stacking controls in 480
testing 126, 127, 128, 533, 534
upgrading 747
verifying structure of 23, 129
viewing libraries in 574, 590
desktop reporting 475, 479, 676
desktop viewing requirements 472
Destination folder field (iServer Profile) 595
destination folders (publishing) 595, 596, 597
destination URLs 454, 455, 462
detail data 96, 125, 150
detail data fields 17
detail frames 23, 781
detail reports
See also master reports; subreports
adding table of contents to 467
creating 425, 460, 548, 549
displaying contents of 23, 548
displaying properties for 426
generating 505
linking to master reports 421, 426, 520, 549, 551, 553
naming 551, 552
placing data in 96
running 427
sorting data for 125
testing 129
detail rows 17
Detail sample report
embedding in web pages 634
locating searchable objects in 701
opening 686
viewing table of contents for 689
Detail.roi 97, 686
developers 61, 62, 700
developing reports 61, 72, 376, 504, 692
development languages 61, 749, 804, 856
DHTML (defined) 784
DHTML converter
adjusting scaling values and 651
generating HTML code with 644
processing scripting controls 643
setting properties and 646
DHTML format 642
DHTML forms 654–656
DHTML reports 781
See also web pages
applying fonts to 474, 480
converting to PDF formats 653
converting XML formats to 663
creating Flash objects for 649
debugging browser code for 646
displaying 642
enabling page-level security and 530
generating 480, 647
previewing 597
searching 452
DHTML Viewer 352, 473, 480, 599, 782
DHTMLForm scripting control 654–656
DHTMLMenuControl scripting control 654, 656–659
directories
See also paths; search paths
accessing libraries in 590
accessing templates in 630
adding to search paths 617, 618
linking to reports and 456
removing from search paths 618
saving reports and 9
searching 616
setting default 609
storing image files and 152
viewing paths for 21
disabling cascading security 544
display formats. *See* formats; formatting
display names 510, 513, 520
display resolution 473
Display Sample Data setting 100, 608
Display Unit options 615

displaying
access control lists 535
aggregate data 297, 299, 310
application status bar 36
balloon help 440
bi-directional text 341
charts 271
class hierarchies 55
class-specific methods 492
component names 45
component properties 52, 56, 57
component references 584, 586
components 44
computed values 312
context menus 608
cross tabs 299, 310, 331, 333
custom browser code 647, 653
data 75, 97
data groups 95
database fields 21
database tables 21, 74
detail reports 548
DHTML reports 642
errors 487
external functions 572
grid lines 10
images 151
large numbers 145
libraries in designs 574, 590
link expressions 464
literal values 341
member variables 568
method calls 487, 489
overflow characters 144
page layouts 123, 163
PDF output 653
query output 474
query statements 21
related information 691
report designs 18, 20, 44
report layout styles 17
report parameters 519, 521
report structures 6, 23
Reportlets 636
reports 59, 67, 472, 686–694
sample data 270, 608, 645
sample reports 686
scripting control values 652
search results 52, 704, 708, 709
security IDs 535, 536
source code 59, 493
specific parts of a report 688, 690, 691
specific report pages 173, 530, 688
status messages 27
table of contents 689
text 312, 338
toolbars 43
units of measurement 613
variables 487, 488, 491, 498
XML properties 666
XML reports 667
DISTINCT clause 118, 119
distortion 646
Distribute command 139
distributing reports 152, 594
DIV element 635, 644
DLLs 785
DOCTYPE declarations 667
document files. *See* report object instance files
Document Properties dialog 715
document structure information 711
document type definitions 662, 663, 665, 666, 782
documentation xxi, 6, 7
documents. *See* reports
DoNotSplit value 358, 359, 360
dot notation 783
Double data type 783
double quotation mark (") character
as literal value 454, 739
link expressions and 454, 458
search expressions and 459
double quote (") character tags 739, 742
double-precision floating point numbers 783
down bars (charts) 250, 258
downloading report files 548
downloading Reportlets 640
dragging components 46, 58
drawing controls 150, 783
See also line controls; rectangle controls
drivers 4, 783
defined 783
drop lines (charts) 197, 237, 259
drop shadows 148

drop targets. *See* slots
drop-down lists
  displaying property values and 558
  generating items for 511, 513
  providing user with 511, 512
  testing items in 524
DTD (defined) 782
DTD files 662, 663, 665, 666
dual y-axis charts 204, 784
duplex printing 715
duplicate file names 594, 597
duplicate values 118
duplicating toolbox classes 624
dynamic formatting 155
dynamic hyperlinks 784
Dynamic HyperText Markup Language. *See* DHTML
dynamic hypertext markup language. *See* DHTML
dynamic link libraries. *See* DLLs
Dynamic List of Values Builder 514, 515
dynamic page size 785
dynamic parameters
  adding to hyperlinks 462–464
  creating 459–460
Dynamic Size & Position properties group 382
Dynamic Text Control command 338, 341
dynamic text controls 785
  adding 341–342, 351, 366
  adjusting tabbed text in 344, 356
  changing contents of 341
  changing padding for 352
  controlling height of 377
  controlling page breaks for 348, 361, 365
  converting text controls to 370
  displaying reports and 376, 390
  embedding fonts for 476
  formatting 339, 342
  generating content for 342, 347, 352, 356
  generating HTML output and 342, 344, 610, 740–742
  generating PDF output and 610
  generating RTF output and 342, 344, 736–739
  getting font information for 474, 475
  getting tagged text in 340
  hiding 347
  inserting in page footer frames 351
  inserting in page header frames 351
  optimizing output for 341, 352, 354, 357
  overview 338–341
  placing in nested frames 350
  resizing 347, 348
  setting margins for 362–364
  setting maximum heights 348, 351
  setting maximum widths 349
  setting minimum heights 347
  setting static height for 351
  setting tagged text in 340
  spanning multiple pages 358–365
  specifying page size for 392
  troubleshooting output for 372–374
dynamically sized components
  formatting controls and 376
  horizontal resizing and repositioning for 380–382
  overview 376
  repositioning controls relative to 387–389
  resizing controls relative to 383–387
  vertical resizing and repositioning for 378–380

E

e.Analysis 751
e.Analysis for iServer 751
e.Report Designer Professional 752
  accessing options for 608, 619
  application window described 36
  backward compatibility for 746
  basic design concepts for 32, 37
  debugging tools for 39, 484, 485
  default component names and 25
  default page layouts for 167
  default page lists for 165
  default page settings for 167
  design options for 30, 69
  developing with 60–62, 72
  information objects and 602
  opening reports and 602
  overview 4, 34
  personalizing 41, 43, 608
  publishing from 598

restarting 35
restoring defaults 41, 44
sample configurations for 5
saving window layouts for 41
setting search paths for 616–618
setting up sample reports for 8
starting 5
status messages in 27
e.reporting server. *See* iServer
e.reporting technology 787
e.reports 787
See also Actuate Basic reports; reports
e.Reports option 752
e.Reports Page Level Security option 752
earnings reports 68
Edit categories labels setting 220
Edit Property dialog 563
Edit series labels setting 220
Edit slice labels dialog 223
editing. *See* changing
editor page (Library Organizer) 579
editors 59, 578, 610, 621
See also specific Actuate editor
elements 786
defined 786
ellipsis (...) button 786
ellipsis character (…) 145
Ellipsis property 145, 646
Else slot 428
em spaces 739
e-mail 152
e-mail templates 851
embed 786
Embedded property 152, 154
embedding font information 474, 476, 480
embedding images 152, 154
embedding reports in web pages 634, 636, 637
embedding tags in designs 341
em-dash character tag 739
empty design windows
See also blank report designs
empty slots 6, 7, 95
empty strings 654
empty values 250, 256–257, 786
en spaces 739
encapsulation 786
encoding 621, 739, 742, 854
encoding declarations (XML) 667, 669
encryption 510
Encyclopedia volumes 787
accessing 620
linking to 457, 464
publishing to 594, 598–599
setting up server profiles for 595
setting up user accounts for 594
uploading files for 597
en-dash character tag 739
enum data type 569
enum icon 570
enum property 569
enumeration 569–570
environments 67, 151, 352, 473, 825
equal sign (=) character
searching for 707
equals operator 705
erdpro command 746
error messages
scripting controls and 646
searching and 448, 449, 453
selecting 487
error-checking routines 526
errors
See also error messages
browser scripting controls and 646
debugging and 39, 484
displaying 487
exceeding stack limit and 490
expressions and 128
testing report layouts and 128, 129
escape characters 643, 644, 787
escape codes 51
Essential setting (New Property) 561
etc directory 479
Euro currency symbols 742
event handlers 647, 788
events 440, 788
examining variables 488, 491, 496, 498
example reports. *See* sample reports
Examples directory 549
Excel spreadsheets 474, 475
exceptions 788
exclamation point (!) character
as literal value 707

exclamation point (!) character *(continued)*
HTML tags and 742
executable files 788
bundling with reports 506
compiling 27
creating 59
embedding fonts in 474, 476, 477, 480, 610
generating reports from 459, 548, 676, 711
including large libraries and 575
linking to 456
opening 676
publishing 594, 598
running 676, 677
specifying encoding for 621
upgrading 747
executing queries 548
executing reports 27, 59, 369, 676, 717
executing subreports 416, 419, 433
expanding
parts of a report 23, 45, 95
table of contents 689
expense reports 68
Expert setting (New Property) 561
Expert settings (Properties page) 376
experts. *See* wizards
exploding pie sectors 268
exponential trendlines 262
Export to Information Object button 603
Expression Builder
charting data and 220, 223, 226, 227, 229
displaying external functions in 572
opening 102, 220
expression builder 788
defined 788
Expression Builder button 220
Expression setting 243
expressions 788
See also Expression Builder; formulas
adding nonprinting ASCII codes to 708
adding to group headings 110, 111
aggregating data and 754
aggregating data and. *See* aggregate expressions
automatically entering parts of 102
building chart layers and 227, 229
building charts and 210, 220, 222, 223
building label text and 243, 249
converting to strings 531
creating computed fields and 101, 102
creating conditional 405
creating hyperlinks and. *See* link expressions
creating search 704–708
creating security IDs and 531, 533
creating value. *See* value expressions
customizing properties and 564, 565
defined 788
defining category keys and 221
defining series keys and 221, 224
determining pie size and 208, 209, 210, 223
displaying results of 100
errors in 128
formatting data with 312
manipulating numeric values with 815
manipulating string data with 845
placing in function libraries 578
returning Boolean values from 759
returning stock price data with 213
reusing 578
searching with. *See* search expressions
extended price 102
ExtendSearchPath function 617
Extensible Markup Language. *See* XML
extensible markup language. *See* XML
external applications 696
external functions 572
external libraries 582
external source code 642
ExternalInterface class 649

F

–f option (erdpro) 747
FACE parameter 742
factory 789
Factory service
designing cross-platform applications and 472
generating ACLs and 531, 544
getting font information for 473, 474, 475
monitoring 495
registering custom fonts for 479
Factory window 676, 678
False values. *See* Boolean values

fatal errors 490
fetch 789
Field list (Chart Builder) 220
fields 789
  See also columns; computed fields
  adding to cross tabs 296, 303, 306, 309
  adding to reports 46, 75, 98, 99, 104
  building axis labels from 249
  calculating values for 100
  creating group headings from 110, 111
  defined 789
  displaying database 21
  displaying multiple text lines in 141
  grouping aggregate data and 308
  grouping report 75
  highlighting 147
  linking on common 426, 520
  removing from search conditions 704
  searching on 700, 701, 702
  selecting as category keys 221
  selecting as chart layers 227, 229
  selecting as sector keys 223
  selecting as series keys 221, 224, 227, 229
  selecting database 69, 70, 74
  selecting for specific chart type 220, 222, 223, 225
  summarizing values from 76
  viewing list of 75
Fields command 75, 99
Fields List 790
Fields list (designer) 46, 75, 99, 623
Fields To Sort By column (Sorting page) 125
file attribute 456
file dependencies 790
file descriptions 598
File menu 620
file name extensions 667
file names 504, 597
file systems 456, 464, 572, 581, 686
file types 790
FileNameExp property 154
files
  See also specific type
  building information objects and 602
  building reports and 27, 58, 59
  closing 59
  defining hyperlinks for 456, 457, 461, 462
  downloading 548
  embedding font information in 610
  limiting size of 548
  locating configuration 5
  naming 20, 77
  opening 60, 620, 686
  publishing 594, 597, 598
  saving 609
  setting default directories for 609
  setting recently opened list for 620
  setting search paths for 616–618
  upgrading multiple report 746
  uploading 597
Fill Color setting (charts) 250
fill colors 147, 148, 149, 151
FillColor property 149, 151
FillPattern property 646
filter components 774
filter conditions 74
filter options 791
filtering
  database records 74
  methods 492
  properties 56, 561, 565
  structure members 561
  summary data 118
filters 774, 791, 842
  defined 791
  defining parameters as 509
  entering conditions for 74
  removing parameters for 522
financial statements 68
finding
  data. *See* searching
  searchable report fields 700
Fixed format type 141
fixed-width text 741
Flash object 792
Flash objects 649
Flash Player 649
Flash software 649
flat file data sources 11
flat files 792
Flip axes option 219
floating point controls 792
floating point numbers 783, 841
floor colors (charts) 241

flow 793
See also flow components
adding borders to 183
adding watermarks to 184
assigning to subreports 409, 415, 437
defining multicolumn 406
laying out multiple 408, 415
placing page components and 179
renaming 415
resizing 172
specifying background colors for 183
flow components 166, 793
flow names 416
FlowName property 409, 437
folders 798
linking to 457
placing detail reports in 551, 553
publishing to 595
report bursting and 549
uploading to 597
font attributes
adding label controls for 478
applying from toolbar 140
changing 610, 611, 612
dynamic text controls and 338, 738, 741
Font dialog 611, 612
font files
accessing master 475
creating custom 477, 479
defining font information in 476, 477
registering 479
saving 479
font information 472, 474, 475, 610
font names 742
Font page (Format) 132
Font property 476, 477
font style settings. *See* font attributes
font substitution 474, 480
font tables 736, 738
FONT tag 344, 741
fonts 793
See also font attributes
adding to charts 271, 611–613
adding to designs 474, 478, 480
applying conditional formatting to 154
changing application 613
changing chart 239, 242, 248, 254
cross-platform considerations for 473, 479
customizing 476, 479, 609–611
embedding 474, 476, 480, 610
generating dynamic content and 353, 356
generating HTML output and 344, 741
generating RTF output and 345, 738
getting information for 474, 475, 479
installing 271
not available 271
resetting default 611, 612
selecting 140
setting colors for 610, 611, 741
setting dynamically 476, 480
setting properties for 478
setting size 345, 742
substituting 474, 480
Fonts page (Options) 610
Footer command 180
footers 793
adding dynamic text controls as 351
adding to reports 180
creating 96, 98, 179
defined 793
designing for 27, 68
printing 173
ForcePageHeightToFit property 378, 391, 392
ForcePageWidthToFit property 378
Forecast sample report 400, 433–438, 468
Forecast.roi 690
foreground colors 741
See also colors
FORM element 648, 655
form letters 338
Format dialog 132
Format function 102, 182
Format menu 76, 133, 137
format patterns 142, 143, 182
Format property
cross tabs 329
data controls 140, 141
date controls 143, 183
text controls 143
Format setting 26
format symbols 142, 143
Format toolbar 76, 132, 140, 610
format toolbar 794
Format$ function 102, 312

formats 793
See also formatting; output formats; styles
adding dynamically sized components and 377
applying conditions to 154–159
building document type definitions for 662
changing 25, 76, 155, 181, 183
copying components and 57
creating language-independent reports and 662
creating platform-independent reports and 662
customizing 141, 143, 182
defined 794
displaying multi-style 338, 341
labeling charts and 250, 251
preserving 746
setting at run time 506
specifying expressions for 312
updating 140
FormattedPageNumber value 182
formatting
axes values 205, 237
chart areas 239–241
chart axes values 243, 244
chart labels 222, 248, 254
chart titles 239
cross tabs 312, 325, 326, 329
currency controls 26, 141
data 76, 140–145, 158
date and time values 142–143
dynamic text controls 339, 342
numbers 141, 252, 329
reports 31, 68, 132, 154
text 76, 143, 222, 352
formatting options 76, 132, 140
formulas. *See* expressions
frames 794, 798
adding borders to 147
adding data controls 45, 98, 365, 376
adding drop shadows to 148
adding graphical elements to 150, 151, 153
adding watermarks to 184
adjusting spacing in 146
arranging components in 138, 377
browser scripting controls in 645, 647
centering components in 138
conditionally displaying 405
converting to HTML forms 642
cross-tab controls and 299, 302, 331, 335
dynamic text controls in 338, 341, 350, 351
dynamically sized components and 376, 377, 378, 380
dynamically sizing 378
eliminating extra space in 348
extracting as Reportlets 638
grouping controls in 362
keeping contents from splitting 361
mapping to XML elements 666
monitoring object reference variables for 496
nesting 350
placing in conditional sections 428
placing in group sections 98
repositioning controls in 380, 382, 387, 388
resizing 136, 146, 348, 350
resizing controls in 383, 384, 386
searching in 449, 450
setting margins for 365
splitting across multiple pages 358–365
framework 794
FROM statements 22
See also SQL statements
FTP servers 456
fully qualified names 457
function libraries
accessing 573
adding to configuration files 589
creating 578–579
displaying 574, 590
including in designs 573, 579–584
naming 578
overview 572
protecting 591
restricting changes to 591
saving 578
selecting 580, 581, 582
setting as read-only 591
setting location of 590
functions 794
See also methods; nested functions
adding to group headings 110, 111
adding to multiple designs 572

functions *(continued)*
aggregating data and. *See* aggregate functions
calling browser code 642
creating Flash objects and 649
defined 794
defining link fields and 426
generating XML documents and 668
mapping security roles with 545
returning values from 128
fundamental data types. *See* data types

G

General Date format type 142
General format type 141
General page (Options) 619, 621
GenerateXML method 666, 669
GenerateXMLDataFile method 669
generating
access control lists 531, 544, 545
cross tabs 299, 307, 314, 331
DHTML reports 480, 647
dynamic text content 341, 347, 352, 353, 356
HTML tags 644
information objects 602–606
PDF files 480
report object instance files 27, 548, 676
reports 58, 341, 352, 676
sample data 100, 645
source code 58, 486, 494
table of contents 466, 468
web browser output 650, 651
XML output 664, 665, 668
generation requests 504
GetAppContext function 651
GetCurrentPageACL method 535
GetFullACL method 543, 544
GetKeyValue method 93, 94
GetReportContext function 650, 651
GetReportData directive 639
GetReportScalingFactor function 652
GetTaggedText method 340
GetText method
browser scripting and 650, 653
security IDs and 536
getting started 35
GetUserACL method 536, 541, 542
GetUserAgentString method 651
GetXMLText method 668
GIF files 151
global browser code 642, 648
global name space 795
global reporting environments. *See* locales
global scope 838
global search path 617, 618
Global Search Path page (Options) 618
global variables 488, 795
See also variables
GlobalDHTMLCode property 648
Go To button 709
Go To command 688
Go To Page button 688
Go To Page dialog 688
gopher protocol 456
grand totals 16
grandchild classes. *See* descendant classes
grandparent classes. *See* ancestor classes
GrantExp property
customizing ACLs with 544, 545
defining security IDs with 531, 544
linking security roles with 544
setting 533
specifying access restrictions with 541
graphical controls 150–151
See also image controls
graphical tools 35, 36, 37, 60
graphics 473, 649
See also images
graphs. *See* charts
gray bars (Layout window) 44
greater than (>) character tag 742
greater than operator 705
greater than or equal to operator 705
grid lines
in charts 197, 237, 244, 245
in report designs 10, 613
grids 795
group bursting 549
group bursting example 552–553
GROUP BY clause 120, 121
See also SQL statements
Group By page 795

group footers 96, 98
See also After slot
group headers 96, 98, 797
See also Before slot
group headings 99, 109–113
group indicator tag 736
group keys 15, 25, 466, 796
group names 519, 622
group of values 558
group range and span (charts) 232, 233, 234
group section components 33, 87, 796
group section icon 402
group sections 796
adding summary data to 97
adding to reports 87, 98
aggregating data and 114
applying custom sort keys to 94
creating column headings for 106–109
creating group headings for 109–113
described 402
displaying properties for 93
expanding 23
generating table of contents from 466
hiding 608
inserting page breaks in 122
linking from 421
placing cross-tab controls in 300
placing data in 96
searching 447, 449
group/totals reports. *See* grouped reports
grouped reports 796
See also listing reports
adding database fields to 100
adding group headings to 109–113
aggregating data and 114–118
bursting example for 549, 552
creating 10, 84
defined 796
developing table of contents for 466
grouping data for 83–87, 88, 93
highlighting information in 117
labeling data columns in 107
laying out 80
multiple levels of groups in 95
omitting headings from 109
overview 82
placing data in 94–124
setting page breaks for 122
sorting data for 124–126
testing layouts for 126, 127, 128
viewing structures 87, 95
Grouping and Sorting dialog 86, 98, 125
Grouping and Sorting tool 83
grouping by interval 89–93, 318
grouping conditions 93
grouping data
See also group sections; grouped reports
charts and 230–232
cross tabs and 308, 316, 318
reports and 10, 75, 81–94
Grouping Interval setting 93
grouping intervals 91, 230, 232
grouping on sort keys 93–94
grouping options 231, 318
Grouping Options button 15
Grouping Options dialog 15, 93, 125
Grouping page (Chart Builder) 231, 234
Grouping page (Crosstab Builder) 318, 319, 323
Grouping page (Grouping and Sorting) 86, 92
Grouping page (Sorting and Grouping) 75
grouping parameters 509, 510, 519
grouping related controls 350, 362
grouping units 230, 232
GroupInterval property 91, 93
GroupOn property 91, 93, 94
GroupOn<time period> value 93
GroupOnCustom value 93
GroupOnInterval value 93
GroupOnPrefix value 93
groups 14, 81, 710
See also grouped reports; grouping
defined 795

H

Halt command 487, 496
hand cursor 691
handles (sizing) 26
Having page 797
HEAD element 647
Header command 180

headers 797
adding images to 151
adding to reports 179, 180
adding visual dividers for 150
creating 96, 98, 351
defined 797
designing for 68
printing 173
headings. *See* column headings; row headings
headlines 507
Height property
cross tabs 326
dynamic text controls 347, 348
page size 168, 169
help
See also balloon help
including for searches 448
overlapping controls and 473
providing balloon 440
hexadecimal numbers 797
hidden properties 562, 565
Hidden setting (New Property) 562, 565
Hidden value (CSS attribute) 646
hiding
application windows 41
axis labels 248
dynamic text controls 347
grid lines 10
group sections 608
items in Create New Report 627
legends 241
parameters 520
report parameters 427
Reportlet content 636
slots 7, 95
status bars 36
tick marks 244, 245
toolbars 43
hierarchical structures 33
hierarchy 797
high values, plotting 203, 204, 212, 226
Highlight Objects button 692
highlighting aggregate data 117
highlighting data 147, 150
high-low lines (charts) 197, 250, 259
high-resolution monitors 473
home folder 798
Horizontal property 646
HorizontalOverlap property 333
HorizontalPosition property 382, 389
HorizontalPositionFrameCenter value 389
HorizontalPositionFrameLeft value 389
HorizontalPositionFrameRight value 389
HorizontalSize property 349, 378, 382
HorizontalSizeFrameRelative value 378, 382, 387
HorizontalSizeRelative value 349, 378, 382, 384
HTML (defined) 799
HTML documents. *See* HTML reports
HTML elements
See also HTML tags
HTML extension 784
HTML formats 340, 342, 355, 742
HTML forms 642, 648–649
HTML frames 798
HTML headers 740
HTML lists 344
HTML reports
See also web pages
creating Flash objects for 649
dynamically resizing elements in 652
formatting dynamic text controls for 340, 342
generating tags for 644
importing Reportlets and 638, 640
misaligned text in 610
optimizing output for 355
HTML tags
adding dynamic text controls and 342, 344, 740–742
assigning literal values with 341
creating Reportlets and 635, 638
displaying as literal text 342
generating 644
HTTP (defined) 800
http attribute 456
HTTP requests 650
HTTP servers 456
HTTP service 595
hypercharts 798
Hyperlink Builder 270
Hyperlink Builder button 270

hyperlink buttons 690
Hyperlink page (Chart Options) 269, 270
hyperlinked fields 691
hyperlinks 799
  See also dynamic hyperlinks; static hyperlinks
  accessing objects from 695
  adding dynamic parameters to 462–464
  adding to charts 214–215, 269–270
  adding to master reports 549, 551, 553
  changing chart 237
  creating 454, 460
  defining for temporary reports 456
  defining platform-independent 464
  defining specific destinations for 457, 459
  defining URLs for 454, 455
  enabling in Reportlets 640
  including literal text in 454, 458
  moving through reports with 690–691
  multiple search conditions and 458
  report bursting and 549
  returning in search results 447
  running reports and 460, 462
  setting up Information Console 465
hypertext markup language 642
  See also HTML reports
HyperText Markup Language pages. *See* web pages
hypertext markup language. *See* HTML
hypertext transfer protocol. *See* HTTP
hyphen (-) character
  finding 707
  non-breaking 739
  search expressions and 705
hyphen (-) character tag 742

I

icons, as labels 151
identifiers 800
  See also unique IDs
IfExp property 405, 406
IFRAME element 634, 638
illustrations (documentation) 6, 7
Image command 153
image controls 97, 800
  See also drawing controls; images
image files 800
  referencing 152
  selecting 153, 184
  setting search paths for 616
  types supported 151
ImageDesignTime property 152
ImageFactoryTime property 152
images 800
  See also image controls
  adding 151–154, 184
  associating with components 623
  changing 152
  copying 696
  creating animated 649
  displaying 151
  drawing 150
  embedding fonts for 476
  embedding in designs 152
  generating output and 473, 480
  placing in background 166, 184
  XML converter and 662
ImageViewTime property 153
implicit declarations 800
inches 614, 615
Include Library dialog 581, 582
Include Module dialog 581
incorrect values 485
indenting text 345, 737
indexed searches 442, 839
information 30, 31
  See also data
Information Console
  creating headlines for 507
  defining URLs for 465
  published reports and 594
Information Object Designer 602
information object files 801
information objects 602
  defined 801
inheritance 801
inherited methods 492
inheriting security IDs 543, 544
inner joins 802
  See also joins
InnerBorder property 327
input
  defining parameters for 504, 509

input *(continued)*
prompting for 506, 518
providing list of choices for 511–518
saving parameters for 524
suppressing prompting for 526
testing parameters for 523
verifying 526–527
input sources. *See* data sources
Insert menu 46
inserting page breaks 420
installation
font files 271, 477
third-party software 30
instance variables 488, 491, 802
See also class variables; variables
instances. *See* objects; report object instances
instantiation 802
integer controls 97, 141, 802
See also numeric controls; numbers
Integer data type 803
integers 803, 811
interactive viewing events 440
interactive web sites and applications 649
interface (e.Report Designer) 36
interfaces 803
defined 803
Internal Basic source encoding options 621
international reporting environments 67
See also locales
internationalization 803
See also locales
Internet Explorer
See also web browsers
clipping output for 645
creating scripting code for 644, 654
generating output for 651
Interpolate setting 256
interpolating missing chart data 256
invalid values 526
inverted exclamation mark 742
inverted question mark character tag 743
investment reports 338
IO Design perspective 803
.iob files. *See* information object files
iServer 750
creating profiles for 594–596, 602
generating large reports and 548
getting privileges for 531
publishing to 594
report bursting and 548
iServer Configuration Console 722
iServer field (iServer Profile) 595
iServer Profile dialog 594
iServer System options. *See* BIRT iServer System options
iServer System. *See* BIRT iServer System
iServer. *See* BIRT iServer
IsFrameDecoration property 383, 384, 386, 387
italic text 738, 741
iterators 490

J

J2EE environments 804, 805
J2SE environments 804, 805
Japanese currency symbols 743
.jar files. *See* Java archive files
Java. *See* Java programming language
Java 2 Enterprise Edition. *See* J2EE environments
Java 2 Runtime Standard Edition. *See* J2SE environments
Java applets 642
Java archive files 804
Java archives. *See* .jar files
Java Database Connectivity. *See* JDBC
Java Development Kit. *See* JDK software
Java interfaces 803
Java Naming and Directory Interface 805
Java Native Interface 805
Java Object Interface 805
Java programming language 804
Java programs 804
Java Virtual Machines. *See* JVMs
JavaScript 635, 642, 649, 806
JavaScript error messages 646
JavaScript functions 464, 647, 649
JavaServer Pages. *See* JSPs
JDBC (defined) 804
JDK software 804
JNDI (defined) 805
JNI (defined) 805
jobs 548

JOI (defined) 805
join conditions 806
joins 21, 806
joint data sets 802
JPEG files 151
JSPs 806
jumps. *See* hyperlinks
justification (data alignment) 140, 737
JVMs 806

K

Keep original order setting 230
key fields. *See* sort fields
Key property 88
keyboard shortcuts 36, 37
keys. *See* group keys; sort keys
keywords 215, 807

L

Label command 107, 126
label controls 807
  See also labels
  adding font information to 476, 477, 478
  creating access control lists with 535
  creating column headings with 107
  displaying literal values with 126, 341
  displaying security IDs with 536
  placing data values in 105
  setting page numbers in 538
labels 775
  See also label controls
  adding images as 151
  adding to charts 247, 249, 252
  adding to cross tabs 314
  adding to reports 97, 107, 117
  arranging chart 248, 255
  changing 107, 237, 250, 251
  changing font attributes for 611
  creating expressions for 243, 249, 250
  creating for static groups 236
  customizing 249
  displaying addresses and 338
  displaying literal text and 105
  formatting chart 222, 248, 251, 254
  formatting numbers in 252
  setting fonts for 610, 611
  setting orientation of 248
Labels page 248, 249
Landscape orientation 18, 170
language-independent formats 662
large libraries 575
large numbers 145
large reports 505, 548, 549, 687, 714
layering components 139
Layers command 140
laying out data 94–124
laying out reports 31, 38, 68, 80–81, 163
  See also page layouts
laying out subreports 406, 408, 414, 436
layout editor 808
  defined 808
layout options 17
layout painter. *See* Design Editor page
layout pane. *See* Layout window
layout styles 17
Layout window 7, 18, 44, 163
  See also layout editor
layouts 807
  See also designs
leading spaces 454, 458, 459
left and right page layouts 187–189
left outer joins. *See* outer joins
Left property 362
left/right page lists 165, 187, 188
Legend page (Chart Options) 242, 243
legends
  adding to charts 237, 241, 243
  building expressions for 243
  changing chart 237, 241
  hiding 241
  reformatting numbers in 242
  repositioning 241, 242
  setting fonts for 611
less than (<) character tag 742
less than operator 705
less than or equal to operator 705
letters 338
  See also characters; text
level break listings. *See* grouped reports
LI tag 740
libraries 809
  See also DLLs
  accessing 572, 573

libraries *(continued)*
adding classes to 624
adding components to 576, 578
adding scripting controls 648, 654
adding to configuration files 589
changing components in 584, 585, 587
creating component 574–578
creating function 578–579
deleting 585
displaying 574, 590
generating large reports and 575
hiding properties for 565
including in designs 573, 579–584, 588
limiting size of 575
naming 576, 577
overview 572, 573
protecting 591
publishing to 34, 68, 586–587
referencing components in 584, 587
removing classes from 129
restoring 585
restricting changes to 591
saving 576, 577, 578
saving custom properties and 559
selecting 580, 581, 582
selecting components from 47, 584
setting as read-only 591
setting location of 590
setting search paths for 616
specifying default 583
subclassing components in 585–586
upgrading and 746
Libraries page (design perspective) 590
Libraries page (Options) 583
Libraries tab 20
libraries view. *See* libraries viewer
libraries viewer 809
Libraries window 7, 20, 46, 587
See also libraries viewer
Library element 588, 589
library files 834
See also report object library files
library icon 576, 587
Library Organizer 809
accessing 574
adding libraries to configurations and 573
adding libraries to designs and 577, 580, 581, 582
creating component libraries with 577
creating function libraries with 578
displaying configuration paths from 590
displaying included libraries in 590
overview 574
selecting libraries from 580, 581, 582
Library Organizer command 574
Library Structure window 576
Library wizard 627
licensing options 751
line breaks 352, 739, 740
line charts
See also charts
adding drop lines 259
adding high-low lines 259
adding trendlines 263
aligning components in 255, 256
as study chart layer 206
changing series labels for 251
creating 200
displaying data points for 257
generating sample data for 270
labeling data points in 253
plotting data for 257
selecting data for 207, 220, 250
setting line styles for 252
Line command 151
line controls 810
See also lines
adding 117, 150
expanding 378
resizing 383, 384
line feed characters 708
line feed control codes 339
line heights 353, 378
line spacing 354, 355, 738
Line Style dialog 244, 245, 246, 252
line styles
borders 147
charts 237, 246, 252
line or rectangle controls 151
tick marks 244, 245
line widths 147, 352, 378
linear trendlines 260
linefeed characters 739

lines
as graphical elements 150
as section dividers 117, 150
as static element 97
bar charts and 265, 266
connecting data points with 257
line or scatter charts and 257
line or stock charts and 259
pie charts and 254
LineSpacing property 354, 355
LineStyle properties group 151
LineWidthPadding property 352, 353
link expressions
See also LinkExp property; search expressions
adding absolute paths to 456
adding dynamic parameters to 460, 462
adding to controls 462
creating 215, 453, 454
displaying 464
including search expressions in 457–459
specifying destinations in 457, 459
link fields 426
See also sort fields
LinkExp property
character encoding and 466
creating reports with 459
defining parameters with 459–460
JavaScript functions and 464
overview 453
report bursting and 551, 553
setting from Parameter Editor 462
setting programmatically 454, 455, 457, 460
syntax for 454
temporary reports and 456
linking charts to reports 269
linking security roles 544, 545
linking subreports 400, 421–427, 520
linking to parts of a report 690
linking to reports 214, 465
links. *See* joins; hyperlinks
Listing Report option 627
Listing Report wizard
accessing 10
creating detail reports with 425
creating subreports with 411, 418
finishing designs for 18
grouping data with 14
hiding 627
overview 70
selecting data for 13
selecting data sources from 12
selecting layout options 17
sorting data with 17
summarizing data and 16
listing reports 810
See also grouped reports
adding summary data to 16, 297
changing designs for 18, 25
compared to cross tabs 297, 299, 308, 316
creating 10–18
displaying queries for 21
grouping data for 14
laying out 81
previewing designs for 18
running 27
saving designs for 18
selecting data for 13
selecting data sources for 11, 12
selecting layouts for 17
sorting data for 17
viewing structure of 23
lists
adding to HTML forms 344, 648, 740
customizing properties and 558
displaying 338
generating items for 511, 513
ordering values in 513
selecting values in 511, 512
setting indentation levels for 357
literal characters
date formats and 143
number formats and 141
search expressions and 704
literal strings 458, 681
literal text 105, 107, 126, 342
literal values 341, 454
loading sample configurations 5
local file systems 581
local reports 676
Local Subclass command 586
local variables 488, 811
See also global variables; variables

locale map file 142
locale maps 811
localemap.xml 142
locales 811
  building charts for multiple 196
  changing default fonts for 613
  defined 811
  designing for 67, 167
  displaying bi-directional text and 341
  formatting currency values and 26
  formatting dates and time values for 142
  installing fonts for 271, 480
  setting default 619
localization 811
  See also locales
Locate configuration file dialog 5
Locate Default Library dialog 583
locating
  data. *See* searching
  searchable report fields 700
Lock View button 55
logarithmic trendlines 261
logic errors 485
logical values 141
  See also Boolean values
login information 595
login sessions 531
logos 151
Long data type 811
Long Date format type 142
long reports 700
Long Time format type 142
Long Year format type 143
lookahead aggregates 118, 120, 121, 122, 812
low values, plotting 203, 204, 212, 226
lowercase characters 706
low-resolution monitors 473

M

mailing label reports 812
mailing labels 338
main window. *See* application window
major axis interval options 246
major tick marks 244
Management Console
  accessing Encyclopedia volumes 620
  creating iServer profiles for 595
  defined 812
  defining URLs for 619, 620
  file types displayed in 790
  publishing reports and 594, 598
Management Console URL text field 620
management tools 812
map files 602
map files. *See* data source map files; locale maps
mapping security roles 545
mapping XML data 665, 666, 669
Margin properties group 362, 363
margins
  changing report 76, 168, 177, 332
  creating cross tabs and 331
  printing oversized pages and 176, 332
  printing reports and 177, 332
  setting dynamic text 362–364
  setting frame 365
  setting page 172
markup languages 784, 788, 799
  See also elements; tags
master font files 475
master reports
  See also subreports
  adding to designs 548
  building queries for 422
  creating 400, 401
  designing subpages for 407
  linking to detail reports 421, 426, 549, 551, 553
  report bursting and 549
  running 427
master_fonts.rox 475
master-detail relationships 421, 520
matching specific characters 704, 705, 706, 707
matrix reports. *See* cross-tab reports
Max function 114
maximum values 114, 247
MaximumHeight property 348, 351
MaximumWidth property 349
measurement units
  adding suffixes to 615
  customizing 615
  selecting 10, 613

setting 614–615
Medium Date format type 142
Medium Time format type 142
member variables 566, 567, 568, 813
See also field variables; variables
members 812
memo fields 338
memos 338
menus
creating browser-specific 648, 654, 656
displaying 608
selecting items on 36
messages 440, 496
meta languages 662
metacharacters 50
metadata 813
defined 813
method editor 485, 813
See also Methods page
methods 813
See also functions
conditionally formatting data with 155
debugging 484, 492, 495
defined 813
developing with 61
displaying call stack for 487, 489
displaying class-specific 492
filtering 492
generating XML data and 668
overriding 298, 489, 818
removing breakpoints from 487, 494
setting breakpoints on 487, 494
stepping into 487, 495
stepping over 487, 495
subclassing cross tabs and 298
verifying input values with 526, 527
viewing in code 489
XML data formats and 662
Methods page 813
Methods page (Properties) 527
Microsoft Excel spreadsheets. *See* Excel spreadsheets
Microsoft Internet Explorer. *See* Internet Explorer
Microsoft Windows. *See* Windows systems
millimeters 614, 615
MIME types 667
Min function 114
minimizing blank space 334, 352, 354
minimizing windows 20, 41, 42
minimum values 114, 247
MinimumHeight property 347
minor tick marks 245, 247
minus sign (-) icon 45
misaligned content 610
missing groups (charts) 230, 232, 234
missing groups (cross tabs) 318–325
missing values 256
mistyping reserved words 484
modal windows 813
mode 814
modeless windows 814
modifying. *See* changing
modules 814
monitoring report generation 677, 678
monitoring variables 488, 491, 496, 498
monitors 473
monospaced fonts 480
monospaced text 741
More button 690
Move Forward command 140
Move to Front command 140
moving
application windows 41
components 135
controls 348
toolbars 43
moving average trendlines 263
moving through chart data 214
moving through reports 173, 687–690, 708
Mozilla Firefox. *See* web browsers
multicolumn flows 406
multicolumn page layouts 106, 166, 185
multicolumn reports 814
multidimensional data 815
multifield searches 700, 701, 703, 704
multiflow page layouts 408
MultiLine property 646
multi-line text controls 141, 338, 352, 353, 473
multilingual reporting 67
See also locales
multi-object searches 711
multi-page reports
adding dynamic text controls and 338, 358

multi-page reports *(continued)*
adding headers and footers to 27, 68
controlling page breaks for 348, 358, 361, 365
defining multiple page layouts for 187–193
designing for 174
repeating headings for 107, 110
report bursting and 548
setting margins for 177, 363, 365
splitting controls for 383
splitting cross tabs and 332, 333
multi-platform applications 472–473, 479
Multiple Expressions options (charts) 220
multiple page layouts 167, 187
multiple views 693
multiplication operator character tag 743
Multipurpose Internet Mail Extensions. *See* MIME types
multi-style formats 338, 341
multi-table reports 10, 21, 296, 421
multithreaded applications 815
My Reports directory 9

N

name prompts 48
name space. *See* global name space; XML namespace
named values 569, 570
names
adding group headings and 99
case sensitivity for 416
creating search expressions and 457
customizing properties and 559, 561, 563
displaying 608
prompting for 609
publishing and duplicate file 594, 597
restoring group 622
restrictions for 25, 47, 77
sorting on display 520
sorting on group 519
storing temporary components and 127
subclassing components and 586
viewing component 45
naming
component libraries 576, 577
controls 25
custom properties 562
data structures 567
detail reports 549, 551, 552
enum data types 570
flow components 416
function libraries 578
report components 47, 48
report files 20, 77
server profiles 594
naming conventions 25, 47, 77
navigating through chart data 214
navigating through reports 173, 687–690, 708
navigation bar 815
navigation tools 466
nested frames 348, 350, 376
nested reports. *See* subreports
nesting conditional sections 428
nesting reports 400, 401
network administrators. *See* administrators
network file servers 686
New command 69, 71
New Horizontal Tab Group command 693
new keyword 460
New Library dialog 577, 578
New Property dialog 562, 568
New Vertical Tab Group command 694
NoClipping value 646
node 815
NoHeaderOnFirst value 447
non-breaking hyphens 739
non-breaking spaces 739, 742
nonprinting ASCII characters 708
non-TrueType fonts 480
non-visible components 46
non-zero numbers 141
NoSplitBottom property 361
NoSplitTop property 361
not operand 705
NotSearchable value 442
Null keyword 681
null symbol 739, 742
null values 114, 681, 815
numbered lists 338, 357
numbering report pages 181, 534
See also page numbers

numbers
See also numeric controls
analyzing 297
charting comparisons for 202
converting to strings 578
copying 696
customizing measurement units for 615
displaying 97, 145
formatting chart 250, 251, 252
formatting in reports 141, 329
grouping 91, 93, 316
selecting at run time 510
numeric controls 25, 792, 802
See also numbers
numeric data types 16
numeric expressions 815
numeric formats 141
numeric searches 706

O

object inspector 692
Object Inspector command 692
object reference variables 489, 496, 498, 816
See also objects; variables
object-oriented designs 60
object-oriented programming 816
objects 816
See also persistent objects; transient objects
creating Flash 649
defined 816
highlighting 692
referencing 496
searching for 442, 448, 710
selecting 700, 702
tracing variables for 497, 498
obsolete properties 488
ODA (defined) 816
ODA drivers 817
defined 817
ODBC (defined) 817
ODBC Data Source Administrator 73
ODBC data sources 4, 73
ODBC databases 30
See also databases; ODBC data sources
ODBC drivers 4
OL tag 740
old keyword 460
On/Off format types 141
on-demand reporting 505
one-to-many relationships 449, 450
one-to-one relationships 449, 450
online documentation xxi, 6, 7
online help. *See* online documentation
online reports 173, 174, 473
See also reports; web pages
OnRow method 650, 652
Open command 676
open data access 816
open data access technology 816
See also ODA
open database connectivity. *See* ODBC
Open dialog 686
open values, plotting 203, 212, 225
opening
blank report designs 73
call stack 487, 489
Chart Builder 218
color palettes 147
configuration files 5
context menus 608
Crosstab Builder 303
Custom Browser Code Editor 645
data perspective 21
Database Browser 21
design window 5
executable files 676
Expression Builder 102, 220
Hyperlink Builder 270
Library Organizer 574
Listing Report wizard 10
Options dialog 5, 608
Parameter Editor 521
Publish and Preview Options dialog 595
report files 60, 620, 686
Requester dialog 507
sample configurations 5
sample reports 686
Scratch Pad 127
search definition files 712
Search dialog 702
Sorting and Grouping dialog 86
source files 493
table of contents 689

opening *(continued)*
view windows 693
operators 707, 817
optimizing
dynamic text content 352, 354, 357
performance. *See* performance
reports 341, 392
searches 448, 453
optional parameters 504, 507
optional values 504
options (designer) 608, 619
options (licensing) 751
Options command 5
Options dialog
accessing 5, 608
adding default libraries 583
customizing designs with 608
prompting for new names and 48
selecting configuration files from 5
setting class options 623, 624, 625
setting default fonts 610, 611
setting design options 608
setting general options 619, 620, 621
setting global search paths 618
setting measurement unit options 614, 615
setting up designs and 8
or operand 705
Order By page 817
ordered lists (HTML) 740
ordering items in lists 513, 740
orientation (charts) 239
orientation (reports) 18, 76, 168, 170, 715
outer joins 817
See also joins
OuterBorder property 327, 328
outline styles (charts) 266, 269
output
adjusting for specific environments 352
clipping browser 645, 653
converting to PDF 481, 653
designing for 173, 174, 472, 473
displaying query 474
font information and 472
generating for specific browsers 650, 651
generating XML 664, 665, 668
optimizing 392
removing obsolete properties for 488
scaling 652
specifying alignment for 646
testing 128, 129, 353
troubleshooting dynamic text 372–374
troubleshooting report 394–397
Output File Name parameter 504
output file names 504, 507
output formats 818
See also formats
output parameters 506, 510
Output window 485, 487
OVERALL clause 118, 120, 121, 122
overflow characters 141, 144
overflow CSS attribute 645
Overlap settings 265
overlapping areas in charts 199
overlapping controls 473
overlay axis titles 238
overlay charts. *See* dual y-axis charts
Overlay Data page (Chart Builder) 226
Overlay layer legend text setting 243
overlay layers (charts)
adding 204–205, 219
formatting values and 243
specifying data for 213, 226
Overlay page (Axes) 245, 246
Overlay page (Labels) 248, 249
Overlay page (Plot)
selecting up or down bar options 259
setting bar formatting options on 266
setting drop line options 260
setting high-low line options 259
showing markers and lines 258
Overlay page (Point Labels) 253, 255
Overlay page (Scale) 247
Overlay page (Series) 252
overloaded methods 818
override 818
overriding methods 298, 489
overriding output file names 504
oversized reports 331, 332
overwrite keyword 460
overwriting report files 459, 597, 746

P

P tag 741

package (defined) 818
packages 818
padding 352, 356, 357
page 818
See also tabs; windows
page access properties 533
page breaks
adding to subreports 420
displaying dynamic text and 348, 358, 365
generating RTF output and 737
implementing page-level security and 532
inserting after title pages 192
inserting in group sections 122
splitting frames and 358, 361
page components 165, 179, 341, 818
page footers 793, 819
page headers 797, 819
page headers and footers
adding date and time stamps to 182
adding dynamic text controls to 351
adding images to 151
adding report titles to 123
adding to reports 27, 68, 96, 179–180
changing page numbers in 181–182
creating 98
defined 793
inserting visual dividers for 150
printing 173
searching 447
testing 129
page layout components 165
page layouts 807, 819
See also designs; multi-page reports
adding borders to 183
adding headers and footers. *See* page headers and footers
adding watermarks to 184
changing default 123, 167
checking output for 128, 129
checking running headers and footers for 129
choosing blank reports and 123, 164, 167
compared to report layouts 163
controlling dynamic sizing and positioning for 376, 378
controlling page breaks for 358, 361, 365
controlling uniform size of 391
creating cross tabs and 331, 378
creating for specific sections 401, 407
creating for subreports 406–409, 414, 436
customizing 31, 126, 406
defined 807
defining multiflow 408
defining multiple 167, 187–193
designing 80, 165, 167, 173, 473
displaying 123, 163
dynamic text controls and 338, 341, 358, 365
dynamically sized components and 376, 377, 378, 380
font information and 472
grouping controls for 362
overview 81, 162, 165
placing searchable controls in 448
planning 30, 66, 67, 68, 167
prerequisites for 81
prototyping 80
resizing frames for 136
resizing pages for 390–394
selecting 17
setting margins for 172, 177, 363
setting page access properties for 533, 543, 544
setting page attributes for 168, 169, 170
specifying background colors for 183
specifying page numbers for 181
testing 126, 127
XML data formats and 662, 663
page list components 165, 819
page list slot. *See* PageList slot
page number components 123
page number controls 173, 174, 181, 537, 819
page numbers
adding 181–182, 534
customizing formats for 182
displaying 173, 537
generating output and 173, 174
removing blank space for 334
page orientation 18, 76, 168, 170, 715
page ranges 714
page security. *See* Page Level Security Option; SmartSheet Security Option
page settings (default) 167, 181
Page Setup command 76, 168

Page Setup dialog
setting margins with 172, 177
setting oversize page options on 176
setting page size options on 168, 169
specifying page attributes and 168
specifying page orientation from 170
splitting cross tabs and 332
page size
changing 168, 392
displaying output and 173, 390
fixing 168, 391, 392
setting 169, 170
page styles 165
PageBreakAfter property 122, 192
PageBreakBefore property 122
PageFooter property 820
PageFooter slot 96, 351, 820
PageHeader property 821
PageHeader slot 96, 351, 821
page-level security
creating rules for 531, 532
customizing ACLs for 543–546
defining security roles for 545
defining virtual security IDs for 541–543
designing for 532–538
disabling 531
generating ACLs for 531, 544, 545
numbering report pages for 534
overview 530
testing 533, 534, 538–541
XML data formats and 662
page-level security. *See* Page Level Security Option; SmartSheet Security Option
PageList property 821
PageList slot 163, 821
See also page list components
PageNo$ function 102
PageNumberFormat property 182
PageNumberType property 182, 534, 537
pages 818
PageStyle property 822
PageStyle slot 822
pagination 358
See also page breaks
Pagination properties group 358, 363
paging commands 678, 687
paging through reports 173, 687
palette (designer). *See* toolbox
palette files. *See* class representation files
palette representation files. *See* class representation files
palettes. *See* toolbox
panes 822
paper size 18, 76, 714
paragraph breaks 739
paragraph formatting tags 737
paragraph spacing 352, 354, 355, 738
paragraph tag (HTML) 741
parallel section components 411, 822
parallel section icon 402
parallel sections 822
adding components to 404
adding report sections to 411, 413
adding subpages to 408
adding to reports 411
assigning flow components to 409
creating subreports and 408, 410, 411, 436
customizing page layouts for 406
described 402
displaying properties for 409
searching 446, 448, 449, 452
sharing connections and 413
parallel subreports
assigning flows to 409, 415
changing designs for 416
connecting to 413
creating 411, 413
defining multiple flows for 408
examining example for 433
laying out 406, 414
running 416
Parameter Editor
creating parameters and 461, 510
deleting parameters and 522
hiding parameters and 427, 521
opening 521
viewing parameter information in 521
parameter editor 823
Parameter Properties dialog
accessing 510
creating check boxes and 519
creating drop-down lists and 512
creating parameter lists and 515
creating radio buttons and 512

hiding parameters and 521
setting general properties on 510
setting parameter properties and 510, 512
parameter value pairs 460
parameter values files. *See* report object value files
parameters 823
adding to designs 507
adding to link expressions 454, 462–464
adding to Requester 509–511, 519
assigning values to 678–680, 681
changing 504
choosing custom values for 680
conditional expressions and 405
debugging reports and 484, 491
defining ad hoc 430, 507
defining dynamic 459–460
defining link fields and 426
defining required 504, 509
deleting 521–523
displaying 519, 521
filtering data and 522
getting information about 521
grouping 509, 510, 519
hiding 427, 520
loading Requester 525
overview 504
prompting for values with 506, 509, 518, 526
providing list of choices with 511–518
publishing and 598
resetting defaults for 525
saving 491, 524–525
setting display names for 510
sorting 520
specifying default values for 509, 510, 526
testing 523–524
verifying 527
Parameters command 427, 521
Parameters page 824
parent class. *See* superclasses
parsing 646
Password field (iServer Profile) 595
passwords 824
accessing sample databases and 12
adding to server profiles 595
defined 824
defining parameters for 504
entering at run time 510
Paste command 58
paths
creating hyperlinks and 456
displaying database 21
displaying search 590
saving reports and 8
setting search 590, 616
pattern matching 705, 706
pattern matching. *See* regular expressions; search expressions
patterns 824
See also object-oriented programming
patterns (backgrounds) 183
patterns (formatting) 142, 143, 182
pausing code execution 487
pausing report execution 494
PCX files 151
PDF converter 480, 653, 825
PDF files
converting XML data for 663
creating multi-platform applications and 480
displaying contents 653
generating 480
getting font information for 474
printing from 653
problems reading 610
saving cross tabs as 331
setting AlternateText property for 646
PDF formats 640
PDF reports. *See* PDF files
PDF Viewer 352, 473
See also Acrobat Reader
Pen property 151
percent bar charts 265
Percent format type 141
percentages 201, 254
performance
dynamic text controls and 341
file size and 548
large libraries and 575
subreports and 421
XML documents and 667
period (.) character in class names 25
permissions. *See* privileges

Persistent Object Storage Mechanism. *See* POSM
persistent objects 825
personal folder. *See* home folder
perspective toolbar 37
perspectives
  See also specific application perspective
  arranging windows in 43
  changing defaults for 619
  managing workflow on 37
  restoring defaults for 41, 44
  saving 693
  switching among 37
pie charts
  See also charts
  adding chart layers and 204
  changing 239
  creating 201
  defining sectors for 223
  displaying point label lines for 254
  exploding sectors in 268
  generating sample data for 270
  grouping sectors in 223
  labeling data points in 252, 253
  renaming sectors in 223, 267
  selecting data for 208, 222
  setting sector attributes for 267–269
  sorting sectors for 223
plain text formats 339, 342, 355
platform 825
platform-independent formats 662
platform-independent hyperlinks 464
plot (charts) 197, 237
Plot as zero value setting 256
Plot background setting 240
Plot border setting 240
Plot page (Chart Options)
  adding and changing drop lines 260
  selecting bar formatting options from 266
  selecting data point marker and line display options 257
  selecting up or down bar options from 258
  setting high-low line options 259
  setting pie sector attributes on 268, 269
plotting empty values 250, 256–257
plotting missing values 256
plus sign (+) icon 45
point labels 197, 250, 252
Point Labels page (Chart Options) 252, 253, 254
points (charts). *See* data points
points (measurement unit) 614, 615
polynomial trendlines 261
pop-up menus. *See* context menus
Port settings (iServer Profile) 595
Portrait orientation 170
ports 595, 604
POSM (defined) 825
pound sign (#) character 459
power trendlines 262
PowerPoint converter 825
PRE tag 741
predefined classes 61
predefined connections 6
predefined data sources 6, 12
predefined databases 12
predefined values 770
previewer 825
previewing reports 31, 39, 59, 716
Print command 714
Print dialog 714
Print Preview command 716
printers 472, 714, 715
printing
  across platforms 472, 473
  cross tabs 331–336
  dates 334
  dynamic text controls and 352
  oversized reports 176, 332
  page headers and footers 173
  page size options and 168, 169, 173, 174
  page-level security and 530
  PDF files 653
  Reportlets 640
  reports 714
printing options 714
printouts 173, 174, 472
private access type 749
privileges 826
  copying 594, 598
  page-level security and 530
  report bursting and 548
  security IDs and 531
  XML reports and 662

procedures 826
procedures. *See* methods
ProcessText method 341
profile information 594, 596
Profile name field (iServer Profile) 594
profiles
  creating server 594–596, 602
  selecting 598
  setting publishing options for 597
  testing 596
programmers 61, 62, 700
programming basics 60
programming languages 61, 749, 804, 856
programs
  See also applications
  clearing from memory 496
  controlling execution of 494
  debugging 484
progressive viewing 826
Project browser 827
project files 602
projected values 203
prologs (XML) 663, 666, 667, 669
Prompt for new name setting 48
Prompt for URL setting 620
prompting for component names 609
prompting for input 506, 518, 526
properties 828
  assigning to groups 561, 562
  assigning values to 563
  changing chart 237, 238
  changing component 56, 586
  changing default formats and 76, 133, 140
  converting browser scripting 646, 653
  copying components and 58
  copying file 597
  creating 558, 561, 562
  cross-tab controls and 298, 327, 328, 331
  customizing 558–565
  defined 828
  defining as value expressions 558, 564–565
  defining structure types and 566, 568
  displaying 52, 56, 57
  dynamic resizing and 376, 377
  entering values for 614
  filtering 56, 565
  hiding 562, 565
  identifying obsolete 488
  removing parameters for 522
  saving 559
  selecting values for 56
  setting at run time 506
  setting component 56, 60
  setting font 478
  setting Reportlet 636
  setting table of content 466
  setting XML 665, 666–667
Properties command 56
Properties page 827
  See also properties
  accessing data and 533
  charting data and 238
  creating parameters and 521
  customizing properties and 558, 563, 564, 565
  displaying cross tabs and 311, 330
  displaying Reportlets and 636
  filtering properties on 56
  searching for components and 442, 444
  selecting read-only components and 591
Properties window 827
property attributes 562, 563
property groups 558, 561, 562
property icons 559
property lists. *See* Properties page
property sheets. *See* Properties page
protocol 828
prototypes 30, 66, 68, 163
publish 829
Publish and Preview Options dialog
  accessing 595
  publishing from 598
  setting publication options in 596, 597
  setting version options in 597
Publish and preview setting 597
Publish and show folder contents setting 597
Publish only setting 597
Publish the component setting 587
Publish to Server command 595, 598
publishing
  components 34, 586–587
  duplicate file names and 597
  files 594, 597, 598
  reports 594, 598–599

publishing destinations 595
publishing options 597
pull-down menus. *See* menus
pushpin icon 20, 41

Q

QBE (defined) 829
QBE syntax 829
qualified names 457
quarterly reports 575
quarterly sales reports 207
queries
  See also Query Editor; SQL statements
  building cross tabs and 302, 303, 315
  building for master reports 422
  building grouped reports with 84
  building listing reports with 13
  creating 22, 74–75
  defined 829
  displaying output for 474
  displaying statements for 21
  executing multiple 548
  narrowing scope of 507
  prompting for user input and 511, 513
  retrieving data with 31, 46
  synchronizing 830
  testing 31
Query by Example. *See* QBE
query data streams 829
Query Editor
  adding columns to 86
  adding tables to 74, 84
  deleting parameters with 521, 522
query editor 829
query languages 753
query parameters. *See* parameters
question mark (?) character
  class names and 25
  parameter declarations and 460
question mark (?) character tag 743
Quick Report option 627
Quick Report wizard 69, 627
Quikdata sample database
  changing designs for 25
  connecting to 12
  creating listing reports with 12
  grouping data from 14
  selecting tables from 13
  sorting data from 17
quotation mark character tag 742
quotation mark characters. *See* double quote character; single quote character

R

–r option (erdpro) 747
radio buttons 511, 512, 513, 524, 680
range 830
range of values 203, 321
  defined 830
range operator 705
raster fonts 480
Read Palette Class Representation dialog 626
read privilege 530, 830
read-only libraries 591
recently opened files list 620
records 69, 74, 75, 230
  See also rows
Rectangle command 148, 151
rectangle controls 831
  adding shadows with 148
  as component border 147
  as graphical elements 150
  controlling dynamic sizing for 378, 383, 384
  grouping data and 150
  setting fill colors for 151
recurring tasks 624
recursion 490, 831
references 831
referencing
  components 47, 584, 587
  custom browser code 648
  custom properties 560
  image files 152
  objects 496
Refresh button 6
refresh delays 678
refreshes 678
regions. *See* frames
registered trademark character tag 743
registered trademark symbol 743
registering font files 479

registry keys 613, 627
caution for 613
regression analysis 260
relational databases. *See* databases
relational expressions 454, 458, 459
relative font sizes 742
relative page numbers 534
relative paths 456, 590, 616, 630
relative URLs 640
removing. *See* deleting
Rename command 25
Rename dialog 25
renaming
components 25, 26, 576
controls 25
data categories in charts 220
data series in charts 220
flow components 415
pie sectors 223, 267
rendering environments 67, 151, 352, 473
replacing default configurations 72
report bursting 548–549, 831
report bursting examples 549–553
Report command 411
report components 526
See also components
report design environment 6, 34, 36, 608
report design files. *See* report object design files
report designs 770, 780
See also designing reports; page layouts
accessing configurations for 5, 588
adding parameters to 507
adding report components to 6, 44, 46, 584
adding section components to 411, 418, 428, 466
applying fonts to 474, 478, 480
arranging components in 134–140, 613
backward compatibility for 60
building multiple reports and 400, 401, 402
building templates for 71, 73, 628
changing 18, 21, 25, 28, 69
closing 59
creating 66, 67, 72, 73, 100
defined 780
developing 60, 692
displaying 44
embedding browser code in 642, 645, 648
embedding tags in 341
enabling page-level security in 532, 535–538
generating information objects for 602
grouping components in 350
hiding empty slots in 95
including libraries in 573, 575, 579–584, 588
missing components in 585
organizing data in 45, 75
overview 30, 32, 34
placing data in 66, 75, 95, 97, 123
preserving previous versions 746
previewing 18, 20
prototyping 68, 163
referencing components in 584, 586, 587
renaming components in 25, 26
report bursting and 548, 549
resizing components in 136
reusing components in 34
saving 9, 20, 58, 127
selecting components in 135
selecting data sources for 12
selecting database columns for 13
setting default directories for 609
setting search paths for 616
stacking controls in 480
testing 126, 127, 128, 533, 534
upgrading 747
verifying structure of 23, 129
viewing libraries in 574, 590
report documents 832
report documents. *See* reports
Report Encyclopedia. *See* Encyclopedia volumes
report executable files 832
See also report object executable files
report executables. *See* report object executable files
report file types 790
report files
See also specific type
building reports and 58, 59
bundling 506
closing 59

report files *(continued)*
compiling 27
copying properties 597
defining hyperlinks for 456, 457, 461, 462
downloading 548
duplicate names and 597
embedding font information in 610
limiting size of 548
naming 20, 77
opening 60, 620, 686
overwriting 459, 597, 746
publishing 594, 597, 598
saving 609
setting default directories for 609
setting recently opened list for 620
setting search paths for 616–618
upgrading multiple 746
uploading 597
report footers. *See* After slot
report-generation date stamps 182
report generation requests 504
report headers. *See* Before slot
report object design files 833
See also report designs
adding custom fonts to 476, 478
adding libraries to 579
closing 59
creating source files from 621
overwriting 746
publishing components from 587
saving 20, 609
setting default directory for 609
setting search paths for 617
upgrading 747
report object designs 833
report object executable files 833
See also executable files
bundling with reports 506
compiling 27
creating 59
defining custom fonts in 477
defining hyperlinks for 456
embedding fonts in 474, 476, 480, 610
generating reports from 459, 548, 676, 711
opening 676
publishing 594, 598
running 676, 677
specifying encoding for 621
upgrading 747
report object instance files 833
bursting examples for 550, 552
bursting techniques for 548, 549
creating 59, 459, 460
defining hyperlinks for 456
extracting content from 635
extracting data from 711, 712
generating 27, 548, 676
opening 686
upgrading 747
report object instances 833
report object library 833
See also component libraries
report object library files 834
See also component libraries; libraries
accessing 582
overview 572
overwriting 746
publishing components to 34, 587
saving component libraries as 577
setting as read-only 591
report object parameter files 834
report object search definition files 711
report object value files 834
browsing for 525
creating 524
defining hyperlinks for 456
generating reports from 459
loading specific 525
running debugging sessions and 491
report objects 832
See also reports
report objects. *See* reports
report output. *See* output formats
report painter. *See* Design Editor; Design Editor page
report parameters 834
defined 834
report parameters. *See* parameters
report section components 411
report section icon 402
report sections 834
See also sections
adding components to 404–405
adding to reports 411

aggregating data and 120, 406
creating cross tabs and 299
described 402
disabling security restrictions for 531
displaying 608
expanding 23
extracting as Reportlets 639
generating ACLs for 531, 544, 545
generating table of contents from 468
inheriting security IDs 543
nesting 532, 543
placing data in 96
searching 447, 449, 453
setting page breaks in 420
setting security restrictions for 541, 543
visual dividers for 150
report server. *See* iServer
report servers. *See* iServer; servers
Report Settings page (Design Properties)
displaying reports and 538
preserving default parameters and 526
selecting search paths and 617
specifying report object value files and 525
testing designs and 533
testing page-level security and 539
report specifications. *See* report designs
Report Structure command 23
Report Structure window 835
double question mark (??) characters in 585
hiding empty slots in 7
opening 23
overview 44
publishing components and 587
setting font for 613
viewing components of 6, 47, 450, 584, 586
report templates 71, 628–629, 851
Report Title component 76, 123
Report Totals dialog 16
report wizards. *See* specific wizard; wizards
reporting system. *See* iServer System
reporting tasks 46
Reportlets 835
creating 636, 637, 638
displaying in web pages 634, 635, 636, 640
enabling hyperlinks for 640
entering HTML tags for 635, 638
importing 636
saving 640
setting visibility properties for 636
reports 787, 831
See also persistent reports; temporary reports
accessing 594
adding interactive features for 439, 649
adding table of contents to 466–469
adding titles to 18, 76, 123
building 46, 58, 400, 686
changing page orientation for 171
collapsing or expanding parts of 45
controlling access to 530, 543
controlling size of 168, 376, 390
converting to XML 662, 663, 666
creating
available options for 69
from templates 71
guidelines for 66
international considerations for 67
platform-independent 662
with wizards 4, 69–70
without wizards 73–77
customizing 60, 61
defined 831
defining optional parameters for 504, 507
defining required parameters for 504, 507
designing. *See* designing reports; designs
developing 61, 72, 376, 504, 692
displaying 59, 67, 472, 686–694
distributing 152, 594
embedding images in 152, 154
embedding in web pages 634, 636, 637
extracting portions of 636, 637, 638
fixing errors in 484
formatting 31, 68, 132, 154
generating 58, 341, 352, 676
laying out 31, 38, 68, 80–81, 163
linking charts to 214
linking to 454, 457, 465, 690
managing large 505
misaligned content in 610
monitoring generation status of 677, 678
moving through 173, 687–690, 708
moving to specific pages in 28, 688
moving to specific parts of 688, 691, 708

reports *(continued)*
naming 77
nesting 400, 401
opening multiple windows for 693–694
optimizing 341, 392
overview 504
previewing 31, 39, 59, 716
printing cross-platform 472, 473
printing from PDF files 653
printing oversized 176, 332
prototyping 30, 66, 68, 163
publishing 594, 598–599
refreshing 678
replacing default configurations for 72
resizing pages for 390–394
running 27, 59, 369, 676, 717
saving structure information for 711
searching 48, 440–453, 700, 701
selecting components for 572, 575
selecting data for 74
selecting data sources for 11
selecting layouts for 17
structuring 30, 66–69, 94, 401
caution for 83
testing 31, 67, 173, 352, 526
troubleshooting output for 394–397
updating data for 676
upgrading 746–747
viewing specific pages 173, 530, 688
viewing specific parts 155
viewing structure of 6, 23, 87, 701
zooming in on 695
repositories 835
requester 836
Requester dialog
adding check boxes to 518
creating drop-down lists for 511, 512
creating parameters for 509, 509–511, 519
creating radio buttons for 511, 512
disabling 526
displaying ad hoc parameters in 508
displaying optional parameters in 504
displaying required parameters in 504
entering ad hoc parameters and 507
hiding parameters in 520
loading parameters for 525
opening 507
ordering parameters in 519
overriding parameter suppression 526
prompting for input and 506, 509, 526
resetting parameter defaults for 525
running reports from 677
saving parameters for 524–525
setting parameter values in 678
suppressing prompting from 526
testing parameters for 523–524
requests 504, 836
required parameters 504, 507, 509
required values 504
reserved words 215
reserved words. *See* keywords
Reset All Groups command 622
Reset Items in this Group command 622
Reset Workspace command 44
resetting active perspective 619
resizing
browser scripting controls 652
chart controls 237
columns 326
components 136, 613
controls 26, 383–387
cross tabs 311
dynamic text controls 347, 348
frames 136, 146, 348, 350
report pages 390–394
rows 326
resolution 473
resources 72, 588, 836
Restore All Panes command 43
restoring component libraries 585
restoring defaults 41, 44
restricting access to reports 530, 543
restricting data sorting 230
result sets 836
defined 836
Results page (Search) 704, 708, 709
resuming code execution 487
resuming code generation 495
return values 128
reusing components 34
RGB color values 147, 736, 741
rich application interface 649
rich information application 836
rich-text formats. *See* RTF formats

right outer joins. *See* outer joins
Right property 362
.rod files. *See* report object design files
.roi files. *See* report object instance files
RoiIsTemporary method 548
.rol files. *See* report object library files
roles 544, 545, 840
  See also privileges
.rop files. *See* report object parameter files
.ros files. *See* report object search definition files; search definition files
round settings (charts) 232, 234
round settings (cross tabs) 321, 324
rounding units 234
.rov files. *See* report object value files
row headings
  adding to cross tabs 296, 306
  displaying computed values in 312
  displaying static text in 312–314
  setting border properties for 327
row keys 837
  See also cross-tab reports
RowHeadingsBorder property 327
rows
  See also data rows; records
  adding background colors to 330
  adjusting spacing between 146
  conditional expressions and 405
  counting 114
  cross tabs and 309, 314, 326, 331
  defined 837
  monitoring values in 496
  resizing 326
  setting border properties for 327, 329
  sorting data in 17
.rox files. *See* report object executable files
RTF files 736
RTF formats 340, 342, 737, 739
RTF readers 737
RTF tags
  adding dynamic text controls and 342, 344, 736–739
  adjusting line spacing with 355
  adjusting paragraph spacing with 355
  assigning literal values with 341
  concatenating 736
  displaying as literal text 342
rules
  applying to nested sections 532
  copying autoarchiving 594, 598
  creating virtual security IDs and 541
  defining security 531
  testing 535, 538
run 837
run requests 504
run time 837
running aggregates 837
running executable files 676, 677
running headers and footers 123, 129
  See also page headers and footers
running queries 548
running reports 27, 59, 369, 676, 717
running search definition files 712
running subreports 416, 419, 433
run-time errors 484, 485
run-time values 496, 499

S

sales, tracking 205, 207, 208, 210
sample configuration files 5
sample data 100, 270–271, 608, 645
Sample Data page (Chart Options) 270
sample databases
  building charts with 220, 222, 223, 225
  building subreports from 410, 417
  changing designs for 25
  configuring 4
  connecting to 12
  creating grouped reports with 83, 84
  creating listing reports with 12
  grouping data from 14
  report bursting techniques for 549
  running reports for 27
  selecting tables from 13
  sorting data from 17
sample reports
  examining subreports in 433
  opening 686
  report bursting techniques in 549
  running 27
  saving 8, 20
  setting up 6, 8
  viewing table of contents for 689

Sample Template option 628
sample templates 628
sample_configuration_file.xml 6
SampleValue property 645
Save As dialog box 20
Save Class Representation dialog 625
Save command 58
saving
  component libraries 576, 577
  custom properties 559
  font files 479
  function libraries 578
  parameters 491, 524–525
  report designs 9, 20, 58, 127
  report files 609
  Reportlets 640
  sample reports 8, 20
  search conditions 711
  toolbox configurations 624, 625
  window layouts 41, 693
scalable fonts 480
Scale page (Chart Options) 246, 247
scaling chart axes values 246–247
scaling factor 651
scaling range (zoom) 695
scatter charts
  See also charts
  as base chart layer 205
  as study chart layer 206
  changing series labels for 251
  creating 202
  displaying data points for 257
  generating sample data for 270
  labeling data points in 253
  plotting data for 257
  selecting data for 210, 223
  setting line styles for 252
schemas 837
  See also ROM schemas
  defined 837
Scientific format type 141
scope 488, 527, 838
scope resolution operator 838
Scratch Pad 7, 127, 838
Scratch Pad command 127
SCRIPT tag 740
scripting controls. *See* browser scripting controls
scripts 740
scroll bars 645
Scroll value (CSS attribute) 646
Scrollbar value 646
search 838
Search button 702
search conditions 839
  adding multiple controls to 448
  adding multiple fields to 701, 703
  changing 704
  creating link expressions and 458
  entering 48, 702–704
  saving 711
search criteria. *See* search conditions
search definition files 711–712, 839
Search dialog
  accessing 702
  adding multiple fields to 703
  adding search conditions to 702, 703, 710
  adding searchable objects to 700, 702
  changing search conditions in 704
  displaying search results from 708, 709
  entering wildcard characters in 705
  running search definition files from 712
  saving search conditions for 711
search expressions 839
  See also link expressions
  adding nonprinting ASCII codes to 708
  adding pattern operators to 706
  adding wildcard characters to 703, 705
  creating 50, 704–708
  entering conditions for. *See* search conditions
  linking to reports and 457–459
  locating special characters with 706, 707
search indexing 839
search paths
  adding directories 617
  changing search order 618
  creating hyperlinks and 456
  displaying 590
  getting font information and 474
  linking to external images and 152
  removing directories 618
  setting 590, 616–618

search results
displaying 52, 704, 708, 709
generating 448
navigating reports and 708
returning hyperlinks 447
selecting 52
Search Results window 52
search URLs 638–640
Search window 442, 443
Searchable property 442, 453
searchable report fields 700, 701
SearchAlias property 442, 443, 444
searching
cross tabs 298
directories 616
for literal strings 458
for multiple components 444
for nonprinting characters 708
for single text values 700
for special characters 706, 707
for specific components 53
for specific data 701–704, 710
from search definition files 711–712
maximizing report size for 548
on literal values 704
on multiple fields 700, 701, 703, 704
page headers and footers 447
page-level security and 530
relationships preventing 449
relationships supporting 450
reports 48, 440–453, 700, 701
web pages 448
with conditions 702
with web browsers 663
SearchNoIndex value 442, 444
SearchPath element 590
SearchTag property
default search behavior and 443, 448, 453
described 442, 443–446
link expressions and 459
multiple class instances and 445
parallel or sequential sections 446
setting programmatically 448
specific classes and 444
SearchValueExp property 442, 636, 637
SearchWithIndex value 442, 444
section components 401, 402, 466, 839
sections 839
See also specific type
adding components to 404–405
adding headers and footers to 180
adding subpages to 407, 408
adding to reports 401, 402
aggregating data and 120, 406
creating cross tabs and 299
creating subreports and 408, 410, 411, 417
defining page layouts for 187, 406
disabling security restrictions for 531
displaying 608
expanding 23
extracting as Reportlets 638, 639
generating ACLs for 531, 544, 545
mapping to XML elements 666
nesting 428, 532, 543
placing data in 96
searching 449, 453
setting conditions for 405
setting security restrictions for 541, 543
types described 402
visual dividers for 150
sector keys 223
sectors (pie charts) 201, 208, 223, 254, 267
secure read privilege 530, 840
security
customizing ACLs for 543–546
defining security roles for 545
defining virtual security IDs for 541–543
designing for 532–538
disabling restrictions for 531
generating ACLs for 531, 544, 545
overview 530
testing page-level 533, 534, 538–541
XML data formats and 662
security IDs 840
access control lists and 530
assigning to users 539
creating 531, 533
defining as virtual 541–543
displaying 535, 536
inheriting 543, 544
nested sections and 532, 543
overview 531
testing 533, 534

security options. *See* e.Reports Page Level Security option; SmartSheet Security option
security roles 544, 545, 840
  See also privileges
security rules
  creating virtual security IDs and 541
  defining 531
  testing 535, 538
select 840
Select Fields page (Report Wizard) 13
Select File dialog 676
Select page (Search) 710
select painter. *See* Query Editor; query editor
SELECT statements 22, 843
  See also SQL statements
Selectable property 442
selecting multiple components 134
selection criteria. *See* search conditions
selection formulas. *See* parameters
semantic errors 128
semantic types 840
semicolon (;) character
  parameter declarations and 460
  search expressions and 458
Send Backward command 140
Send To Back command 184
Send to Back command 140
sequential page numbers 534
sequential section components 418
sequential section icon 402
sequential sections 841
  adding components to 404, 418
  adding to reports 402, 418
  aggregating data and 120, 406
  creating subreports and 417
  described 402
  searching 446, 449, 452
sequential subreports
  creating 417, 418
  example for 403, 433
  running 419
  setting page breaks for 420
series 841
  adding to charts 221, 224, 227
  building chart titles from 238
  changing appearance of 250
  changing color of 250
  connecting data points for 257
  creating chart layers and 227, 229
  defined 841
  described 196
  empty values in 237, 256
  generating legends with 241
  generating sample data for 270, 271
  grouping 222
  labeling 220, 250, 251
  linking to 215, 269
  multiple value expressions as 220, 223, 250
  plotting values for 199, 207, 210, 237
  positioning data points for 255
  renaming 220
  selecting line styles for 252
  selecting multiple columns for 210
  sorting 221
series key expressions 221, 224, 229
series keys 221, 224, 227
Series keyword 215
series lines 257, 266
Series page (Chart Options) 250, 251
SeriesLabelExp property 215
Server Configuration Templates 722
server profiles
  creating 594–596, 602
  naming 594
  selecting 598
  setting publishing options for 597
  testing 596
servers 464, 572, 582, 594, 750, 857
  See also iServer
SetSearchTag method 444
SetSecurity method 531, 544
SetTaggedText method 340
settings. *See* properties
Sfdata sample database
  building charts with 220, 222, 223
  building cross tabs from 303
  creating grouped reports with 83, 84
  creating subreports and 410, 417
  report bursting techniques for 549
SGML formats 662
shadows 148
shared access type 749
shared libraries 572, 841
sharing connections 413

Short Date format type 142
Short Half format type 143
Short Month format type 143
Short Quarter format type 143
short reports 169
Short Time format type 142
Short Week format type 142
shortcut key combinations 36, 37
shortcut menus. *See* context menus
Show bars as lines setting 266
Show Empty Slots command 7, 95
Show group sections setting 608
Show point label lines setting 254
Show Siblings button 55
ShowFooterOnLast property 447
ShowHeaderOnFirst property 447
ShowInDHTML property 480
ShowInPDF property 480
ShowInReportlet property 636, 637
ShowWhenPrinting property 481, 653
ShowWhenViewing property 481, 653
shrinking dynamic text controls 347, 348, 349
side-by-side subreports 403, 406, 410
simple page lists 165
simple variables 489
simulated reports 534, 538
Single data type 841
single quotation mark (') character 739
single quote character tags 739
single table reports 70
single-page reports 169
single-precision floating point numbers 841
singleton pattern 824
SIZE attribute 344
Size command 136
SIZE parameter 742
sizing handles 26
slice key expressions 208, 223
slice size expressions 208
slices. *See* sectors (pie charts)
slot names 9, 608
slots 842
  See also specific
  adding components to 44
  adding cross-tab controls to 299
  adding dynamic text controls 351
  adding sections to 402, 428
  described 95
  grouping data and 98
  hiding 7, 95
  placing data in 95, 97, 114
  placing report titles in 123
  repeating group headings and 107, 110
  searching controls in 449
.sma files. *See* data source map files
small hand cursor 691
SmartSearch. *See* search
SmartSplitHorizontally property 333
SmartSplitVertically property 333
snap to grid setting 135
soft hyphens 739
sort 842
sort fields 17, 125
sort filters 842
sort keys 88, 93–94, 842
sort order
  changing 125
  grouping data and 15
  setting 75, 125, 229
sort-and-group-by fields. *See* group keys
Sorting and Grouping command 75, 86
Sorting and Grouping dialog 75
Sorting and Grouping page (Chart Builder) 236
Sorting and Grouping page (Crosstab Builder) 313, 315, 318
Sorting and Grouping tool 91
sorting data
  charts and 229–230
  cross tabs and 315
  grouping compared with 83
  reports and 75, 124–126
sorting options 229, 316
Sorting page (Chart Builder) 229
Sorting page (Crosstab Builder) 315
Sorting page (Designer) 75, 125
sorting parameters 520
SortParamsByAlias property 520
source code
  See also Actuate Basic
  changing fonts for 610, 611
  controlling execution of 494–496
  creating Flash objects and 649
  customizing reports and 60, 61

source code *(continued)*
debugging 39, 484, 493, 495
displaying 493
editing 621
generating 58, 486, 494
grouping data and 93
removing breakpoints from 487, 494
setting breakpoints for 494
setting encoding options for 621
stepping through 487, 489, 495, 498
testing and debugging 780
verifying input values with 526, 527
viewing browser scripting 647, 653
viewing errors in 487
viewing methods in 489
writing for web browsers 644, 650
source editor 610, 621
source files 758
See also source code
adding to Library Organizer 582
creating 621
debugging 484, 493
defining functions in 572, 578, 579
encoding in external 621
opening 493
referencing browser code in 648
setting search paths for 616
tracing through 495
updating 59
Space Evenly command 137
SpaceBetweenLines property 355
SpaceBetweenParagraphs property 355
spacing (dynamic text) 352, 354, 355, 738
span settings (charts) 232, 235
span settings (cross tabs) 321, 324
special characters
DHTML converter and 643, 644
generating RTF output and 739
including in search expressions 458, 459
including in URLs 466
naming reports and 77
searching for 706, 707
Specify Grouping page (Report Wizard) 14, 15, 16
Specify Parameter File Name dialog 525
split box 694
Split Oversize Pages setting 176, 331
SplitIfNecessary value 358, 359, 360
SplitIfPossible value 358, 360
SplitMarginBottom property 177, 332, 363, 365
SplitMarginLeft property 177, 332
SplitMarginRight property 177, 332
SplitMarginTop property 177, 332, 363, 365
splitting
cross tabs 332, 333
frames 358–365
oversized reports 176, 332
windows 694
spreadsheets 474, 475
SQL (defined) 842
SQL language 842
SQL languages 753
SQL page 843
SQL page (Query Editor) 22
SQL statements 843
See also queries
adding columns to 85
adding tables to 84
building 22
defined 843
displaying 21
limiting data returned by 507
Srchdef directory 711, 712
stack 487, 490
stacked bar charts 265
stacked controls 45
stacking application windows 41, 42
stacking controls 480
stand-alone file systems 686
Standard format type 141
Start method 526
starting
debugging sessions 492–494
e.Report Designer Professional 5
starting values, plotting 203, 204, 212
state
See also instance variable
defined 843
statement objects 844
statements 484, 495, 844
static controls 97, 844
static groups (charts) 230, 235–236
static groups (cross tabs) 319

Static Groups page 236, 313
static headings 312
static hyperlinks 844
static parameters 844
See also global variables
static text 341, 807
static variables 844
See also dynamic variables; variables
status bar 36
Status Bar command 36
status messages 27
step charts
See also charts
as study chart layer 206
changing series labels for 250
creating 203
generating sample data for 270
labeling data points for 253
plotting empty values for 256
selecting data for 207, 220
Step Into command 487, 495
Step Over command 487, 495
stepping through code 487, 489, 495, 498
stock charts
See also charts
adding bars 258
adding drop lines 259
adding high-low lines 259
adding legends to 241
adding trendlines 263
changing series colors in 250
creating 203
generating sample data for 270
labeling data points in 253
plotting missing values for 256
selecting data for 212, 225
stocks, tracking value of 200, 203, 212
stopping code generation 487, 496
stopping report generation 677
stored procedure browser 845
Stored Procedure Data Source Builder 845
stored procedures 845
string calculations 855
String data type 845
string expressions 845
string variables 652
String$ function 111
strings 845
assigning literal values to 341
browsing scripting controls and 643
converting expressions to 531
converting integers to 578
defining empty 654
defining for online help 440
displaying 97
grouping characters in 90, 93
link expressions and 454
locating special characters in 707
measurement units and 616
monitoring values of 499
not displaying correctly 480
PDF conversions and 653
returning null values with 681
searching 458, 705
structural components 846
structural relationships 846
structure aliases 566
structure icon 567
structure members 567
structure pane. *See* Report Structure window
structured content 33, 846
Structured Query Language. *See* SQL
structures
See also source code
creating 566–568
customizing properties and 561, 564
debugging 489, 491
structuring reports 30, 66–69, 94, 401
caution for 83
study axis titles 238
study charts 846
Study Data page (Chart Builder) 228
Study layer legend text setting 243
study layers (charts) 205, 213, 219, 228, 243
Study page (Axes) 245, 246
Study page (Labels) 248, 249
Study page (Plot)
selecting up or down bar options 259
setting bar formatting options on 266
setting drop line options 260
setting high-low line options 259
showing markers and lines 258
Study page (Point Labels) 253, 255
Study page (Scale) 247

Study page (Series) 252
STYLE element 647
style sheets 645, 647
styles 846
  See also formats
  defined 846
subclasses 55, 60, 298, 559, 585, 667, 847
  See also descendant classes
subclassing components 60, 585
subcolumns (cross tabs) 308, 309
subpage components 407, 408, 848
Subpage property 848
Subpage slot 407, 848
subpages 341, 407, 408
subreport bursting 549
subreport bursting example 550–551
subreports 848
  adding to parallel sections 411, 413, 436
  adding to sequential sections 418
  assigning flows to 409, 415, 437
  changing designs for 416
  creating 400, 410, 417, 422, 427
  defined 848
  defining page layouts for 187
  defining shared connections for 413
  designing 400, 401, 403, 433
  displaying properties for 415
  examples for 403
  hiding structures 95
  laying out 406, 408, 414, 436
  linking 400, 421–427, 520
  optimizing performance for 421
  running 416, 419, 433
  setting page breaks for 420
  specifying conditions for 405–406
  structuring reports for 400–401
subroutines 849
subrows (cross tabs) 308, 309
substituting fonts 474, 480
subtotals 16
SuggestRoiName method 549, 551, 552
Sum function 114
sum function 16
summary data
  See also aggregate data
  adding 16, 76, 97
  comparing 118, 120
  creating cross tabs and 298, 310, 316
  creating listing reports and 297
  designing for 68
  filtering 118
  grouping data and 114
  inserting graphical elements for 150
  placing in charts 220, 222, 227, 229
Summary Options button 16, 220
summary subreports 436
summary values 296
  See also summary data
SummaryControl value 564
Super keyword 849
superclasses 55, 586, 849
  See also ancestor classes
suspending code execution 487
suspending report execution 494
symbol (programs) 849
synchronization 830
syntax (programming languages) 849
syntax errors 128, 484

T

tab characters 339, 708, 739
tab stops 356, 357, 738, 739
tabbed pages 41
tabbed-text formats 356
table of contents 850
  accessing 689
  adding to reports 466–469
  expanding or collapsing items in 689
  moving through reports from 688
  searching reports and 448, 701
Table of Contents button 701
Table Parameter Editor 681
tables 850
  See also databases
  adding to Query Editor 84
  creating joins for 21
  creating reports and 10, 69
  defined 850
  displaying database 21
  displaying fields in 21
  grouping data from 14
  selecting 13, 74
TabPadding property 357

tabs 850
See also page
TabSpacing property 357
tags 850
See also elements
adjusting line spacing with 355
adjusting paragraph spacing with 355
appearing as literal text 342
assigning literal values with 341
embedding 341
generating HTML 644, 740
generating Reportlets and 635, 638
generating RTF output and 344, 736, 737, 738
XML data formats and 662
Target window name setting 269
targets. *See* slots
Template element 628, 629
templates 71, 73, 628–629, 851
temporary components 127
temporary reports 456
testing
connections 596
designs 533, 534
output 353
page layouts 126, 127
page-level security 538–541
parameters 523–524
queries 31
report designs 128
reports 31, 67, 173, 352, 526
security IDs 533, 534
security rules 535, 538
server profiles 596
user selection elements 524
text 851
See also text controls
adding as watermarks 184
adding literal 107, 126
adding to reports 97, 126, 166, 338, 339
adjusting spacing for 352, 353, 356
aligning 117, 141, 737, 741
changing default padding value for 353
changing font attributes for 611, 612
changing style of 140
comparing with dynamic text 339
controlling appearance of 144
copying 696
creating help 440
displaying 312, 338
eliminating blank lines of 347
entering at run time 510
formatting 76, 143, 222, 352
generating HTML output and 344
generating missing values for 319
generating RTF output and 344, 737, 738
indenting 345, 737
misaligned 610
searching for 700, 704, 705, 706, 707
setting default fonts for 610
setting margins for 362–364
setting orientation for 239, 248, 254
setting tab stops for 357
text attributes
adding label controls for 478
applying from toolbar 140
changing 610, 611, 612
dynamic text controls and 338, 738, 741
text boxes 679
text controls 851
See also dynamic text controls; text
adding 166, 338
changing 370
displaying data with 97
displaying security IDs with 535, 536
text editor page 579
text editors 59, 578, 610, 621
text fields 338, 710
text files 342
Text Layout property 352
text patterns. *See* search expressions; wildcards
Text property 107
text strings 845, 855
See also strings
TextAlignRight value 117
TextFormat property 342
TextFormatHTML value 342
TextFormatPlain value 342
TextFormatRTF value 342
TextPlacement properties group 145, 646
textual data. *See* text
Textual Query Editor 602
textual query editor 851

TGA files 151
Then slot 428
Thickness property 147
third-party software 30
thousand separators 679
tick 851
tick interval 851
tick marks
  aligning labels with 256
  changing 244
  defining major 244
  defining minor 245, 247
  described 197
  placing in charts 255
  positioning category markers between 255
  specifying number of 246
TIFF formats 151
tilde (~) character 456
time
  displaying 97
  grouping values 91, 93, 316
time continuums 203
time controls 97
time data types 250
time formats 142–143
time groups 236
time stamps 182
time-sensitive reports 505
title pages 190–193
title/body page lists 165, 190, 192
titles
  adding borders to 239
  adding to charts 238, 239
  adding to reports 18, 76
  changing chart 237, 239
  changing placement of 123
  formatting 239
  setting fonts for 611
  setting orientation for 239
Titles, Legend, and Formatting page
  creating legends from 241
  setting formatting options on 222, 240, 241
  setting up chart titles with 239
TocAddComponent property 466, 468
TocAddContents property 466, 468
TOCIfAllVisible value 466
TOCIfAnyVisible value 467
Toggle Breakpoint command 487, 494
toolbar buttons 36, 43
toolbars 36, 43, 700, 852
Toolbars command 43
toolbox 852
  adding components to 621, 622, 623
  applying predefined representations 625
  associating with classes 623, 624
  customizing 621–624
  restoring defaults 622, 624
  saving configurations 624, 625
  selecting components from 46
ToolTips 36
Top property 362, 363
Total command 76, 116
Total dialog 116
totals
  adding controls for 105, 115
  adding to reports 76
  aggregating data and 16, 114, 310
  building cross tabs and 297, 303, 308
  calculating 116
  comparing 118
  displaying in charts 201
  summarizing data and 16
TQE. *See* textual query editor
tracing 495, 496
trailing spaces 454, 458, 459
transient objects 852
  See also objects
transient reports 456
transparent colors 184, 654
Transparent value 184
TreeViewFontName registry setting 613
trendlines 260–265
Trendlines page (Chart Builder) 260, 263
Trendlines.rod 260
True values. *See* Boolean values
True/False format types 141
TrueType fonts 610
truncated controls 395
truncated cross tabs 391
truncating data 141, 145, 338
trusted execute privilege 852
twips 853
twips (measurement unit) 614, 615
Type 1 fonts 610

type aliases 569
Type components 566
type declaration characters 853
See also data types; declarations
Type page (Chart Builder) 219
Typedef components 569
Types dialog 562, 566, 569
types. *See* data types; semantic types

U

–u option (erdpro) 747
UK currency symbols 742
UL tag 741
unavailable fonts 271, 473
underlined text 738, 741
undoing changes 298
unexpected search results 710
Unicode characters 621, 737, 738, 742, 855
Unicode encoding 854
Unicode standard 854
uniform resource identifiers. *See* URIs
uniform resource locators. *See* URLs
unique IDs 854
unique values 119, 296
Unit Recognition options 616
units of measurement
adding suffixes to 615
customizing 615
selecting 10, 613
setting 614–615
Units page (Options) 9, 614, 615
universal hyperlinks 464
universal hyperlinks. *See* hyperlinks
Universal Resource Locators. *See* URLs
UNIX systems
designing for 67, 473
displaying charts on 196, 271
generating reports for 352
getting font information for 474
viewing images and 151
unordered lists 741
unresolved references 647
unstacking application windows 41
up bars (charts) 250, 258
updating
data 676
data formats 140
report components 47
source code 58
upgrades 746–747
uploading files 597
URIs 854
URLs 854
See also hyperlinks
accessing external libraries with 582
accessing image files and 153
adding dynamic parameters to 462
adding search parameters to 638
adding to charts 269, 270
creating 453, 454, 455, 465
defined 854
displaying Reportlets and 636, 638
displaying reports and 456, 457
importing report sections and 639
server profiles and 595
setting Management Console 620
specifying search paths with 590
Use default values setting 526
user accounts 594
user agent strings 650
user interface (e.Report Designer Professional) 36
User name field (iServer Profile) 595
user names 854
accessing sample databases and 12
adding to server profiles 595
as required parameters 504
user-defined classes 610
user-defined data types 855
users
assigning security IDs to 539
displaying security IDs for 536
entering invalid values 526
prompting for input 504, 506, 509, 518, 526
providing list of choices for 511–518
testing security IDs for 533, 534, 535
testing security rules for 538

V

value axis. *See* y-axis values
value categories. *See* y-axis values
Value Expression button 564

value expressions
See also ValueExp property
adding dynamic text controls and 342
building calculated columns and 116
changing 105, 220
charting data and 210, 222, 223
restrictions for 223
creating 342
defining chart layer 227, 229
defining chart titles and 238
defining custom properties and 558, 564–565
defining data structures and 564
plotting multiple 250
returning ACLs from 535
selecting data points and 207
specifying high and low values with 226
Value properties command 329
ValueExp property 104, 128
See also value expressions
ValuePlacement property 311
values 855
See also data
adding to drop-down lists 512
adding to link expressions 454
aggregating data and 76, 114
assigning literal 341
assigning to parameters 678, 681
assigning to radio buttons 512
binding to controls 104
changing 105, 496, 499
checking 128, 523
comparing 202
conditionally formatting data and 155, 156
cross tabulation and 296, 310, 312, 319, 330
customizing properties and 559, 563
defined 855
defining programmatically 520
describing multiple 564
describing single 564, 566
displaying in scripting controls 652
displaying missing 319, 321, 323
displaying projected 203
displaying sample 608
entering invalid 526
entering overflow characters for 144
entering property 614
getting incorrect 485
getting maximum 114
getting minimum 114
grouping data and 88, 90, 91
interpolating missing 256
labeling chart 247, 248, 249
linking to specific 458, 459, 463
master-detail relationships and 520
monitoring 496
placing limitations on 569
plotting empty 250, 256–257
plotting linear 200
plotting missing 232, 234
prompting for 506, 518, 678
scaling chart 246–247
searching for single 700
searching on literal 704
selecting at run time 510
setting maximum/minimum 247
setting measurement units for 614, 615
showing range of 203
showing relationships to whole 201
specifying default 509, 510, 513
specifying optional 504
specifying required 504
verifying user-specified 526
ValuesVertical value 311
ValueType property 564
Varchar data type 855
variable-length text 338
See also dynamic text controls
variables 855
adding conditional sections and 405
assigning security IDs to 541
changing values of 496, 499
customizing properties and 559
defined 855
describing single items 566
displaying names 487, 498
displaying values of 487, 488, 491, 498
monitoring 488, 489, 491, 496, 498
Variables Browser command 487
Variables page 855
Variables window 485, 488–489, 496, 497
Variant data type 856
VBScript 642
Verify Design command 129, 488

verifying input values 526–527
version information 594
version numbers 736
Version options (Publish and Preview) 597
versions 460
Vertical property 646
VerticalOverlap property 333
VerticalPosition property 382, 387
VerticalPositionFrameBottom value 388
VerticalPositionFrameMiddle value 388
VerticalPositionFrameTop value 387
VerticalSize property 378, 380, 383, 385
VerticalSizeFrameRelative value 378, 380, 385
VerticalSizeRelative value 378, 380, 383
View Browser Code command 647, 653
View command 37, 59
View menu 37, 687
view perspective
  displaying reports in 59, 686–694
  generating multipage reports and 677, 678
  missing fonts and 473
  overview 39
  refreshing reports in 678
View service 651, 664
view time 856
view window. *See* view perspective
View with full read privilege setting 538
Viewer 856
viewers. *See* DHTML Viewer; PDF Viewer
viewing
  access control lists 535
  aggregate data 297, 299, 310
  application status bar 36
  balloon help 440
  bi-directional text 341
  charts 271
  class hierarchies 55
  class-specific methods 492
  component names 45
  component properties 52, 56, 57
  component references 584, 586
  components 44
  computed values 312
  context menus 608
  cross tabs 299, 310, 331, 333
  custom browser code 647, 653
  data 75, 97
  data groups 95
  database fields 21
  database tables 21, 74
  detail reports 548
  DHTML reports 642
  errors 487
  external functions 572
  grid lines 10
  images 151
  large numbers 145
  libraries in designs 574, 590
  link expressions 464
  literal values 341
  member variables 568
  method calls 487, 489
  overflow characters 144
  page layouts 123, 163
  PDF output 653
  query output 474
  query statements 21
  related information 691
  report designs 18, 20, 44
  report layout styles 17
  report parameters 519, 521
  report structures 6, 23
  Reportlets 636
  reports 59, 67, 472, 686–694
  sample data 270, 608, 645
  sample reports 686
  scripting control values 652
  search results 52, 704, 708, 709
  security IDs 535, 536
  source code 59, 493
  specific parts of a report 688, 690, 691
  specific report pages 173, 530, 688
  status messages 27
  table of contents 689
  text 312, 338
  toolbars 43
  units of measurement 613
  variables 487, 488, 491, 498
  XML properties 666
  XML reports 667
viewing events 440
views 693, 694, 710
  defined 856

virtual security IDs 531, 533, 541–543
virtual x-servers 271
visibility options (properties) 561
visibility properties 636
visible components 47
visible page numbers 181, 537
visible privilege 856
Visible value (CSS attribute) 646
VisiblePageNofM value 537
VisiblePageNumber value 534
Visual Basic 856
visual components 856
visual distortion 646
Volume Administration console. *See* Management Console
Volume name field (iServer Profile) 595
volumes. *See* Encyclopedia volumes

W

wall colors (charts) 241
Watch command 487
Watch window
  debugging and 485, 491
  editing in 499
  monitoring variables and 496, 498
watermarks 183, 184
web browsers
  See also progressive viewing
  accessing reports with 594
  aligning output for 646
  changing scaling values 651
  clipping output for 645
  converting special characters for 643, 644
  creating menus for 648, 654, 656
  defining hyperlinks for 456, 464
  displaying browser scripting values and 646, 653
  fixing visual distortion in 646
  generating output for specific 651
  publishing reports and 594
  searching with 663
  unresolved references in 647
  viewing code for 647, 653
  writing code for 644
  XML data formats and 662
web directives 857
web pages 857
  adding images for 151
  adding scroll bars to 645
  aligning output for 646
  building interactive 649
  clipping output for 645
  creating hyperlinks to 456, 464
  defined 857
  determining application context for 651
  displaying cross tabs and 331, 333
  embedding reports in 634, 636, 637
  generating output for 650, 651
  getting current scaling factor 650
  limitations of 333
  optimizing output for 392
  printing Reportlets in 640
  relative URLs and 640
  saving 640
  scaling 651
  searching 448
web servers 464, 572, 582, 594, 857
web-based hyperlinks 454, 460, 464
web-based reporting 169
web-based URLs 456
weekly reports 711
Welcome dialog 35
well-formed XML 857
WHERE clause 22, 118, 119, 507
  See also SQL statements
WHERE statements. *See* WHERE clause
white space, minimizing 334, 352, 354
WidowAndOrphanControl property 365
Width property
  cross tabs 326
  dynamic text controls 349
  line styles 151
  page size 168, 169
wildcard characters 703, 705, 858
wildcards 858
Window menu 693
windows 813, 814
  See also application window
  closing 43, 717
  minimizing 20
  opening design 5
  opening multiple 693–694
  personalizing 41, 693

restoring defaults for 44
saving layouts 41, 693
splitting 694
Windows registry keys 613, 627
Windows systems
customizing wizards for 627
designing for 67, 473, 480
displaying charts on 196, 271
displaying reports on 352, 472
getting font information for 474
wizards
See also specific report wizard
accessing report 10, 69
creating charts with 196
creating reports with 69–70
creating subreports with 411
customizing item list for 627
limitations of 72
word processing applications 59
WordPad 697
WordWrap property 646
work area 36, 38
working directory 617
write privilege 858

X

X Axis page (Axes) 245, 246
X Axis page (Labels) 248, 249
X value expressions 210, 211, 224
x-axis labels 197, 220
x-axis title 197
x-axis values
See also axes values; charts
adding titles to 238
aligning data points and labels for 255, 256
arranging labels on 255
changing line styles for 246
formatting labels for 248
grouping 224
labeling 248, 249
location of 197
overview 243
positioning data points for 255
positioning intersecting 245
scaling 246
selecting data for 224
setting options for 245
sorting 224
XML (defined) 662, 788
XML converter 858
XML Data command 665
XML data formats 662, 663, 668, 858
XML data mapping example 669
XML Data property group 666
XML data streams 858
XML display formats 858
XML documents 662, 857
XML files
See also configuration files
generating 665, 667, 669
limiting size of 667
XML Files value 667
XML prolog information properties 666
XML properties 665, 666–667
XML reports 859
controlling appearance of 667
debugging 667
default encoding for 669
designing for 662, 664, 665
displaying data in 663
generating output for 664, 665, 668
generating prologs for 663
publishing to 664–671
searching 663
structuring 667
viewing 667
XML schemas 837, 838
See also ROM schemas
defined 838
XML tags 662
XMLAddContents property 668
XMLAttribute value 666, 668
XMLAttributes property 668
XMLCharSet property 667
XMLCustom value 666, 668
XMLDataProlog method 663, 669
XMLDocType property 667
XMLElement value 666, 668
XMLEmptyAttribute value 668
XMLFileDescription property 667
XMLFileExtension property 667
XMLIgnore value 668

XMLIndent property 667
XMLMimeType property 667
XMLTag property 666, 668
XMLType property 666, 668
x-servers 271
Xvfb server 271

Y

Y Axis page (Axes) 244, 245, 246
Y Axis page (Labels) 248, 249
Y Axis page (Plot)
  plotting empty values 257
  selecting up or down bar options 258
  setting bar formatting options on 266
  setting bar position on 267
  setting drop line options 260
  setting high-low line options 259
  setting marker and label positions 255
  showing markers and lines 257
Y Axis page (Point Labels) 252, 253, 254
Y Axis page (Scale) 246, 247
Y Axis page (Series) 250, 251
Y Axis page (Trendlines) 263, 264
Y value expressions 210, 211, 223
y-axis labels 197
y-axis title 197
y-axis values
  See also axes values; charts
  adding drop lines to 260
  adding high-low lines for 259
  adding titles to 238
  adding trendlines for 263
  adding up or down bars for 258
  changing line styles for 246
  connecting data points for 257
  formatting bars for 266
  formatting labels for 248
  labeling 248, 249
  location of 197
  overview 244
  plotting dual 204
  plotting empty values for 257
  positioning intersecting 245
  scaling 246, 247
  selecting data for 223
  setting minor tick marks for 247
  setting options for 244
  setting point label text for 252, 253
Year function 102
yearly sales reports 210
Yes/No format types 141

Z

zero values 256
zoom settings 652, 695
zooming 695
Z-order (components) 139