# Contents

Introduction ............................................................ ix
About Actuate e.Reporting Suite 5 ................................... ix
About Designing e.Reports Java Edition ............................ xiv
About Actuate e.Report Designer product .......................... xv
Online documentation .................................................. xvi
  Using online manuals ................................................. xvii
  Online help .......................................................... xvii
  Using the Actuate online help system ............................. xvii
Typographical conventions ........................................... xix
Syntax conventions ...................................................... xx

Chapter 1
**Building your first report** ................................. 1
About Actuate e.Report Designer Java Edition .................... 2
Before you begin ...................................................... 3
Lessons in building the example report ............................. 3
  Running RunACJ.bat on diverse platforms ....................... 4
  UNIX ............................................................... 4
  Windows ......................................................... 4
Lesson 2. Loading a template ......................................... 4
  Loading the Chart template ....................................... 4
  Report editor ....................................................... 6
  Understanding the layout pane .................................. 6
Lesson 3. Creating or modifying a report ........................... 6
Lesson 4. Previewing the report ....................................... 7
Lesson 5. Navigating the report ....................................... 7
Lesson 6. Generating the report ....................................... 8
Lesson 7. Working with the Table of Contents ..................... 9
  Viewing details .................................................... 10
  Designing the Drill Down TOC .................................. 11
Lesson 8. Printing the report ......................................... 13
Lesson 9. Exporting the report ....................................... 14

Chapter 2
**Understanding the design process** ....................... 17
Overview of the report design process ............................ 18
  Planning a report ................................................ 18
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About report templates</td>
<td>19</td>
</tr>
<tr>
<td>Working with sample report templates</td>
<td>21</td>
</tr>
<tr>
<td>About the Report Explorer</td>
<td>21</td>
</tr>
<tr>
<td>Analyzing the report design</td>
<td>22</td>
</tr>
<tr>
<td>About sections and section IDs</td>
<td>22</td>
</tr>
<tr>
<td>About zones</td>
<td>23</td>
</tr>
<tr>
<td>About report headers</td>
<td>24</td>
</tr>
<tr>
<td>About page headers</td>
<td>24</td>
</tr>
<tr>
<td>About details</td>
<td>24</td>
</tr>
<tr>
<td>Naming zones</td>
<td>25</td>
</tr>
<tr>
<td>About groups and nested groups</td>
<td>25</td>
</tr>
<tr>
<td>About report footers</td>
<td>26</td>
</tr>
<tr>
<td>About page footers</td>
<td>26</td>
</tr>
<tr>
<td>Hiding and showing portions of a report</td>
<td>26</td>
</tr>
<tr>
<td>Hiding controls and zones</td>
<td>26</td>
</tr>
<tr>
<td>Resizing zone height</td>
<td>27</td>
</tr>
<tr>
<td>Laying out a report</td>
<td>29</td>
</tr>
<tr>
<td>Adding color</td>
<td>29</td>
</tr>
<tr>
<td>Adding borders</td>
<td>30</td>
</tr>
<tr>
<td>Paginating a report</td>
<td>32</td>
</tr>
<tr>
<td>Repeating headings on subsequent pages</td>
<td>33</td>
</tr>
<tr>
<td>Using database field names as column headings</td>
<td>34</td>
</tr>
<tr>
<td>Creating a new report</td>
<td>34</td>
</tr>
<tr>
<td>Creating your own templates</td>
<td>36</td>
</tr>
<tr>
<td>Saving a report design</td>
<td>40</td>
</tr>
<tr>
<td>Creating batch files to launch reports</td>
<td>40</td>
</tr>
<tr>
<td>Creating your own templates</td>
<td>36</td>
</tr>
<tr>
<td>Using database field names as column headings</td>
<td>34</td>
</tr>
<tr>
<td>Creating a new report</td>
<td>34</td>
</tr>
<tr>
<td>Creating your own templates</td>
<td>36</td>
</tr>
<tr>
<td>Saving a report design</td>
<td>40</td>
</tr>
<tr>
<td>Creating batch files to launch reports</td>
<td>40</td>
</tr>
<tr>
<td>Chapter 3</td>
<td></td>
</tr>
<tr>
<td><strong>Working with controls</strong></td>
<td>41</td>
</tr>
<tr>
<td>About formatting options</td>
<td>42</td>
</tr>
<tr>
<td>Using formatting options</td>
<td>42</td>
</tr>
<tr>
<td>Using the formatting toolbar</td>
<td>44</td>
</tr>
<tr>
<td>Setting control properties</td>
<td>45</td>
</tr>
<tr>
<td>Setting text and fill color of a control</td>
<td>45</td>
</tr>
<tr>
<td>Using Labels</td>
<td>46</td>
</tr>
<tr>
<td>Modifying a label</td>
<td>47</td>
</tr>
<tr>
<td>Dating reports</td>
<td>49</td>
</tr>
<tr>
<td>Using record counts</td>
<td>50</td>
</tr>
<tr>
<td>Understanding reset criteria</td>
<td>50</td>
</tr>
<tr>
<td>Using reset criteria in a grouping</td>
<td>51</td>
</tr>
<tr>
<td>Adding graphics</td>
<td>53</td>
</tr>
<tr>
<td>Inserting images</td>
<td>54</td>
</tr>
</tbody>
</table>
Opening a report and viewing a SQL query ........................................... 90
Inserting a data field ........................................................................... 93
Uniquely identifying data ...................................................................... 95
Understanding section queries .............................................................. 95
Modifying the section query or its ID .................................................... 96
Creating SELECT and FROM clauses ................................................... 96
Creating WHERE and ORDER BY clauses .............................................. 98
Generating the WHERE clause ............................................................. 98
Creating an import query section .......................................................... 99
Defining joins between tables ............................................................... 102
Defining relationships among columns ................................................. 104
Using the Relationships page to remove joins ....................................... 104
Defining filter conditions .................................................................... 107
Sorting query results ........................................................................... 107
Grouping information .......................................................................... 108
Requirements for creating grouped reports ........................................... 108
Setting grouping by formula ................................................................. 112
Defining joins between tables ............................................................... 112
Using multiple-section reports with multiple groups ................................ 113
Creating master-detail section reports ................................................ 113
Using the Master Detail Query wizard ................................................ 113
Invoking the Master Detail Query wizard ............................................. 114
Examining a multiple-section report ..................................................... 116
About nested sections ......................................................................... 117
Query substitution for nested sections ................................................... 118
Formatted string substitution ................................................................. 120
Understanding date field formatting guidelines ................................... 122
Understanding data field formatting guidelines in application data source modules ........................................... 122
Understanding three-part SQL .......................................................... 122
Understanding a MetaData data source ................................................. 123
Introducing a MetaData layer ............................................................... 123
Working with parameter fields ............................................................ 123
Customizing parameter fields ............................................................. 128
Working with stored procedures or functions ....................................... 134
Supported types of stored objects ....................................................... 134
Examples of stored objects ................................................................ 134
Oracle example ................................................................................... 135
Sybase SQL Anywhere example .......................................................... 136
MS SQL Server 7 example ................................................................. 137
MS Access example ............................................................................ 137
Understanding requirements ............................................................... 137
Entering constant values ......................................................... 138
jConnect driver ................................................................. 138
Oracle Thin driver ............................................................. 138
MSAccess ODBC driver ....................................................... 139
MS SQL Server ................................................................. 139
Output parameters ............................................................. 139
Inserting a stored object programmatically ......................... 139
Inserting stored objects manually ....................................... 144
Working with multiple result sets ........................................ 146
Managing an empty result set .............................................. 148
Limiting the result set to specific rows ................................... 148
Saving result set columns in a template ............................... 149

Chapter 6

Customizing e.Report Designer Java Edition ......................... 151

Customizing designer functions .......................................... 152
  Options Properties .......................................................... 152
  General Page .................................................................. 152
  Control Defaults ............................................................ 153
  Workspace .................................................................... 153
  Query ........................................................................... 155
  Template ....................................................................... 156
  Output ........................................................................... 156
  Debug .......................................................................... 157
Customizing general report properties ................................. 158
  Report Properties ........................................................... 158
Customizing report page properties ..................................... 159
  Page Setup .................................................................... 159
  Export properties ............................................................ 160
Understanding sample ACJDesigner.properties ..................... 163

Chapter 7

Presenting data in charts, graphs, tables, and crosstabs ...... 167
Types of graphs and charts .................................................. 168
  Selecting data fields ........................................................ 170
Understanding X-axis and Y-axis .......................................... 173
  Defining the X-axis .......................................................... 173
  Defining the Y-axis .......................................................... 174
Sending charts and graphs to an output device ...................... 175
Creating asymmetrical grids ................................................ 175
  Inserting an AsyncGrid into a report ................................. 176


**About Actuate e.Reporting Suite 5**

Actuate is the leading provider of information delivery solutions for e.Business. e.Business customers use Actuate® e.Reporting Suite 5 to develop and deploy high resolution structured content to hundreds of thousands of users. Actuate takes web reporting to the next level by providing options for needs as varied as seamless personalized web pages and traditional online and printed reports.

Actuate’s customer list includes commercial banks, securities, financial services, insurance, high tech, telecom, .com, internet, global 2000, and federal government. OEMs, system integrators, and others building e.Business sites for information delivery face challenges. Actuate e.Reporting Suite 5 offers the following solutions.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Actuate solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivering high resolution information</td>
<td>DHTML provides a fast, no-download option</td>
</tr>
<tr>
<td>Viewing structured content</td>
<td>Supporting standard browsers means there is no need to support installations of plug-ins for hundreds of thousands of users</td>
</tr>
<tr>
<td>Compromising information display because of lack of integrated tools</td>
<td>Provides template-based design and display</td>
</tr>
<tr>
<td>Exploding use of web-based content delivery</td>
<td>Ability to support a million hits per day per CPU</td>
</tr>
<tr>
<td>Delivering personalized secure information</td>
<td>Open security directory integration and page security</td>
</tr>
</tbody>
</table>
Designing e.Reports Java Edition

Actuate tools and reports do the following:

- Solve complex data access problems.
- Solve formatting problems that go beyond the scope of other tools.
- Scale to support hundreds of thousands of users.

The following summary describes the products in Actuate e.Reporting Suite 5.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Actuate solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusing existing integrated content</td>
<td>Open server provides access to content from other applications</td>
</tr>
<tr>
<td>Maintaining data integrity on hard copy</td>
<td>PDF provides high-resolution printed copy</td>
</tr>
<tr>
<td>Transferring information into other applications</td>
<td>XML output provides access to data across applications</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product name</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuate e.Report Designer Professional</td>
<td>An object-oriented application used by professional developers of structured content to design, build, and distribute report object designs for delivery on the Web. The Actuate Basic Language and Actuate Foundation Class Library support extensive customization capabilities.</td>
</tr>
<tr>
<td>Actuate SDK (Software Development Kit) included as part of Actuate e.Report Designer Professional</td>
<td>Actuate ActiveX Controls embed Actuate reporting functionality into custom applications. Actuate Requester API accesses attributes and values of report parameters, changes the values of report parameters, controls how and when a report is generated, displays and prints reports, and configures report print setup. Access the Requester API using Actuate Basic, Visual Basic, C, or C++. Actuate search extension API supports developing search extensions to transfer data to any third-party productivity or analysis tool.</td>
</tr>
<tr>
<td>Product name</td>
<td>Use</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
</tr>
<tr>
<td>Actuate archive driver</td>
<td>Supports the use of third-party archiving software and hardware.</td>
</tr>
<tr>
<td>Actuate e.Report Designer</td>
<td>An application that complements e.Report Designer Professional and is used by business users to design and distribute a variety of reports. These reports require no programming. This application supports both modifying complex reports and using sophisticated components from libraries.</td>
</tr>
<tr>
<td>Actuate e.Report Designer Java™ Edition</td>
<td>A report development application used by Java developers to design and distribute a variety of reports. 100% Java compliant, the e.Report Designer Java Edition includes both AWT and Swing APIs.</td>
</tr>
<tr>
<td>Actuate e.Reporting Server</td>
<td>A server application that generates Live Report Documents, manages them in the Report Encyclopedia®, and makes them available to users. This product includes Actuate Administrator Desktop, an application for system and network administrators to manage and control one or more Actuate report servers. This product also includes Actuate ReportCast™ that transforms the Report Encyclopedia into a dynamic, secure website. ReportCast provides the foundation for Channels and seamless integration with other web sites.</td>
</tr>
<tr>
<td>Product name</td>
<td>Use</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

This product includes Actuate Administrator Desktop, an application for system and network administrators to manage and control one or more Actuate report servers.

This product also includes Actuate ReportCast that transforms the Report Encyclopedia into a dynamic, secure web site. ReportCast provides the foundation for Channels and seamless integration with other web sites. |
| Actuate e.Reporting Server Progress Edition | A server application designed to work exclusively with Progress databases that generates Live Report Documents, manages them in the Report Encyclopedia, and makes them available to users.

This product includes Actuate Administrator Desktop, an application for system and network administrators to manage and control one or more Actuate report servers.

This product also includes Actuate ReportCast that transforms the Report Encyclopedia into a dynamic, secure web site. ReportCast provides the foundation for Channels and seamless integration with other web sites. |
| Actuate Advanced e.Reporting Server | An application that adds two capabilities to the basic e.Reporting Server, open server and page security.

- Open server supports the use of third-party report generators with the Actuate e.Reporting Server.

- Page security supports personalized viewing of parts of reports for various users |
<table>
<thead>
<tr>
<th>Product name</th>
<th>Use</th>
</tr>
</thead>
</table>
| Actuate Advanced e.Reporting Server Progress Edition | An application designed to work exclusively with Progress databases that adds two capabilities to the basic e.Reporting Server, open server and page security.  
- Open server supports the use of third-party report generators with the Actuate e.Reporting Server.  
- Page security supports personalized viewing of parts of reports for various users |
| Actuate e.Analysis | An application used to transform data from an Actuate e.report into interactive information. Users can view and analyze data to determine relationships and trends. Actuate e.Analysis is an optional product that installs with ReportCast and the e.Reporting Server and extends its functionality. |
| Actuate End User Desktop | An application used by end users to request, generate, view, and print report documents. The ReportQuery™ capabilities enable seamless transfer of data from an Actuate report to any productivity tool or analysis tool. |
| Actuate Viewer | Application for end users to find, view, and print report documents. The ReportQuery capabilities are also part of the Actuate Viewer. |
| Actuate Live Report Extension (LRX) | Application for end users that works with both Microsoft Internet Explorer and Netscape Navigator to support report viewing and printing on the Web. |

Actuate Viewer and Actuate Live Report Extension (LRX) are included with all products except Actuate e.Report Designer Java Edition.
About **Designing e.Reports Java Edition**

*Designing e.Reports Java Edition* describes how to extract critical business data from diverse data sources and transform it into effective, information-rich e.reports for use on the Web. These reports display in standard browsers such as Netscape Navigator and Microsoft Internet Explorer. *e.Report Designer Java Edition* provides direct access to Java data sources, ODBC database data sources, and text files.

*Designing e.Reports Java Edition* includes the following chapters:

**Introduction.** This chapter provides an overview of this guide, the Actuate e.Report Designer Java Edition documentation, and the typographical conventions used.

**Chapter 1. Building your first report.** This chapter provides a tutorial for designing sample reports.

**Chapter 2. Understanding the design process.** This chapter explains the basic report design process. It introduces templates, the report explorer, zones, report layout, and launching reports.

**Chapter 3. Working with controls.** This chapter discusses formatting options, control properties, and web links.

**Chapter 4. Connecting to a database.** This chapter discusses connecting to data sources.

**Chapter 5. Querying the database.** This chapter discusses database queries, defining joins in tables, defining filter conditions, sorting, grouping, creating master-detail reports, and working with parameters, stored procedures, and functions.

**Chapter 6. Customizing e.Report Designer Java Edition.** This chapter discusses customizing designer functions and report properties. The chapter includes a discussion of a sample properties file.

**Chapter 7. Presenting data in charts, graphs, tables, and crosstabs.** This chapter discusses types of graphs and charts and how to design these reports. The chapter includes sections about tabular, columnar, and crosstab reports.

**Chapter 8. Working with formulas.** This chapter discusses formulas, formatting formulas and functions, and using different types of functions.

**Chapter 9. Conditional formatting.** This chapter discusses using scripts, formatting properties, using operators, and commonly used scripts.
About Actuate e.Report Designer product

Actuate e.Report Designer documentation includes printed manuals, an installation guide, online help, user documentation in PDF format, and release notes. Information about the product that could not be included before the book printing deadline is in the release notes.

The Actuate web site, http://www.actuate.com, contains late-breaking news about the product and its features, as well as product update information. To obtain the password necessary to access the portion of the web site available only to customers, telephone Actuate Customer Support. The engineers in Actuate Customer Support can also help you with technical questions about the product according to your service contract. The Customer Support telephone number, fax number, and e-mail information can be found among the printed materials in the product box.

Actuate e.Report Designer Java Edition includes a Demofolder that contains numerous report examples. The documentation discusses a number of these sample reports folders.

The printed and online documentation includes the following manuals.

<table>
<thead>
<tr>
<th>For information about</th>
<th>See the following</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation and upgrading</td>
<td>Installation and upgrading guide</td>
</tr>
<tr>
<td>Configuring an EJB sample application with Sun J2EE</td>
<td>Configuring the EJB Sample Application with Sun J2EE 1.2.1</td>
</tr>
<tr>
<td>Configuring an EJB sample application with Silverstream</td>
<td>Configuring and running the EJB Sample Application with Silverstream 3.7</td>
</tr>
<tr>
<td>Late-breaking information about the software and documentation</td>
<td>Release notes</td>
</tr>
<tr>
<td>Overview of Actuate reporting concepts</td>
<td>Designing e.Reports Java Edition</td>
</tr>
<tr>
<td>How to build your first report</td>
<td></td>
</tr>
<tr>
<td>How to design reports using the graphical user interface</td>
<td></td>
</tr>
<tr>
<td>Basic data source connection</td>
<td></td>
</tr>
</tbody>
</table>
For information about | See the following
---|---
Using the APIs  
Overview of classes  
Database reporting  
Server solutions | [Programming e.Reports Java Edition](#)

Using the Swing and AWT APIs | [e.Report Designer API Java Edition](#)

Terminology map  
Glossary | [Actuate Glossary](#)

---

**Online documentation**

The information in the printed manuals is also available as online books in Adobe Acrobat PDF format and in the online help system for the Actuate products. For products without a Windows interface, such as the various versions of Actuate e.Reporting Server, Actuate e.Report Designer Java Edition, Actuate ReportCast, and Actuate e.Analysis, we provide HTML help files. The HTML files install automatically with the product. These files can be viewed with standard browsers.
Using online manuals

The online manuals install automatically with the product. On the product CD, you can also find those files in the Manuals directory. Open the introductory PDF file to get an overview of the manuals. The items in the table of contents and the page numbers in the index both contain links to the appropriate topics in the text. In the index, you access the link by positioning your cursor over the page number, not the topic.

Online help

Actuate products provide both context-sensitive online help about the product and report-specific online help about the report you are viewing. Actuate e.Reporting Suite makes it possible for developers to create customized report-specific online help.

Using the Actuate online help system

Use two windows to access and view information in the e.Reporting Suite help system. The window on the left displays the table of contents or the index of the online help system. The window on the right displays the contents of the online help topics.

The tabs at the top of the left window access different views. Use these tabs to switch views among the Table of Contents, Index, and Search. The Table of Contents provides an overview of the help file contents. For example, in the preceding window, you see the introduction to the documentation for the e.Reporting Server product. The Search tab accesses any word in the help files.
The following two illustrations show an example of the Index and the result of an Index search. The following illustration shows the result of the search as it appears in the window on the right.

To view the topic double-click the topic in the list. The topic displays in the window on the right.

Use the Search tab to find all instances of a keyword in the help files. The following illustration shows the results of a search for the keyword, templates. The topics containing the keyword appear under Search. Double-click a topic in the Search list to display the topic in the right window.
## Typographical conventions

The following table describes the typographical conventions used in this guide.

<table>
<thead>
<tr>
<th>Item</th>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code examples</td>
<td>Sans serif</td>
<td>Dim As String</td>
</tr>
<tr>
<td>File names</td>
<td>Initial letter capitalized.</td>
<td>Detail.roi</td>
</tr>
<tr>
<td></td>
<td>Except e_Report Designer Java Edition, where file names are case sensitive.</td>
<td></td>
</tr>
<tr>
<td>Key combination</td>
<td>A + sign between the keys means to press both keys at the same time</td>
<td>Ctrl+Shift</td>
</tr>
<tr>
<td>Menu items</td>
<td>Capitalized. No bold.</td>
<td>File</td>
</tr>
<tr>
<td>Submenu items</td>
<td>Separated from main menu item with small arrow</td>
<td>File ➔ New</td>
</tr>
<tr>
<td>User input or user response</td>
<td>Sans serif</td>
<td>M<em>16</em></td>
</tr>
</tbody>
</table>
# Syntax conventions

The following table describes the symbols used to present the syntax.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>[]</td>
<td>Optional item</td>
<td>[Alias&lt;alias name&gt;]</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Argument you must supply</td>
<td>&lt;expression to format&gt;</td>
</tr>
<tr>
<td>{}</td>
<td>Groups two or more mutually exclusive options or arguments</td>
<td>{While</td>
</tr>
<tr>
<td></td>
<td>Separates mutually exclusive options or arguments in a group</td>
<td>Exit {Do</td>
</tr>
</tbody>
</table>
Chapter 1, Building your first report

This chapter contains the following topics:

- About Actuate e.Report Designer Java Edition
- Before you begin
- Lessons in building the example report
About Actuate e.Report Designer Java Edition

Actuate e.Report Designer Java Edition can be run as an application, an applet, or an embedded module inside a larger application. For instructions about installing and running e.Report Designer Java Edition, see *Installing and Upgrading e.Report Designer 5.0 Java Edition*.

e.Report Designer Java Edition is a 100 percent pure Java web reporting tool that includes a report designer and a report server for the Unix or Windows platforms. e.Report Designer Java Edition can report on data residing in the following types of data sources:

- Databases
- Java applications
- Text files
- EJB servers

You can view the report in the e.Report Designer Java Edition Viewer, export reports to Adobe System's PDF file format, standard HTML, high-resolution HTML, or ASCII-CSV file formats. You can send the report to a printer for a WYSIWYG hardcopy. e.Report Designer Java Edition provides support for several Java Virtual Machines, including those from Microsoft and Netscape. The e.Report Designer Java Edition server supports numerous features such as:

- Ad Hoc reports
- Cached reports
- Scheduled reports
- Remote administration
- Page-on-demand

You can analyze data and determine trends using e.Report Designer Java Edition to create:

- Crosstab reports
- Multiple crosstab reports
- Multiple section reports
- Form reports
- Parameterized fields reports
- Charts and graphs

e.Report Designer Java Edition provides an easy-to-use report designer and extensive formatting capabilities such as fonts, colors, alignment, word-wrap,
page breaks, overlaying controls and report styles such as summarized, tabular and columnar reports to help format data.

Before you begin

The lessons in this chapter assume you are running the ACJDemo report, which is a Java Product Demo file. For information about how to run this report, refer to “Lesson 1. Starting e.Report Designer Java Edition,” in this chapter.

When you create a report, e.Report Designer Java Edition stores it in a report file that has a .jod extension. You can run all templates located in e.Report Designer Java Edition by choosing ACJ50Swing\templates\ForACJDemoApp. If you are testing JDBC connectivity with the Northwind.mdb database, use templates located in the ForNorthwind directory located in ACJ50Swing\templates\ForNorthwind.

Lessons in building the example report

Lesson 2. Loading a template
Lesson 3. Creating or modifying a report
Lesson 4. Previewing the report
Lesson 5. Navigating the report
Lesson 6. Generating the report
Lesson 7. Working with the Table of Contents
Lesson 8. Printing the report
Lesson 9. Exporting the report


When you open e.Report Designer Java, a demo called ACJDemoApp displays. This application demonstrates the powerful design capabilities of e.Report Designer Java Edition and how you can use it, and other templates like it, to develop application data reports.
Running RunACJ.bat on diverse platforms

If you run e.Report Designer Java Edition on a non-32 bit platform, you must modify the RunACJ.bat file to conform to your platform. If you run e.Report Designer Java Edition on UNIX, change permissions using chmod 755 so that you can execute the file.

UNIX

You run this batch file on the command line as follows:
1. Open a command window on UNIX.
2. Go to the directory in which you installed e.Report Designer Java Edition.
3. Start e.Report Designer Java Edition by entering:
   
   RunACJ.bat

Windows

1. In Windows Explorer, navigate to the directory in which you installed e.Report Designer Java Edition.
2. Double-click RunACJ.bat.

e.Report Designer Java Edition loads a template called ChartTemplate.jod that shows a sample report with charts and graphs. The designer window displays the report that e.Report Designer Java Edition generates based on the template design and data source called CSVSource.txt.

Lesson 2. Loading a template

A model report can be a basis for creating other reports. e.Report Designer Java Edition includes examples that store the exact format of a report to the hard drive. Templates included with e.Report Designer Java Edition and instructions on loading these templates and on how to use them follow.

Loading the Chart template

1. Choose File ➤ Open.

![Image of the Load file from URL dialog box]
2 Choose Browse.

3 Navigate to the templates directory and select ACJ50Swing\templates\ForACJdemoApp\ChartTemplate.jod.

4 Select ChartTemplate.jod from the list of templates. Choose OK or double-click the file name.

ChartTemplate loads into your report designer.
Report editor

The report editor has two panes:

- The structure pane is on the left.
- The layout pane is on the right.

The structure pane displays parts of the report called zones. Each zone contains a section of the report such as the report header and details. Each section has different characteristics. For instance, a report header is usually a title and prints only once at the start of the report, while Details contains calculated values and data from the database and can print on many pages.

Understanding the layout pane

The layout pane contains items with different text values. These items are controls. Here is a partial list of controls you can use to create a report:

- Label
- Data field
- Formula
- Image
- Chart

Lesson 3. Creating or modifying a report

Modifying an existing report is an easy way to create a new report. You can perform most major tasks, including creating a new report, with a wizard. Wizards simplify the task, guiding you through the process one step at a time. You can create most types of reports using File ➤ New. When you issue these commands, New Report appears. In this lesson you modify the ChartTemplate report design.

Available or active menu options are highlighted and readily apparent during report design. The task names on these menus relate to the task function, which makes the design process more intuitive.

How to add today’s date to the report header

1. On the Design page, choose Insert ➤ Today’s Date.
2. Position the cursor in the report header of your report designer.
3. Click to insert the date.
Lesson 4. Previewing the report

Creating a well-designed report is an iterative process in which you change the report many times. As you change it, you preview the changes. In Preview, the current page and the total number of pages in the report display at the bottom left corner of the report.

How to preview the report

Choose Preview. The report is generated and today’s date appears to the right in the page header.

From the Preview page, you can send the report data to a printer, or to any other supported output device such as an HTML file or a PDF file.

Lesson 5. Navigating the report

The Preview page contains the following tools and controls:
Using the toolbar on the Preview page, you can go to the next and previous pages. You can go to a specific page number by choosing Goto Page in the toolbar.

**How to navigate your report**

2. Specify page 2.
   
   Page 2 appears.

3. Choose First Page.
   
   Page 1 appears.

4. Choose View ➔ Zoom.
   
   You can choose from 50% to 200% of the report size. When you zoom out, report clarity decreases, depending on the font used.

5. Scroll the display, using the scroll bars or by dragging the bar to the right of the report.
   
   You are now ready to navigate through your design sessions.

---

**Lesson 6. Generating the report**

As you modify the template during the design of a report, you Preview periodically to make sure your changes are correct. Because querying a data source is time-consuming, e.Report Designer Java Edition queries only when a setting requires it.

During the design process, e.Report Designer Java Edition reflects some changes, such as resizing an object, as soon as they are completed. As a result, there is no need to preview. Changes to data fields, such as adding a data field or inserting an image are not reflected immediately and you must preview in order to view these changes.

If you do not see the changes you made to the report reflected in Preview, generate the report by choosing View ➔ Refresh.


**How to generate your report**

In the layout pane, make a change, such as inserting a formula, that require e.Report Designer Java Edition to query the database.
1 Select the formula icon and drag it to the desired position in your detail zone.

2 Select the Preview page.

3 Choose the Refresh tool. Alternatively, you can choose View ➔ Refresh. 
e.Report Designer Java Edition reconstructs the preview of the report.

Lesson 7. Working with the Table of Contents

Reviewing long, multilevel reports with many pages of information can be cumbersome. e.Report Designer Java Edition displays a Table of Contents in the left pane of the Preview page. The Table of Contents makes multilevel reports easy to view and analyze by revealing the report structure. Most often this structure looks like an outline and is based on report groupings and sections. Choose any item in the Table of Contents and see details related to that item reflected in the layout pane.

How to display the Table of Contents

1 Choose the Preview page.

2 Choose Show/Hide to show Table of Contents in the left pane or to hide it.

If the Table of Contents is hidden, choose View ➔ Table of Contents for a display in the structure pane.
3 Use the mouse to position the cursor on the edge of the splitter window between the Table of Contents and preview pane. Drag the mouse to resize the Table of Contents pane.

4 Choose Report Footer in the Table of Contents window to display the footer.

5 Choose Report Header in the Table of Contents to display the header in the layout pane.

6 Choose the icon pictured above to hide the Table of Contents. By default, it is visible.

**Viewing details**

You need a web browser to view HTML output, Adobe Acrobat Reader to view PDF output, and e.Report Designer Java Edition to view the JOD output.

You can view large reports in a summarized format and then progressively drill down into details. While viewing the report, you can either display or hide details to facilitate viewing the data, as the next section explains.

**How to access report details**

1 Choose the Preview page.

2 If the Table of Contents does not appear, choose View ➤ Table of Contents.

3 Choose the magnifying glass icon to explore the next level or to drill down to details in your report.
4 Choose one of the subordinate items under SEC_00. For example, choose Confections.

e.Report Designer Java Edition extracts information on confections only and displays it.

5 Choose the magnifying glass in Table of Contents to collapse the outline.

**Designing the Drill Down TOC**

e.Report Designer Java Edition generates the Table of Contents from grouping criteria that you specify in the report. You establish the grouping criteria within the following zones:

- Sections
- Groups
- Header and Footer

If you specify multiple levels of grouping, e.Report Designer Java Edition generates a multi-level TOC tree. The Table of Contents uses the value of the grouping criteria as its label.
How to design your Drill Down TOC

1. Choose the Design page.
2. Choose Tools ➔ Drill Down Options.
   TOC Option Sheet appears.

3. Choose Data.
4. Select Details Only to display details of data in the preview pane.
5. Choose TOC Tree.

How to use your drill down results

2. Select the magnifying glass next to the Table of Contents.
3. Select any one of the subordinate details below SEC_00.
4. After you choose one of the items, the data for that item is displayed in the layout pane with no page header, footer, or group header.

This option allows you to send report data to a desired output device, selectively, which is very convenient when you want to print a small portion of a large report. For more information, see “Lesson 9. Exporting the report,” later in this chapter.

Lesson 8. Printing the report

How to print your report

2. Choose the Printer icon in the toolbar.
3. If you want to use the default printer, choose OK. Otherwise, select another printer and choose OK.
   e. Report Designer Java Edition sends the report to the printer.

How to set page orientation and margins

You can select the page orientation, portrait or landscape, for printing the report.

1. Choose the Design page.
Choose Orientation.

Select Portrait mode or Landscape mode.

Choose Margins.

The default for the margins is .50. If your report takes less room, you can adjust the margins accordingly.

Print the report.

Lesson 9. Exporting the report

Keeping most of the report’s formatting information, such as colors, fonts, and page layout, intact, e.Report Designer Java Edition can export reports to the following formats:

- CSV
- DHTML
- EMAIL
- HTML
- PDF

How to export your report


   Current Export Properties appears.

2. Choose the export format for the report by choosing one of the tabs.

   For example, use CSV format to export to the Microsoft Excel spreadsheet.
Specify the file name for the exported report.

**How to set margins**

The rulers at the top and left edge of the report designer indicate the physical size of your report. This information is necessary to aid you in fitting your report on a specific paper size. You often have to adjust the size of a report to successfully export it to a particular file format.

You can select the markings of the rules at every inch or at every centimeter.

1. Choose Design.
2. Choose Tools ▶ Options.
   
   Options appears.

3. Select the units of measurement, either centimeters or inches.
4. Select the divisions per unit measurement. For example, if you select Inches as the units of measurement, select from 1 to 48 divisions per inch.
5. Choose Apply to change the margins for this session.
   
   If you choose Save and Apply, your selection becomes the default for e.Report Designer Java Edition.

   e.Report Designer Java Edition displays a vertical line marking the margins of the page and displays controls placed either on or to the right of this line on the second page of the report.
Chapter 2

Understanding the design process

This chapter contains the following topics:

- Overview of the report design process
- About report templates
- About the Report Explorer
- Analyzing the report design
- Hiding and showing portions of a report
- Resizing zone height
- Laying out a report
- Paginating a report
- Creating batch files to launch reports
Overview of the report design process

This section describes the tasks involved in creating a report design that has been saved as a template for future use. Planning reports on paper helps clarify your requirements.

Planning a report

Many users try to skip the planning process and want to jump right into the design phase. The design phase is an electronic expression of the plan for the report. As you create more reports, you understand how much time sketching out the report on paper saves. To plan the design of informative reports, ask yourself the following questions:

- What is the purpose of the report?
- What is the target audience for the report?
- What is the data source for the report: a database or an application?
- What is the best structure for the data in the report or how do you want to lay out the data?
  - In single or multiple sections
  - In single or nested groups
  - Using report-user defined page breaks
  - Using specific fonts that your UNIX printers support
- What kind of analytical information do you need to present using sub-totals, graphs, or charts?
- What type of output do report users want?
  - Print
  - Online display
  - HTML or DHTML
  - PDF
  - DAT
  - CSV
  - E-mail

After answering these questions, begin using e.Report Designer Java Edition to create your report.
About report templates

In e.Report Designer Java Edition, a template is a report design. The designer window is the primary window in which you create a report template or design. When you open Actuate e.Report Designer Java Edition, the following screen appears.

![Design page, Layout pane, Zones, Report template]

Among other functions, this template demonstrates the powerful design capabilities of e.Report Designer Java Edition.


A formatted example of the report data appears.
### Category: Beverages

<table>
<thead>
<tr>
<th>PRODUCTID</th>
<th>PRODUCTNAME</th>
<th>UNITSINSTOCK</th>
<th>UNITPRICE</th>
<th>Inventory Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chai</td>
<td>39</td>
<td>18.0</td>
<td>$711.0</td>
</tr>
<tr>
<td>2</td>
<td>Chang</td>
<td>17</td>
<td>19.0</td>
<td>$333.0</td>
</tr>
<tr>
<td>59</td>
<td>Chartreuse verte</td>
<td>69</td>
<td>18.0</td>
<td>$12.42</td>
</tr>
<tr>
<td>35</td>
<td>Côte de Blaye</td>
<td>17</td>
<td>265.5</td>
<td>$44.60</td>
</tr>
<tr>
<td>24</td>
<td>Gourmet Fantastica</td>
<td>20</td>
<td>4.5</td>
<td>$0.90</td>
</tr>
<tr>
<td>43</td>
<td>Ipeh Coffee</td>
<td>17</td>
<td>45.0</td>
<td>$3.82</td>
</tr>
<tr>
<td>76</td>
<td>Lahak-shi</td>
<td>57</td>
<td>18.0</td>
<td>$102.6</td>
</tr>
<tr>
<td>67</td>
<td>Laughing Lumberjack Lager</td>
<td>52</td>
<td>140</td>
<td>$7.28</td>
</tr>
<tr>
<td>70</td>
<td>Outback Lager</td>
<td>13</td>
<td>190</td>
<td>$2.23</td>
</tr>
<tr>
<td>75</td>
<td>Rhode Island Kells</td>
<td>125</td>
<td>7.05</td>
<td>$9.66</td>
</tr>
<tr>
<td>54</td>
<td>Squatch Ale</td>
<td>111</td>
<td>140</td>
<td>$15.54</td>
</tr>
<tr>
<td>15</td>
<td>Steakies Stout</td>
<td>20</td>
<td>18.0</td>
<td>$3.60</td>
</tr>
</tbody>
</table>

2 Scroll down and the report data displays in a pie chart located in the report footer, which is also a zone.

![Productwise Report](image)

From this example, you can see that, in the design environment, you use the report designer to define how sections of a report are formatted and displayed. You also see what information is to be placed in the zones of the report. As you develop a template, you should save your template often. Report template or report design files use the .jod extension to distinguish templates from other files.
Working with sample report templates

The preceding template provides one example of the templates that are supplied in report design. These help you understand how to work with the product to build your own report designs. When you RunACJ.bat, the product uses templates located in the Templates\ForACJDemoApp directory. Templates use a .jod file extension. Source code for the ACJDemoApp is located in the Demo\ACJDemoApp directory. The data file, CSVSource.txt, used by ACJDemoApp is in the same directory.

Some of the other demonstration applications in the demo directory also use Actuate e.Report Designer Java Edition report templates.

About the Report Explorer

You use the Report Explorer to view your report. The Report Explorer is a split window that displays the structure of the report in the left pane and details about the selected control in the right pane. The following illustration shows a sample view in the Report Explorer.
Analyzing the report design

e.Report Designer Java Edition divides a report into sections and zones. Each section has its own query to extract data from the data source. A section is a distinct logical unit that results from a query, while a zone corresponds to an actual segment of the report. You can view sections and zones in the Report Explorer.

Based on your report requirements, you designate the data contained in each section. The preceding template demonstrates that data is formatted and positioned in the details zone of the report design. In this report, each section corresponds to a different product category. The sections appear under the Table of Contents located in the left pane of the Report Explorer. After the design process completes, your formatted data displays in the right pane and the Table of Contents in the left.

Table Of Contents

- Report Header
- SEC_01
  - Beverages
  - Condiments
  - Confections
  - Dairy Products
  - Grains/Cereals
  - Meat/Poultry
  - Produce
  - Seafood
- Report Footer

Table of Contents lists the names of the various sections of the report as well as other report zones. In this instance, sections are delineated by the logical division of the products into categories that correspond to pages of the report and the query. Eight categories of products for the report are listed to the left. Note that, in this example, each of these sections has its own page.

About sections and section IDs

Since e.Report Designer Java supports more than one section in a report, each section is required to have a unique identifier called the Section Identifier, or SectionID. The SectionID plays a critical role in report generation. When a value from a result set is processed for displaying in the report, the SectionID provides the context for the value. For example, if you have a formula field: count(SEC_01:TABLE1.COLUMN1)), it is the SectionID (SEC_01) which provides the context to locate and process the value. This will ensure that even
if you have any other reference to TABLE1.COLUMN1 in the template, it will not be confused with this instance of the data value in the result set. Your Report Designer supports both nested groupings as well as sibling groupings. A report can have as many sections as it needs.

About zones

If you are familiar with other Actuate e.Reporting Suite tools such as e.Report Designer or e.Report Designer Professional, zone is the equivalent of frame in those report designers. A zone is a container for controls. Controls include items such as report headings or footers, page headers or footers, and report details. Zones display in the report structure window. Zones format or hide data as described in the next section. Below are the zones of a report as they appear in the left pane of a blank report:

As indicated by the names in the pane above, there are seven possible zones in a report.

There is no requirement to define each of these zones in your report, nor do you have to display all the zones that you define. You can selectively hide or display zones.
About report headers

The report header appears only once, at the top of the report on the first page. The report header appears before the page header. Some data commonly used in the report header is:

- Company name
- Company logo
- Report title
- Date
- Report summary

The report header displays and prints on the first page only.

About page headers

The page header displays and prints on every page in a report. Some data commonly used in the page header is:

- Page number
- Page record count
- Page subtotal

About details

Details contains data extracted from the data source and values calculated from extracted values. The Details zone is the only zone that can span multiple pages and it consists of one or more sections. The report template defines height within a fixed range for other zones. To achieve the report layout you want, insert the following types of fields:

- Data
  
  Like reports generated using Actuate e.Report Designer Professional, reports created using e.Report Designer Java Edition can pull data from flat files, such as Excel spreadsheets or other stored reports, as well as from databases. Drawn from Table.Column in your data source, you usually place some data fields into Details.

- Formulas
  
  Perform calculations on extracted data. Formulas can range from simple ones, such as sum(), to complex statistical functions such as standard deviation and variance. For more information about formulas, see Chapter 8, “Working with formulas.”
Record count
Computes and displays the number of records of data accessed by the report.

Totals
Report summary.

Lines
Separates zones of the report.

Naming zones
Controlling the amount of detail displayed in a report is an important consideration in the design of large reports with multiple sections and multiple groupings. Drilling down in a report controls the amount of detail displayed. Users can choose to view report details, summaries, or both.

Giving zones meaningful names helps reveal the structure to the user. Like the sections that display in the table of contents, the zone names appear in the structure pane of the report explorer.

How to rename zones
1 Choose Design.
2 In the structure pane, choose the detail zone.
3 Choose Format ➤ Rename Zone.
4 Enter the new zone name in Rename.

Enter only printable, alphanumeric characters. Spaces are not supported.

About groups and nested groups
Grouping details into logical sets makes a report easier to understand and analyze. Groups either can appear at the same level or can be nested within each other. In addition, grouping records that share a common attribute supports calculating subtotals. In another example, grouping user names alphabetically improves usability of a report containing 500 or more user names.

You can nest groups within other groups. When viewed, groups start with a group header and end with a group footer.

For information about grouping, see “Grouping information” in Chapter 5, “Querying the database.”
About report footers
A report footer, which contains summary data, prints once at the end of the report. A report footer appears above the page footer of the last page of the report.

The report footer is the final part of the report generated after the rest of the report is generated. If e.Report Designer Java Edition detects that the report footer appears on a page by itself, the page header and page footer are suppressed.

About page footers
The page footer prints at the bottom of every page of the report including the last page of the report.

Hiding and showing portions of a report
As you design a report, you can hide parts of a report so you can concentrate on another part. As a report runs, you can control certain user access to segments of the report for purposes of security or confidentiality.

Hiding controls and zones
You can hide individual controls and zones.

How to hide a control
1 Choose Design.
2 Choose the control you want to hide.
3 Choose Format ➜ Make Invisible.
   The control is now hidden.

How to hide a zone
1 Choose Design.
2 In the report structure window, choose the zone you want to hide.
3 Choose Format ➜ Hide Zone.
   The zone is now hidden.
How to redisplay a control or zone

1. Choose the Design page.
2. Choose Format ➜ Make Visible ➜ Control ➜ label.
3. Choose Format ➜ Make Visible ➜ Zone ➜ Group Footer or the default name, GF-00.

Resizing zone height

The width of the zone is the width of the page. You can only change the height of a zone.

A common report layout problem is a details zone that is the wrong height, causing proportional spacing problems within the zone.

How to increase or decrease the height of a zone

To change the height of the zone, follow these steps:

1. Choose Design.
2. Choose the details zone to change zone height.
   
   Because the details zone displays the data of the report, it is normally one of the larger zones. In the example below our template has a page header zone much larger than needed and a details zone that cannot hold even one line of data.

3. Position the mouse over the bottom of the splitter line of RH until the cursor changes to a double arrow. Drag the splitter bar to the appropriate position that fits the page header without wasted space.

![Diagram showing how to resize a zone height]
If you cannot drag the splitter bar in the direction you want, the bar is probably hitting the bottom edge of a control. In this case, move the control and repeat the steps.

Once the controls are moved up in the header zone, you will find it easy to drag the splitter bar into place as in the illustration following.

Now your report header zone is proportionally spaced for your header.

**How to move controls**

In the preceding example, resizing the zone required moving the controls. Controls can be moved singly or in groups.

1. Choose a single control or choose multiple controls in a specific zone by holding down the Shift key and selecting the individual controls.

2. Drag the controls in the direction you want or choose Format ➤ AutoFormatWizard.

3. Choose controls using Format ➤ Multiple Selections ➤ Alignment ➤ Top Edges or one of the choices illustrated below.
In the instance of Top Edges, for example, the top edges of all selected controls align with the top edge of the last control you selected.

---

**Laying out a report**

You use zones to change borders, attributes and the size of parts of the report. During the layout of your report, you can do all of the following tasks in Design:

**Adding color**

Setting background color and creating borders around your report can improve the appearance and readability of it. You can use background and fill colors to distinguish certain sections of the reports.

**How to add color or set fill color**

1. Choose the appropriate zone in the report structure window.
2. Choose Format ➤ Zone Fill Color.
   
   Color Selection appears.
3 Choose Add.
   Select a color appears.
4 Choose a swatch from this page.
   The zone displays the selected color in the Designer page.
5 Choose OK.
   The preview of the zone appears in the color you selected.

Adding borders

Using borders to highlight important information, such as a subtotal can improve the appearance and readability of a report. You can add borders to a page, a zone, or a specific control. The border can have different styles, such as single, double, or dotted. You can apply color to the border.

How to add a border to a page
1 Choose the appropriate zone in the report structure window.
2 Select Format ➤ Zone Border.
   Border Style appears.
3 Choose Zone has a Border? if you want to apply a border to the zone. If no border is desired, remove the check from the box.

**How to choose border line style**

1 In Border Style, place your cursor on the line in the box next to Border Style.
2 Click on the line until the desired line style appears.
3 Choose OK.
   The border line style changes.

**How to choose border color**

1 You choose color in Border Style, when you select the box next to Border Color.
   Select a color appears with a color palette.
2 Select the desired color and choose OK.
3 Choose Design to view the border color.

Paginating a report

Typically, you use either the header or footer to number a page. Often, report designers do not want page numbers on the first page of a report. To inhibit the header and footer on the first page, you hide the page number by using the Not on First Page option.

How to suppress page numbers

1 Place your cursor in the page footer or PG_FTR section of the structure pane and right-click.
   The context menu displays.
The page number displays from the second page onward.

**Using database fields as page headings**

You cannot insert database fields into the e.Report Designer Java Edition page header zone. Instead, if you want to draw your headings from database fields, you use the group header zone, not the page header zone. You put the page header information plus the database fields in group header zones and use the Repeat on Every Page property.

**Repeating headings on subsequent pages**

When you design a multi-page report, columns of data often span more than one page. For clarity and user convenience, you need to be able to repeat the data that overlaps from one page on to subsequent pages. You repeat the data in the group header zone as a running header on the subsequent pages of the report.

If you have a formula control such as a page number in a header, you can reset the control data based on the zone. For information about resetting control data, see “Setting control properties” in Chapter 3, “Working with controls.”

**How to repeat group headers**

2. Choose Group Header.
3. Choose Format ➜ Repeat on Every Page.

**How to break a page**

At any point in the report, you can break a page.

1. Right-click to choose a header zone. For example, right-click the group header zone.
2. Choose Page Break Before.

A page break appears.
3 Right-click to choose a footer zone other than the report footer zone.

4 You can right-click the report footer zone and set it to print on a new page; otherwise, it appears on the last page of the report.

Using database field names as column headings

When you create a new report, your headings from database fields are placed in the group and page header zones. Once the column headings are in place, you can select the group header and select the Repeat on Every Page property from the group header context menu.

Creating a new report

You use e.Report Designer Java Edition wizards to create several kinds of reports easily and quickly. The wizards are especially helpful for new users. Use wizards to design most types of reports and perform complex tasks, such as defining parameter fields, defining query dependency for multiple-section reports, and defining charts and graphs. Wizards can be used to help you create the following types of reports:

- Blank Report
- Single Table Report
- Multiple Table Report
- Import Query Report
- Cross Tab Report
- Label Report

Another simple way to create a new report is to modify one of the existing designs we supply in the report templates.

How to create a new report


To create a new report, follow these steps:

1 Choose File ➤ New.

    New Confirmation appears.
2 Choose Yes.
New Report appears.

3 Choose Next.
Select a report style appears.
Select the default, Blank Report option. Choose Finish.

A blank report is now ready for your use.

Creating your own templates

To create your own templates, choose File ➤ New. A wizard helps you select the type of report you want to create. When you make a selection from any of the preceding choices, each one brings up a wizard for each report type that helps you create your report. The exception is Blank report.

How to load and save a report template

To load a template, use the following steps:

1. Choose File ➤ Open.

Load file from URL appears.
2 Type the name of the template you want to use.
If you already have a template loaded, you need to save any changes before loading the new template.

3 Choose Save to save the template currently loaded. If the template has been previously saved, the latest data is updated to the report designer, replacing the old template.

**How to use database field names as column headings**
During the report design process, you will be prompted to select fields from your selected database.

1 File ➔ New.
   New Report appears.

2 Select a Single Table report in Select a report style ➔ Finish.
   Create a simple report appears.

3 Choose Next.
   Choose Fields appears.
4 Select one table from the list in the left pane.
5 From the right pane, select one or more fields from the table. These fields become your column headings and appear in the group and page header zones.
6 Choose Next.
   Create a Simple Report appears.
7 Enter your report name and your footer information.
8 Choose Finish.

Column headings appear in both the group header zones.
Saving a report design

Use the File ➤ Save As function to save your template. If the file already exists, the new template will replace the old.

Creating batch files to launch reports

Batch files contain instructions to generate an e.report. The status of a request can be active, scheduled, or completed. A request requires sending an HTTP URL from the web browser. These requests typically have a format such as:

http://<webserver>/acweb/<reportserver>/path?command

A request for all reports for a particular individual would be:

http://<webserver>/acweb/<reportserver>/<folder>/?name

You create a batch file that resembles RunACJ.bat or ACJjdbc.bat and schedule the batch file to run at a specific time. The RunACJ.bat file contains the following command that invokes e.Report Designer Java Edition:

java -classpath \ACJRuntime.jar;\ACJDesigner.jar;\%CLASSPATH% ACJDemoApp -CSVSource.txt -Crosstab.acj

ACJjdbc.bat issues the following command to invoke the JDBC example:

java -classpath \ACJRuntime.jar;\ACJDesigner.jar;\%CLASSPATH% ACJMain [options...]

The options include database driver and user authentication. These options can be set using a wizard. For more information about working with databases, refer to Chapter 4, “Connecting to a database.”
Chapter 3, Working with controls

This chapter contains the following topics:

- About formatting options
- Setting control properties
- Inserting web links
- Managing controls
About formatting options

e. Report Designer Java Edition uses controls for design requirements such as setting fonts, colors, alignment, word wrap, and page breaks. In addition, e. Report Designer Java Edition uses control to create report styles including summary, tabular and columnar reports. The following types of controls are available:

- Labels
- Date
- Record Count
- Lines
- Rectangles
- Images
- RichText
- Web Links

For information about the following more complex controls, see “Setting control properties,” later in this chapter.

- Expressions and Formulas
- Page Number
- Charts and Graphs
- AsyncGrid
- Data Fields

Using formatting options

There are several options on the format menu that appear or not depending on the control you have chosen.

How to view formatting options for a control

1. Choose a label control.
   
   For example, select the label in the report header zone.

2. Choose Format.
To understand the differences in available properties for a specific control, choose another type of control. For example, choose the first control in the details zone, Products.ProductsID.

4 Choose Format.

Notice the differences in the two preceding menus. In one menu, there is a Label Caption option. In the other menu, there is no Label Caption option.
but there are Data Source and Conditional Formatting options. The available options depend on the zone or control that you select.

5 To close the menu, click somewhere on your report design.

**Using the formatting toolbar**

You can access frequently-used attributes from the toolbar:

- Font type
- Font size
- Foreground color
- Background color
- Left alignment
- Center alignment
- Right alignment

For example, to right justify a control, select the control and choose Right Alignment.

**How to view formatting options for a zone**

Selecting a control also activates the zone that contains it. To view the zone-level formatting options, select a group header zone and choose Format. The available formatting options display.

When you select a control, you are also activating the zone in which the control resides.
Setting control properties

To modify most properties of a control, choose the control. Common Properties appears.

Using the tabs, you can change the label identifier, the font and colors, and the style and alignment of the control.

Most properties such as colors, fonts, borders, text alignment, word wrap, bounds, and control ID can be modified using Common Properties. You also can modify common properties using the menu and toolbar. To use the icons, first select the control and then choose the property icon.

**Setting text and fill color of a control**

To set the text and fill color of a control, choose the control and then choose the text or fill color icon.
How to set the text and fill colors

1. Choose a label control.
2. Choose Format ➤ Font/Color.
   Select object attributes appears.
3. Use Object Attributes to set the colors of the control.

As you select various options, the design window at the bottom-right of the window changes to reflect your selection. Selecting foreground displays the text. Setting background/fill color colors the area behind the text.

Using Labels

You can add a label to any zone in a report design. A label is one of two types of fields that can be moved from one zone to another. You can also move lines.

How to insert a label

1. Choose Insert ➤ Label.
   The format menu appears.
2. Type the label text and choose OK.

Change Label’s Caption appears.
The label appears in the report.

**Modifying a label**
You can modify most properties of the label including width, height, font, and color.

**How to modify a label**
1. Choose a label.
2. Position the mouse on the edge of the label. Select the edge and drag it to shorten or lengthen.
3. With the label still chosen, right-click. Label format menu displays.

<table>
<thead>
<tr>
<th>Label Format Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify Label</td>
</tr>
<tr>
<td>Font/Colors...</td>
</tr>
<tr>
<td>Style/Alignment...</td>
</tr>
<tr>
<td>Delete Label</td>
</tr>
<tr>
<td>Resize to fit content</td>
</tr>
<tr>
<td>Suppress While Printing</td>
</tr>
<tr>
<td>Make Invisible</td>
</tr>
<tr>
<td>Send Behind Others</td>
</tr>
<tr>
<td>Bring In Front of Others</td>
</tr>
<tr>
<td>Copy to clipboard</td>
</tr>
</tbody>
</table>

Only fonts supported by Java are available. As you select options, the preview window displays your selection.

5 Choose font, font style, size, and effects.  
The following font effects are available: no underline, single underline, double underline, single accounting, double accounting and strikethrough.

6 Choose Foreground.  
Select a color appears.
7 Choose the color swatch for the foreground color.

8 To set the background color, choose Background. Select a color appears.

9 Choose your desired color from the color swatch. Choose OK. If the background color appears with a cross hatch in the preview window at the bottom-right, the background color of this field is transparent. If the color is transparent, the color of the zone, if it is set, or the default color is used as background.

10 To specify a background color, uncheck the Transparent option and check the rectangle across from the Background label.

**Dating reports**

You can insert today’s date anywhere in the report. Today’s date is stored in the following format:

```sql
format(now(), "MMM-DD-YYYY")
```

This format displays today’s date as:

April-06-2001
How to insert today's date
1. Choose Design.
2. Choose Insert ➤ Today’s Date.
3. As you move your cursor, you see the label Today’s Date. Drag the date where you want it and drop it.

How to change the format of today’s date
The date field is still selected.
1. Choose Insert ➤ Formula.
   Formula specification appears.
   MMM-dd-yyyy prints out with an abbreviated alpha version of the month, the day with a leading zero, and the complete year. MMMM-dd-yyyy produces a date format of long alpha month, day with leading zero, and full year as 2001.
2. Select the desired format.
3. Choose OK.

Using record counts
A record count is a formula that keeps track of how many records are present in a particular group or in the entire report.

Understanding reset criteria
When you insert a record count in the report, you must set the reset criteria to specify when to restart the count from zero. This count dictates when a
formula internal state needs to reset. Reset criteria are determined on the basis of the zones of the report and are set to the zone in which it is contained.

**How to reset criteria to display page numbers**

1. If you display page numbers, set the reset criteria of that formula to the Report.Footer.
2. If you have a formula in the page footer that sums the values of a certain field on that page only, set the reset criteria for that formula to the Page.Footer zone.

**Using reset criteria in a grouping**

If you have a grouping report, and you want to display the record count for each group, set the reset criteria to the Group Footer zone. If you want the total report count of all records in your report, set the reset criteria to the Report Footer zone.

**How to insert a record count**

1. Choose Insert ➤ Record Count.
2. Position the cursor to insert the record count in the detail zone.
3. Click to insert record count.

**How to reset the record count**

The record count appears in the page footer in a control box as:

\[
\text{this +1}
\]

1. To reset the record count, double-click the count field control. Common Properties appears.
2 Select the button to the right of the value expression. Formula specification displays.
3 Under Reset after zone, set the reset criteria of the record count to start the count from zero. You can also access Formula specification by selecting Format ➤ Expression.

For example, to get the total report count of all records in your report, select RPT_FTR from the Reset after Zone list.

4 Choose OK.

**Adding graphics**

You can insert graphic objects, such as a line or a rectangle, into a report. Lines can help emphasize various sections or group parts of a report. A common place to insert a line is a summary field.

Rectangles are often used in form reports to make a part of a report appear grouped. For example, you can use rectangles as a documentation convention in a form report to indicate that it is for official use only. You can create a rectangle in the background and overlay other controls on top of it.

You can format lines and rectangles. For example, when you add a line to a report, the default orientation is horizontal. You can make it vertical.

**How to insert lines and rectangles**

1 Choose Line or choose Insert ➤ Line.
2 Select the location where you want to insert a line.
   
   By default, a one-inch line displays.
3 Select the line and drag to lengthen or shorten.
4 Choose Format ➤ Make Vertical.
   
   The line rotates 90 degrees.
5 Choose Format ➤ Line Style/Attrs.
   
   Line Edge Attributes appears.
How to modify lines and rectangles
1. Choose edge color by clicking on it.
2. Select a color swatch and then choose OK.
3. Repeatedly select the edge style button until you see the line style you want and then choose OK.
   The line displays in the specified color and style.

How to create and fill a rectangle
1. Choose Rectangle or Insert ➔ Rectangle.
2. Choose the location where you want to insert a rectangle.
   A rectangle appears in your report.
3. Choose the rectangle and drag a handle to increase or decrease its size.
4. Choose Format ➔ Fill Color.
5. Select a color swatch and choose OK.
   The rectangle fills with the selected color.

Inserting images
You can insert .gif or .jpg images anywhere in the report. Inserting images into the details zone can use excessive amounts of memory because the image generates once for each row of the details zone. It would be more efficient to insert the image as a note and position it in the GF section of the report.

How to insert an image
2 Select the location where you want to insert the image.

3 To conserve memory, avoid using the details zone.

Load file from URL appears.

4 You can specify the image location in one of the following ways:
   - URL
   - Absolute path name. An absolute path name is the full path of the image file. It describes the image file directory, independent of the current working directory.
   - Relative path name. The relative path name is a path that is relative to the current working directory. For example, ../imageA.gif is the relative path of imageA.gif that is in the directory two levels above the current directory. You must set an option if you want to use relative path names. For more information about setting the relative file path names option, see “How to set the relative path names option” later in this chapter.

5 If you are publishing your reports on the Web, you must use a URL so your web server can access the image. Type the appropriate file name or URL.

**How to set the relative path names option**

If you want to use relative path names, you must set this option.

1 Choose Tools ➤ Options.

   Options appears.
2 Select Make Imagepath relative.
3 Choose Save and Apply.

**Formatting text**

You can design reports with enhanced text formatting using fonts, color, and alignment. You break text such as headings or titles into sections of text, applying special formatting to select portions. In order to accomplish this, you use the Rich Text Editor.

**How to apply and modify rich text formatting**

1 Choose Insert ➜ Rich Text.
2 Select an area of the report where you want to place a new control. Actuate Rich Text Editor appears.
3  Insert and format text, using the RichText Editor toolbar located at the top of the Editor.

4  When you finish formatting, choose Save Text.

5  Choose Close the Editor.

To modify an existing RichText control, choose the control.

**How to insert a RichText control**

You can embed data from other parts of your report into the RichText control. For example, you can create a salutation for a form letter and personalize it using names from your database or from an existing field in your report template.

Using the following tools, you can insert a control, a formula, or a column from your data source into a RichText control, assuming you are in a zone that accepts a control, formula, or column.

For example, in the group header zone, you can insert a control, formula, or column into a RichText control. In the report header zone, you can only insert a control.

**Working with variable control dimensions**

An important characteristic of a letter report is its dynamic nature. Text inserted into a RichText control may require the control’s dimensions to vary in size. If text is formatted extensively, the requirement for variable control dimensions increases.

Another factor that affects the size of the control is the data source used when you insert a column into a RichText control. When the data comes from your database or some other data source, you may not know the exact size required for the control. To accommodate data that changes in size, you set the Resize to Fit Content property.

**Resizing a control to fit**

Often, the data in a control is not a fixed size. For example, a text field in a database can sometimes occupy one line and other times occupy many lines. If you allocate a large enough size to display the maximum size of data in the field, you can display the largest strings easily. Using that technique means the smaller strings are displayed surrounded by too much white space.
Set the Resize to Fit Content property for a control. The height of the control dynamically adjusts to match the data.

**How to resize a control**

1. Insert a control, such as a Rich Text control.
2. Select the control.
3. Using your cursor, drag the frame around the control, until the text fits.

---

**Inserting web links**

You can insert URLs into a report and create links to web sites. You enter two strings, a label and URL link. The label displays in the report. When you click the label, the web browser opens and makes a connection to the URL associated with the label.

**How to insert web links**

1. Choose Insert ➤ Web link.
2. Select the location in the report to place the control.
   
   Web link settings appears.

3. In Web link caption, enter the text displayed in the report.
4. In the Web link URL, enter the URL.

   When e.Report Designer Java Edition detects a URL that contains a link to an HTML file, it invokes the browser. When it detects a link to a PDF file, it invokes the PDF reader.
Managing controls

This section discusses how to handle controls.

Aligning text

It may be necessary to align text to an edge of a control to make the report appear symmetrical or to make information easier to read. For example, currency values are, by convention, right-aligned, so that the decimal point or comma are aligned. In this case, you align controls that contain currency to the right edge of the control.

You can align text horizontally inside a control.

You can also specify the vertical orientation of the text alignment.

How to align text

1. Choose Design.
2. Select the control you want to align.
3. To align text to the left, center, or right, choose one of the following tools:

   ![Aligning Text Icons]

   Or choose Format ➤ Style/Alignment.

Attributes appears, offering additional options including top, bottom, horizontal and vertical centering.
Placing bounds on a control

You place bounds on a control by setting its width, height, and position. You can also drag for additional precision.

How to place bounds on a control

1 Select a control.
2 Choose Format ➤ Bounds.
   Control Bounds appears.

3 Enter values for the Left Coordinate, Top Coordinate, Width, and Height.
Positions are relative to the top left corner of the report whose coordinates are \((0,0)\). Values are either in inches or in centimeters, depending on the Tools ➤ Options setting.

**Identifying a control**

A key property of every control is its identifier, an alphanumeric string that uniquely distinguishes it from other controls in the report. Each control must have a unique control ID. That unique control ID cannot be blank. You use control IDs when you work with formulas.

**How to set the control ID**

1. Select a control by clicking on it.
2. Choose Format ➤ Control ID or right-click to choose the control. A label menu appears.
3. Choose Modify Label ➤ Control ID. Change control’s id appears.
4. Enter a unique ID. Do not use spaces.
5. Choose OK.

**Bringing information to the front or back**

You can have overlapping controls. The feature is useful when you want to have one label on top of another, for example.

**How to overlap controls**

1. Select the field that you want to send to the back or the front.
2. Right-click.
3. Choose Send Behind Others, or Bring In Front of Others.

**Suppressing duplicates**

Sometimes you have columns of data repeated in your report. For example, a columnar report contains an employee’s name and department. You want to display the employees by department with a department name appearing once, followed by the list of employees who belong to it. You do not want the department name repeated for each employee, so you suppress the duplicate department names. Use this feature only if your data is sorted based on the data in that control.
How to suppress duplicates

1. Select the control with data that you want to suppress, if that data is a duplicate of date in the previous row.

2. Choose Format ➔ Suppress Duplicates.

   If you do not find this menu option, try choosing another control to see if it appears. Remember the exercise earlier where different controls brought up different choices on the menu.

Dynamic formatting

You can change certain attributes of a control dynamically, while the report generates. For example, if your report contains a negative balance, you might want to display the balance in red. For more information about dynamic formatting, see Chapter 9, “Conditional formatting.”
This chapter contains the following topics:

- About using data sources
- Connecting to a data source
- Understanding Java database connectivity
- Connecting to an ASCII data source
- Connecting to a JDBC data source
- Database connection information for programmers
About using data sources

This chapter describes how to set up ODBC data sources and ASCII files for use with e.Report Designer Java Edition.

e.Report Designer Java Edition uses the Java Database Connectivity (JDBC) Open Database Connectivity (ODBC) bridge to connect to the ODBC data sources. e.Report Designer Java Edition reads ASCII files directly.

Use the Data and Tools menu of e.Report Designer Java Edition to access wizards to help you define your database connectivity.

The following list summarizes e.Report Designer Java Edition data source options:

- JDBC Data Source
- ASCII Data Source
- Application Data Source
- MetaData Source
- EJB Data Source

Connecting to a data source

These sections assume you have the following software installed:

- Java Runtime Environment (JRE)
- Java Database Connectivity (JDBC) - Open Database Connectivity (ODBC) bridge installed.

If you do not have Java Runtime Environment installed, see “Understanding Java database connectivity,” later in this chapter.

For the data source connection to work, you need an ODBC entry for the database. Use the ODBC manager to create that entry. The next section describes how to create an entry in the ODBC driver manager.

How to create an ODBC entry for your database

e.Report Designer Java Edition includes several sample reports or templates that use the readily available Microsoft Access product and its sample Northwind.mdb database. To work with those sample reports, you need to follow these instructions to connect to the Northwind.mdb database.

1. Open the Control Panel.
2. Double-click ODBC Data Sources.
ODBC Data Source Administrator appears.

3 Choose Add.
Create New Data Source appears.

4 Choose Microsoft Access Driver (*.mdb).

5 Choose Finish.
ODBC Microsoft Access Setup appears.

6 Enter Northwind for the Data Source Name. Enter Java e.Reports sample database for the Description. Choose Select.
7 In Select Database, navigate to Northwind.mdb on your hard drive. For example:
C:\Program Files\Microsoft Office\Office\Samples\Northwind.mdb

8 Select Northwind.mdb and choose OK.
Northwind appears in the User Data Sources list.

9 To close the ODBC Data Source Administrator, choose OK.
Connecting e.Report Designer Java Edition to the example database

Next you run ACJJjdbc.bat to establish a connection between e.Report Designer Java Edition and the Northwind database. Follow the steps in “How to establish a database connection,” later in this chapter.

Using ACJJjdbc.bat on a Windows platform

Run the ACJJjdbc.bat file to see an example of how to use e.Report Designer Java Edition for database reporting using a JDBC. By default, ACJJjdbc.bat script uses the JDBC-ODBC bridge to connect to the Northwind.mdb file. It also uses the template located in the templates\ForNorthwind directory.

How to run ACJJjdbc.bat on a Windows platform

1. Navigate to the ACJ50Swing directory.
2. Double-click ACJJjdbc.bat.
   
   A script runs in a command window.

   ![Command Window Image]

3. Press any key to continue.
   

4. To load ForNorthwind.jod, choose File ➤ Open.
   
   Load from URL appears.
5 Choose Browse.
   A list of folders appears.
6 In Look in, navigate to c:\ACJ50Swing\templates\ForNorthwind.
7 Choose ForNorthwind by double-clicking.
   Report formats display in the following window.

8 Select SimpleTemplate.jod and choose Open. Choose OK to close Load file from URL.
   ACJJjdbc For Northwind\SimpleTemplate appears.
Using ACJJjdbc.bat on a non-Windows platform

To run ACJJjdbc.bat on a platform other than Windows, you must modify the batch file to conform to that platform. Changes might include changing the command used to display text at the command line, changing location of executables, and changing delimiters separating directories.

How to run ACJJjdbc.BAT on a UNIX platform

To run ACJJjdbc.bat on a UNIX platform, make changes to batch file commands as necessary and use chmod 755 to change permissions so you can execute the file. Run this batch file from the command line as follows:

1. Open Command Prompt.
3. Start e.Report Designer Java Edition by entering the following command:
   `ACJjdbc.bat`

   The file ACJjdbc.sh also ships with e.Report Designer Java Edition. ACJjdbc.sh is a shell script version of ACJJjdbc.bat.
Establishing a database connection

After creating an entry for an ODBC data source on your system, start e.Report Designer Java Edition and connect to the data source.

How to establish a database connection

1. Choose Data ➤ Connect.

   Datasource Property Sheet appears. You can connect to any registered data source, application data source or a JDBC. The Detected Drivers list shows the registered JDBC drivers. To register a data source, see “How to create an ODBC entry for your database,” earlier in this chapter.

![Datasource Property Sheet]

2. Check that the Datasource Property Sheet contains the following information for connecting to the data source. If the DataSource Property Sheet does not contain this information, enter it and choose Apply.

   For example, to connect to a JDBC data source, check for the following information:
   - JDBC Driver

     If you are using the Microsoft JVM (part of Microsoft Internet Explorer 4.0 or part of the Java SDK), use the following driver specification:

     ```
     jdbc.drivers=com.ms.jdbc.odbc.JdbcOdbcDriver
     ```

     If you are using the Sun JVM or Netscape JVM (part of Netscape Navigator 4.4 or later), use the following driver specification:

     ```
     jdbc.drivers=sun.jdbc.odbc.JdbcOdbcDriver
     ```
- JDBC URL
  Use the following driver specification:
  jdbc:odbc:Northwind
  For more information about the JDBC URL, see “About the JDBC URL,” later in this chapter.
- Username for accessing the database
- Password for accessing the database

3 Choose Select Tables.
Select Tables appears.

4 Check that all the tables are selected as shown above. Choose Add Selected or Add All to choose tables to be included.

5 Choose OK ➔ Apply ➔ Close.

Testing the database connection

1 From the toolbar, choose Display Datasource Schema to test the connection.
The following schema appears if you use all tables in the Northwind database for the report.
The Database Browser displays Tables/Columns and organizes the data at the following levels:

- The first level displays table headings.
- The second level displays the columns within each table.

2 Choose Relationships to display the organization of tables and fields.
Understanding Java database connectivity

e.Report Designer Java Edition uses the JDBC™ (Java Database Connectivity) protocol to connect and communicate with databases. JDBC is a Java API that executes SQL statements. It consists of a set of classes and interfaces written in 100% Pure Java.

About the JDBC-ODBC bridge

ODBC protocol is the most commonly used standard for database access. Javasoft provides a JDBC-ODBC bridge to connect JDBC to a ODBC data source. The bridge implements JDBC for any database for which an ODBC driver is available. If you already have either the Java Developer's Kit or the Java Runtime Environment installed on your system, the JDBC-ODBC bridge is already installed.

Connecting to a database over the bridge

e.Report Designer Java Edition uses JDBC to send SQL statements to virtually any relational database. Each database vendor has a different method of integrating with the Java environment. You may need to contact your database vendor for the following information to configure e.Report Designer Java Edition to work with your database:

- JDBC Driver specification string
- JDBC URL that you would need to connect to the database

JDBC connects to a database driver, not the database itself, so the creator of a database specifies the JDBC URL that identifies their particular driver. Most often, the database vendor also provides the appropriate drivers.

About the JDBC URL

A JDBC URL identifies a database so that the appropriate driver recognizes it and establishes a connection with it. A standard JDBC URL has three parts separated by colons:

jdbc:<subprotocol>:<subname>

jdbc
The protocol.
The name of the driver or the name of a database connectivity mechanism supported by one or more drivers. For example, odbc, is reserved for URLs that specify ODBC data source names, which is a subprotocol.

A database identifier varies depending on the subprotocol. The driver writer chooses the internal syntax of the subname. The subname provides enough information to locate the database.

The following URL is an example for accessing a database through a JDBC-ODBC bridge:

```
jdbc:odbc:Northwind
```

In this example, jdbc is the protocol, odbc is the subprotocol, and Northwind is the subname. In this case, Northwind is a local ODBC data source.

**Making remote connections**

Connecting to databases on a remote machine requires entering additional connection parameters. For example, if you want to access a database over your company intranet, include the network address in the JDBC URL as part of the subname, following the standard URL naming convention:

```
//hostname:port/subsubname
```

For example, using a protocol, itnet, to connect to a machine on your company intranet, the JDBC URL looks like this:

```
jdbc:itnet://dbserver:879/sales
```

---

**Connecting to an ASCII data source**

An ASCII data source is a text data file. The text file must be either a comma separated value, CSV, file or a fixed length field file.

**Using the ASCII data source wizard**

You can connect to ASCII DataSource using a wizard.

1. Choose Data ➜ Import ASCII Data.
   
   Import ASCII data source wizard appears.
2 Choose Next.

Data appears.

3 Choose one of the following buttons as appropriate to specify the data source or Load Properties file.

- Specify ASCII source. The ASCII file is either comma separated data or data with a fixed length. Enter the name of the ASCII file data source or
use the Browse button to navigate to the ASCII file. Choose Open. Choose OK.

Load Properties file. The data in the ASCII source file includes table names, column names, delimiter characters or the individual fixed lengths of each column, the boolean truth values, and the date separator. Enter the name of the properties file or use the Browse button to navigate to the file. Choose Open. Choose OK.

4 Choose Next.

Depending on what you select, Tables appears. In some instances, e.Report takes you directly to Columns.

5 Use Tables to select and modify the table. You can also specify the delimiter, or, if the columns have fixed length, you specify the fixed lengths of each column.

6 To display Columns, choose Next.

Columns appears.
Use Columns to:
- Change default table names
- Specify whether table fields require extraction using a delimiter or fixed width
- Specify date format
- Specify true and false values for each boolean column in each table

7 Choose Next to access the Properties page.

8 Choose Finish.
Use Properties to display the table property information in the source file. You can save the ASCII data file properties in a file, so you can specify a properties file with the ASCII source data file properties. When you specify a properties file, you avoid entering information in the wizard when you connect to an ASCII data source. Descriptions of table fields follow.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tables</td>
<td>Number of tables.</td>
</tr>
<tr>
<td>Select table</td>
<td>Table names. Default table names are Table0, Table1, Table2, and so on.</td>
</tr>
<tr>
<td>Modify table name</td>
<td>Changes Table name.</td>
</tr>
<tr>
<td>Sample Data</td>
<td>Displays sample data for the selected table.</td>
</tr>
<tr>
<td>Are columns separated by Delimiter?</td>
<td>When you use delimiters, data is parsed correctly to the application.</td>
</tr>
<tr>
<td>Do columns have a fixed length?</td>
<td>Columns have a fixed length.</td>
</tr>
<tr>
<td>Specify delimiter/lengths</td>
<td>Enter the delimiter when columns are a fixed length. For example, specify length as 3,3,6 for a three columns table with widths of 3,3, and 6 characters. Specify the lengths of each table separately.</td>
</tr>
</tbody>
</table>

**Connecting to a JDBC data source**

Before you can use this wizard to connect to an ODBC database, the ODBC database must be registered on your system. For more information, see “How to create an ODBC entry for your database,” earlier in this chapter.

1. Choose Data ➔ Database Connection Wizard.
   Connecting to a database displays.
2. From the Data menu, choose JDBC data source.
   A check appears beside JDBC Data Source.
3. Choose Database Connection wizard.
   Connecting to a database appears.
The following table describes the JDBC Connectivity options.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver</td>
<td>Enter your database driver main classname. e.Report Designer Java Edition verifies that the class exists when you choose Next button. If e.Report Designer Java Edition cannot find the class, the problem is probably that your CLASSPATH does not include the driver class or you have input an incorrect string for your driver class. To update your CLASSPATH, exit e.Report Designer Java Edition. Go to your control panel. Choose System ➤ Environment. Modify the CLASSPATH to include your driver class, and restart e.Report Designer Java Edition. Contact your database administrator or your driver vendor if you do not have the correct driver.</td>
</tr>
<tr>
<td>URL</td>
<td>Enter JDBC database URL, username, and password. Before you can move to the next page, e.Report Designer Java Edition connects to your database to validate the information you supplied.</td>
</tr>
<tr>
<td>Login</td>
<td>Enter a username, and password if necessary.</td>
</tr>
</tbody>
</table>
Database connection information for programmers

You can create Windows batch files or UNIX shell scripts to start e.Report Designer Java Edition using parameters to specify options.

The Windows batch file RunACJ.bat issues the following command to start e.Report Designer Java Edition.

```
java -verify -classpath .;./ACJRuntime.jar;./ACJDesigner.jar;.
ACJDemoApp.jar;./capsapi_classes.zip;%CLASSPATH% ACJDemoApp demo
ACJDemoApp\CSVSource.txt templates\ForACJDemoApp\ChartTemplate.jod
```

ACJjdbc.BAT issues the following command to invoke the Java e.Reports and the JDBC example:

```
java -verify -mx64m -ms64m -classpath .;./ACJRuntime.jar;./ACJDesigner.jar;.
ACJMain.jar;./capsapi_classes.zip;%CLASSPATH% ACJMain -
ShowDesignerTRUE -IsIndexedFALSE -OutputDevicePREVIEW -
OutputFileOutputSample.pdf -jdbcDRIVERsun.jdbc.odbc.JdbcOdbcDriver -
jdbcURLjdbc:odbc:Northwind -jdbcUSERNAMEadmin -
jdbcPASSWORDpassword -TemplateFileTemplates\ForNorthwind\ ChartTemplate.jod
```

Running ACJMain to set parameters

e.Report Designer Java Edition always needs a wrapper class to run. The ACJMain.jar wrapper application is in the demo/ACJMain subdirectory that is included in your software.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>Specifies tables to use in the report: either a category of tables (Table, View, System, Alias, Synonym) or specific tables. Use this setting to retrieve a large set of tables by specifying just one category name. You can also selectively retrieve tables hiding those with sensitive data.</td>
</tr>
<tr>
<td>Status</td>
<td>Displays the status of the connection you just established. Also displays information about your driver and database.</td>
</tr>
</tbody>
</table>
Use the ACJMain wrapper application to configure the JDBC driver, URL, user name, password, and the other configuration parameters, described in the following table.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-ShowDESIGNER</td>
<td>Specifies whether or not the template designer window displays. Values: true, false. Default: true.</td>
</tr>
<tr>
<td>-TemplateFILE</td>
<td>Specifies template file used. Values: path and file name, URL.</td>
</tr>
<tr>
<td>-OutputFILE</td>
<td>Specifies output file used. Most often used when e.Report Designer Java Edition is in a non-interactive mode. For example, if output device is HTML, use this parameter to specify the name of the output file.</td>
</tr>
<tr>
<td>-JdbcDRIVER</td>
<td>Specifies the JDBC driver. For example, jdbc.thin.oracle.jdbcDriver</td>
</tr>
<tr>
<td>-JdbcURL</td>
<td>Specifies the JDBC URL of the database to which e.Report Designer Java Edition should connect. For example, jdbc:odbc:Northwind</td>
</tr>
<tr>
<td>-JdbcUSERNAME</td>
<td>Specifies the user name used to connect to the database.</td>
</tr>
<tr>
<td>-JdbcPASSWORD</td>
<td>Specifies the password used to connect to the database.</td>
</tr>
<tr>
<td>-Language</td>
<td>Specifies default language. English is the only language currently supported.</td>
</tr>
<tr>
<td>-Indexed</td>
<td>Specifies whether or not the report shows the Table of Contents. Values: True, False. Default is True.</td>
</tr>
</tbody>
</table>

**Configuring database connections for JDBC data reporting**

Running ACJjdbc.bat executes a script that sets up the JDBC connection to the Northwind database. You can then use data reporting examples using the Northwind database without further configuration. Follow instructions in this section to use a database connection wizard for any of the following reasons:

- You want to connect to a database without running ACJjdbc.bat.
- You want to connect to a database other than Northwind.
You want to configure Java e.Reports to use a sub-protocol other than ODBC.

If you want to report from data in Java objects, see “Configuring the application data source,” later in this chapter.

**Defining a database using code**

e.Report Designer Java Edition supplies a wizard to define your database connectivity. Information from the wizard generates Java code you can cut and paste into your source file. After cutting and pasting, you must finish several methods before you can use e.Report Designer Java Edition for application reporting. Follow the documentation in the source code that is created. Implement the necessary methods, compile the code, and run your application again.

**Configuring the application data source**

Using Connecting Application data saves you from using the API to define objects when you want to report from data in Java objects.

Application data reporting requires defining data, passing data, and generating the report. Like the JDBC Connectivity wizard, Connecting Application data generates code that the you can use for to code other applications in java.

**How to configure the application data source**

1. Choose Tools ➤ Code Generator wizards ➤ Application data.

Connecting application data appears.

---

**Connecting application data**

Providing powerful output generation features for your existing Java application

The wizard generates Java code you can use to configure your application as a data source for application reporting with e.Report Designer - Java Edition.

The wizard steps you through the process of defining your new application objects for use with e.Report Designer - Java Edition, and uses the information to generate code you can use in your application.

**NOTE:** Before you can specify any classes as data sources, (they must be present in your CLASSPATH environment variable).

The code generated by the wizard is not complete. The code contains instructions to help you complete the code. After you have finished the code and set the code to your application, you can compile and run your application and use e.Report Designer - Java Edition with your application.

---

82 Designing e.Reports Java Edition
2 Choose Next.
Tables appears.

3 Choose the Browser to locate a class file.
Load file from URL appears.

4 Choose Browse.

5 Navigate to demo\ADJDemoApp\Order.class.
Load file from URL appears with pathname for the file you selected.

6 Choose OK.
If you get an error message, see “How to set classpath” or “How to register a table” later in this chapter. The class file must be in your classpath to run.

7 If there are no changes to tables, choose Next.
Columns appears.
If you have no additions or deletions, choose Next.

How to set classpath

If you get an error message, you may need to compile your Java files into class files. See “How to register a table” for more information. If compiling does not resolve the error, set classpath in the Command Prompt.

1. >Cd acj50swing\demo\acjdemoapp.
2. >Set %classpath%.
3. >Runacjbat.
   RunACJbat brings up a new report.
   New Confirmation appears, asking if you want to save the old report.
5. Choose No.
   New Report Wizard appears.
6. Choose Next.
7. Select Blank Report and choose Finish.
Application Data Connectivity Options

The following table describes the Application Data Connectivity options.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>Map Java objects into tables used to generate reports.</td>
</tr>
<tr>
<td></td>
<td>Specify a table name and a corresponding class. For more information, see &quot;How to register a table,&quot; later in this chapter.</td>
</tr>
<tr>
<td>Columns</td>
<td>Display all registered Tables under Table names.</td>
</tr>
<tr>
<td></td>
<td>Choosing a table displays all registered columns of that table. Rename a column or add new ones. If a registered column is a class method and it accepts a string constant parameter, you need to specify the string constant on this page. You can also add comments for each of the registered columns.</td>
</tr>
<tr>
<td>DataSet</td>
<td>Displays information about how to pass your objects to e.Report Designer Java Edition. Saves work to a file.</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>Enter parameters for a wrapper application.</td>
</tr>
<tr>
<td></td>
<td>- Load a template file at deployment time and its name.</td>
</tr>
<tr>
<td></td>
<td>- Modify the template before running it.</td>
</tr>
<tr>
<td></td>
<td>- The output device you use at deployment time: Preview, PDF, HTML, CSV, DAT, Printer.</td>
</tr>
<tr>
<td>Code</td>
<td>Displays source code based on the information you entered.</td>
</tr>
</tbody>
</table>

Code can be cut and pasted to an existing file or saved to a new file. Then code can be modified to complete the integration between e.Report Designer Java Edition and your application. Comments in the code describe which parts you must complete.

How to register a table

1  Compile the existing Java application into byte code class files.

2  Select the class files from which data is to be extracted.

   e.Report Designer Java Edition detects the fully qualified class name from that class and generates a table name corresponding to the selected class name. You can change this table name.

3  Choose Verify Class and Add Table.
This step attempts to instantiate a 0-args constructor in your class file and registers your Class object as a table. It is essential to have a 0-arg constructor in your class files that represents tables. If your attempt succeeds, your table is registered and a list of public methods and fields display.

4 Select the methods or fields to extract for column data using reflection. e.Report Designer Java Edition supports methods with zero or one parameter only. All other methods are unavailable.

5 Register your next table by choosing browse and choosing the next class file, table object, to register.

6 Continue selecting tables and columns accessors.

7 Choose Next to describe each column in greater detail.

**Configuring the JDBC data source**

Using the JDBC Code wizard instead of the API to define objects when you want to report data in Java objects.

Application data reporting requires defining data, passing data, and generating the report. The Application Data Source wizard generates code that you can cut and paste.

To open JDBC Code wizard, choose Tools ➤ Code Generator wizards ➤ JDBC data.

The following table describes the JDBC Code options.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDBC connection</td>
<td>Enter JDBC database connection information such as the driver, URL, username, and password. Before you can move to the next page, the e.Report Designer Java Edition connects to your database to validate the information.</td>
</tr>
<tr>
<td>Tables</td>
<td>Specify tables to use in the report, either a category of tables (Table, View, System, Alias, Synonym), or specific tables. Use this setting to retrieve a large set of tables by specifying just one category name, or to selectively retrieve tables, hiding those with sensitive data.</td>
</tr>
</tbody>
</table>
Take time to read and understand the code. You can either cut and paste the code into an existing file, or save it to a new file. After you have saved the code, modify a few methods to complete the integration between e.Report Designer Java Edition and your application. Comments describe which parts you must complete.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miscellaneous</td>
<td>Enter parameters for a wrapper application.</td>
</tr>
<tr>
<td></td>
<td>• Load a template file at deployment time and its name.</td>
</tr>
<tr>
<td></td>
<td>• Modify the template before running it.</td>
</tr>
<tr>
<td></td>
<td>• The output device you use at deployment time:</td>
</tr>
<tr>
<td></td>
<td>Preview, PDF, HTML, CSV, DAT, Printer.</td>
</tr>
<tr>
<td>Code</td>
<td>Displays code based on the information you entered.</td>
</tr>
</tbody>
</table>
Chapter 5

Querying the database

This chapter contains the following topics:

- About data reporting capabilities
- Overview of the query process
- Uniquely identifying data
- Understanding section queries
- Creating SELECT and FROM clauses
- Creating WHERE and ORDER BY clauses
- Creating an import query section
- Defining joins between tables
- Defining filter conditions
- Defining filter conditions
- Sorting query results
- Grouping information
- Using multiple-sections with multiple groups
- Creating master-detail section reports
- About nested sections
- Understanding three-part SQL
- Understanding a MetaData data source
- Working with parameter fields
- Working with stored procedures or functions
- Understanding requirements
About data reporting capabilities

e.Report Designer Java Edition is a powerful data-reporting tool that extracts critical business information from multiple data sources and transforms it into reports for the web. As discussed in Chapter 4, “Connecting to a database,” e.Report Designer Java Edition generates reports from data sources such as:

- JDBC and ODBC databases
- Java applications
- Text files such as comma separated values (CSV) files or text files with fixed length fields
- Enterprise JavaBeans (EJB) enabled application servers

Overview of the query process

For each generated report, e.Report Designer Java Edition executes a query to extract data from the data source. You extract data from the data source in the report Details zone. Each report section within that zone has its own SQL query to extract data.

Format of queries

The format for database queries is a standard SQL query. These queries have the following format:

```
SELECT (...) FROM (...) WHERE (...) ORDER BY (...)
```

The format for application data source queries is a SQL-like query. This type of query follows the SQL specifications, but it does not implement them completely. These queries differ according to how the interface is defined.

Opening a report and viewing a SQL query

To view a SQL query, based on the latest state of a report, open the report.

1. From Design, choose File ➤ Open.

   Save Confirmation appears.
2 Choose No.
Load File from URL appears.

![Load file from URL dialog](image)

3 Choose Browse.
The available selection of templates appears.

4 Navigate to the ForNorthwind directory.

![Open dialog](image)

5 Select singlegouping.jod from the list of templates.

6 Choose Open.
Specify a URL for loading a template file: appears.

7 Choose OK.
Single grouping template appears in your report editor.
Choose Tools ➔ Section SQL Query ➔ (Section) SEC_00.

Section’s Query appears.

Section’s Query contains the various elements of a SQL query: the section to which the query belongs, as well as the fields, tables and conditions the query uses.
9 Choose Cancel to close without saving changes.
When you drag and drop a data field to the report, the SQL query is automatically updated.

Inserting a data field

You can insert columns from tables of a database into the Details zone of a report. Before inserting a data field into a report, you must confirm you have a connection to a data source.

How to insert a data field

1 By choosing Data ➤ Connect, you can confirm whether or not you have a connection to a data source.
If there is a driver in detected driver and a URL in Database URL, you are connected to a data source.

2 There are several ways to insert data fields into your report:
- You can drag and drop any field or table you choose on to the report design from the Relationships page of the Data Manager.
- You can drag and drop a field from one part of a report to another.
- You can choose Insert ➤ Data Field. The cursor changes to support dragging and dropping and a field control appears into the report. You place it in your report.

3 Choose Data ➤ Data Source Browser.
Data Manager displays the data source structure.
The data source levels include:

- Tables
- Columns within tables

4 Close Data Browser.

5 Choose Insert ➜ Data Field. Your cursor changes to support drag and drop and a Field control attaches to your cursor.

6 Place the control in the Details zone.

Select Field displays.

7 Choose Customers ➜ ContactTitle. Choose OK.

8 Choose Preview tab to display the new data in the report.
The contact title appears below the customer name.

Uniquely identifying data

The SectionID plays a critical role in report generation. When e.Report Designer Java Edition processes a result set value for display, the SectionID provides the context for the value. It is an alphanumeric string with no spaces, special, or non-printable characters. e.Report Designer Java Edition supports more than one section in a report, so each section must have a unique SectionID.

For example, you create a report that contains a SectionID (SEC_01) that e.Report Designer Java Edition uses to locate and process the value of the following formula control:

\[
\text{count(SEC\_01:TABLE1.COLUMN1)}
\]

The unique SectionID ensures that even if you have another reference to TABLE1.COLUMN1 in the template, it does not confuse the two data values.

Understanding section queries

For each section in the report template, e.Report Designer Java Edition runs a query to extract data from the data source. In most cases, e.Report Designer Java Edition generates the SQL query automatically, based on the fields in
your template, grouping criteria, and other factors. You can also import SQL queries generated in an external application.

Insert ➤ Report Parameters accesses the Parameter Field Definition Wizard to support adding parameters for section and imported queries.

**Modifying the section query or its ID**

1. Open the SingleGrouping.jod.

2. Before you use e.Report Designer Java Edition to create any queries, choose Tools ➤ Section SQL Query to display a menu with only one option, the generic query, SEC_00.

   After you define queries in a template, the Tools ➤ Section SQL Query menu lists them.

3. You choose one of the new section identifiers rather than the generic SEC_00.

   The following example shows multiple-sections in AdvMultiSections.jod. You can edit section queries directly from the Relationships page and watch how the WHERE CLAUSE generation in the Section query changes.

---

### Creating SELECT and FROM clauses

You can modify the section query and the SectionID by choosing Tools ➤ Section SQL Query ➤ SectionID. You can save the modified relationships to a metadata file for review later. You can also extract the relationships from the
data source by choosing Extract menu. This process is slow and requires a capable JDBC driver for proper operation.

e.Report Designer Java Edition constructs query clauses based on grouping criteria in the Details zone and the items you place in the Design:

- **SELECT**
- **FROM**

For example, suppose you have a table called Categories and a column within that table called CategoryName. If you place the Categories.CategoryName field in the Details zone of your template, e.Report Designer Java Edition generates the following query:

```
SELECT CATEGORIES.CATEGORYNAME FROM CATEGORIES
```

To ensure that the controls in the template process correctly and the required tables and columns are extracted from the data source, the e.Report Designer Java Edition updates the SELECT and FROM clauses based on the fields you add to the template. You cannot modify the generated SELECT and the FROM clauses.
Creating WHERE and ORDER BY clauses

You set an option to specify whether or not you want e.Report Designer Java Edition to construct the following clauses for you:

- WHERE
- ORDER BY

If these clauses are not generated, you must enter them in the section query. The auto-detection of joins and auto-creation of the WHERE clause simplifies the task.

Specifying the WHERE clause, especially for complex queries such as those generated for crosstab reports, can be difficult.

Generating the WHERE clause

e.Report Designer Java Edition can generate WHERE and ORDER BY clauses. The e.Report Designer Java Edition generates the WHERE part of the query only if it has the necessary primary foreign key relational data to detect the joins. Define these relationships by choosing Data ► Table Relationships. For more information, see “Defining relationships among columns,” later in this chapter.

If you do not specify table relationships, e.Report Designer Java Edition tries to detect the joins by matching similar names. For example, if you have a table called CATEGORY and a table called PRODUCTS and if each of them has a column called CATEGORYNAME, e.Report Designer Java Edition creates a relationship between CATEGORY.CATEGORYNAME and PRODUCT.CATEGORYNAME.

How to generate the WHERE and ORDER BY clauses

To enable or disable the e.Report Designer Java Edition automatic generation of WHERE and ORDER BY clauses, use the auto update options, as described in the following procedure:

1. Choose Tools ► Options.
2. Choose the Query tab.
3. Disable the Auto update WHERE feature.
4. Enable the Auto update ORDER BY feature.
Creating an import query section

To import a SQL query into a report, you can create an import query section. When you create an import query section, the e.Report Designer Java Edition creates a section query where you can enter your query text.

How to create an import query section

Using the example ForNorthwind\SingleGrouping.jod, create a nested import query section that contains a list of customers and for each customer a list of orders based on the following SQL query:

```
SELECT ORDERS.ORDERID, CUSTOMERS.CUSTOMERID, ORDERS.ORDERDATE
FROM ORDERS, CUSTOMERS
WHERE ORDERS.CUSTOMERID = '@FIE000@' AND
    ORDERS.ORDERDATE BETWEEN '#1-31-1994#' AND '#9-1-1994#'
```

In the query, the reference to a field @FIE000@ references the customer ID in the query section containing the nested section.

1. Navigate to ACJ50Swing\templates\ForNorthwind\SingleGrouping.jod. Choose Open.
2. Right-click the DTL_00 section to display the context menu.
3. Select the Insert IMPORT QUERY Section (Nested).
e. Report Designer Java Edition inserts an import query section in the report and displays Custom Section Query.

4. Type the query listed above and configure the section to display the text, No orders, if the nested section does not contain any data.

If the query is in a text file, use Load to choose the text file and add the query. Save sends the query text to a text file.

5. Choose OK to save the settings.
The e.Report Designer Java Edition displays Select Fields. Fields listed result from the SELECT clause of the SQL query.

6 Select the ORDERID and choose Add to add the ORDERID field to the report section.

7 Choose done to save the selection.

The report has a nested section with the order.

8 Preview the report.

The report contains a nested section with some sections containing order numbers.
Defining joins between tables

If your report confines all its data to a single table, the query is often quite simple. When you have data from multiple tables, use joins and links to create a query correctly. A join is a link among columns of various tables. e.Report Designer Java Edition detects the joins required in your reports and updates your query.

The next example extends the following query:

```
SELECT ORDERS.ORDERID FROM ORDERS
```

Assume there is another table in the data source called Orders that has fields called ORDERID and CUSTOMERID. To create a report that lists each Customer and the customer's OrderID, place the ORDERS.ORDERID field in the Details zone of the report. When you run the report, a union of both tables results. If you select Automatic update WHERE enabled, e.Report Designer Java Edition automatically uses relational joins to specify the database to which the tables are related.

If a report includes data from more than one table, you must join all tables from which you need to pull data. You can join tables only if they have primary and foreign keys. A key column in a database table is always unique so that each record in that table may be identified uniquely.

Using SingleGrouping.jod there is a field in the Orders table called CustomerID and a field in the Customers table that is also called CustomerID. The CustomerID field primary key in the Customer table and a foreign key in the Orders table.
How to join tables

You still have the SingleGrouping.jod open.

1. Choose Tools ➤ Section SQL Query ➤ (Section) SEC_00. Section’s Query appears.

2. Add the following string to the Where:
   CUSTOMERS.CUSTOMERID = ORDERS.CUSTOMERID

3. Using the section query to modify the WHERE clause, creates a primary key to foreign key relationship between the CustomerID fields.
   Do not enter WHERE into the section query. The e.Report Designer Java Edition generates WHERE.

4. Choose OK to update the query.

5. Select Preview to preview the report.
   e.Report Designer Java Edition executes the following query:

   SELECT
   CUSTOMERS.CONTACTTITLE, CUSTOMERSPHONE,
   CUSTOMERS.COMPANYNAME, CUSTOMERSCONTACTNAME,
   CUSTOMERS.COUNTRY, CUSTOMERS.CUSTOMERID
FROM
CUSTOMERS, ORDERS
WHERE
CUSTOMERS.CUSTOMERID = ORDERS.CUSTOMERID
ORDER BY
CUSTOMERS.COUNTRY

e.Report Designer Java Edition updates the From clause to include the Orders table.

Defining relationships among columns

You create and change links among columns of tables using the Relationships page of the Data Manager window to define relationships between primary and foreign keys.

Using the Relationships page to remove joins

1. To access the Relationships page, choose Data ➤ Table Relationships.

   The Relationships page displays the relationships among the available domain of tables used in designing the sample report design. The Data Browser Relationships page is a quick reference to the data source schema and the manner in which the data is organized.
The following example shows the default relationships defined by e.Report Designer Java Edition in SingleGrouping.jod.

2 Choose Add/Remove Relationships on the Relationships page in the Data Manager and use Create Relations.

3 Select Remove. Choose OK.
You can select and remove a single join or remove all joins.

**How to create a manual join using Relations**
You can manually create relationships by dragging a field from one table and dropping it into a field in another table. To create joins manually, use one of the following methods:
1 Choose the primary key column and drag it over the foreign key column. This procedure creates an internal relationship between the two tables using the two fields.

2 The following example drags the primary key CustomerID from the Customers table to the CustomerID a foreign key field in the Orders table.

How to create a join using Create Relation application

1 Choose Relationships. The Relationships page appears.

2 Choose Add/Remove Relationships. Create Relation appears.
3. Select the table and field for the source, the primary key and the destination, the foreign key.


---

**Defining filter conditions**

For many reports, using joins does not control the data sufficiently. Having created the necessary joins, you can choose to filter the report data further by adding conditions. Extending the previous example, assume you want to view only orders from Sales Agents. To add this condition, modify the WHERE clause in the query to:

```
CUSTOMERS.CUSTOMERID = ORDERS.CUSTOMERID AND
CUSTOMERS.CONTACTTITLE = 'Sales Agent'
```

Add more filtering criteria using the AND SQL keyword. Use the OR SQL keyword to define either or criteria.

To delimit Sales Agent use a single quotation mark. You can use other delimiters if your database supports the character to be used in string or date fields. Consult your database documentation.

The report user can also define filtering criteria. For example, instead of specifying Beverages, the report user can choose a different category to filter. Choosing categories is using parameters. For more information about using parameters, see “Working with parameter fields,” later in this chapter.

---

**Sorting query results**

Use the ORDER BY clause in the SQL Query to specify the sort order of the query results. In the ORDER BY clause, you use only the fields that you have used in the SELECT clause of the query. For example, in the following query, the ORDER BY clause can sort only by table.col1 or table2.col2:

```
SELECT table1.col1, table2.col2 FROM table1, table2 WHERE table1.col5 = table2.col9 ORDER BY (table1.col1 ASC or table2.col2 DESC)
```

Sorting occurs in either an ascending or descending order. To specify ascending sort, use the ASC keyword right after the table.column string. To specify descending sort, use the DESC keyword immediately following the table.column string. Sort order for each table.column can be defined individually.
Grouping information

Large reports can be hard to analyze. Grouping makes large reports easier to read and understand. A group is a set of records that satisfy given criteria. For example, if you are creating a report consisting of all your customers, grouping the output alphabetically is typical. Grouping uses one of the following bases:

- A column of a table from the data source
- A formula

Requirements for creating grouped reports

To create grouped reports you must set both of the following criteria.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort</td>
<td>Use the ORDER BY clause of your section SQL query to set.</td>
</tr>
<tr>
<td>Grouping</td>
<td>In most cases, you must also set the table.column, which you use to define the grouping criteria, in the ORDER BY clause.</td>
</tr>
</tbody>
</table>

For each value of the grouping criteria, the underlying data is grouped. For example, assume you have a table called Categories and a field called CategoryName. If you define your grouping criteria to be Categories.CategoryName, for each value that is found in the data source for the Categories.CategoryName, e.Report Designer Java Edition groups and displays the data separately.

How to group data

1. Right-click the Details zone or (DTL_00) to display the context menu.
2 Choose Insert ➔ Insert GroupingAbove.

A new Group Header zone (GH_01) appears just above the Details zone and a Group Footer zone (GF_02) appears just below the Details zone.
3 Select group heading GH_01. Right click on GH_01 zone, to display the context menu.

4 Choose Grouping Criteria → By column to display Group by COLUMN. Group by COLUMN appears.

5 Select the customers table and customerId field. Choose OK to set the grouping.

6 Choose Preview to view the updated report.
   The report data is grouped by Country and then by Customer ID.
7 To create additional levels of groupings, follow the steps above to insert groupings into the detail zone.

8 Select an existing group header and choose Insert ➔ Insert Grouping Above. You can create an unlimited number of nested grouping levels. Grouping criteria for each group must be set or unpredictable results can occur. For more information on this see the sections on grouping later in this chapter.
Setting grouping by formula

Setting groupings by formula has specific requirements:

- Refer to column names directly from inside the formula by means of the expression sectionID:TABLE.COLUMN function.
- Precede any reference to a TABLE.COLUMN in an expression by the sectionID because e.Report Designer Java Edition supports multiple-sections inside a report.

Grouping criteria are often set in alphabetical order. For example, you want to group your report alphabetically by values of a category. You would use the following formula in your grouping criteria:

```
substring("TABLE.COLUMN",0,1))
```

The string() is a function available in e.Report Designer Java Edition. TABLE.COLUMN is a generic name that represents any arbitrary column in the data source. For more information about using formulas, see Chapter 8, “Working with formulas.”

Nested groupings

You can nest multiple levels of grouping. Each group must have its own grouping criteria set. The grouping criteria can be set by formula or by column.

The following example describes the type of information that is ideal for nested groupings. Assume you have a database of 2000 clients across 50 states in the United States. A tabular report with 2000 names is unacceptable. Instead, the report should be designed to display data on a state-by-state basis. Within each state, the report should alphabetize the names.

How to set grouping criteria for nested groupings

1. Open templates\Northwind\SimpleMultiSections.jod.
2. In Design, select GH_00 in the structure pane to set the grouping criteria.
4. Choose By Formula or By Column.

See “How to group data” earlier in this chapter for an example of a report with nested groupings.
Using multiple-sections with multiple groups

A single section consisting of multiple levels of grouping is sufficient for most reports, but report design requirements or the design of the database schema can require using multiple-sections with multiple groups. A report can have an unlimited number of sections. Those sections can either be nested within each other or they can be siblings. Each section has its own query giving you flexibility in extracting data and formatting reports. Multiple-section reports are especially helpful when you need to display several series of summary data.

If multiple-sections are present in the report design and these sections are all at the same structural level, they are called siblings. eReport Designer Java Edition then prints all data for each section in the order the sibling sections occur in the report design.

Creating master-detail section reports

Multiple-sections generate reports that represent data in a hierarchical format. The data in such a report has a master-detail relationship. The key to creating a master-detail report is to link the queries of the detail section to some control of the parent section. Then, for each value of the control in the parent section, the child section uses that value to execute its query.

Using the Master Detail Query wizard

Use the Master Detail Query Wizard to design reports with defined dependencies between a master and a details section. For example, if you have two tables, one containing employee details and another containing the employee payroll history, you can create a report to display general information about the employee in the master section and payroll information for each employee in the detail section. In such a case, the query in the detail section has a dependency on the query in the master section.
Invoking the Master Detail Query wizard

You must invoke the Master Detail Query Wizard from a report that has one or more nested sections. To see an example of a Master Detail report that meets these requirements, see “Examining a multiple-section report” later in this chapter.

How to create a Master and Details section


2. Choose Next. Select a Master and Detail Section appears.

3. Select a section in the Master Section in the left pane.

4. Select a section in the Detail Section in the right pane. Choose Next. Columns appears.
5 From the list of Master Section Controls in the left pane, select a control. From the list of Detail Section Fields in the right pane, select field. Choose Link.

6 Use the lower area to change the type of data to be linked. Choose Next.

7 Specify delimiters used by your database. Check with your system administrator for this information. If you are using application data reporting, use the default values provided. Choose Next.

The Where Clause for the section displays.

8 Confirm that the displayed string is correct.

   Report Designer Java Edition appends this string to your existing WHERE clause in the Detail Section. The keyword AND or OR is added, if you already have an existing clause. If there is no existing clause, the string is simply added.

9 Choose Finish.

   Preview appears.
Examining a multiple-section report

1. Choose File ➤ Open.
2. Save Confirmation appears.
3. Choose No.
4. Load file from URL appears.

Choose Browse.

5. Navigate to templates\ForNorthwind.

Open appears providing a list of possible Java e.report design templates you can choose. All of these options use the Northwind database.


Load file from URL appears with the following URL:
C:\espressoswing\templates\ForNorthwind\AdvMultiSections.jod

7. Choose OK.

The report design template, Employee Sales Report (Using MS Access Northwind) appears.
About nested sections

In a report with a nested section, the query executed for the nested section usually contains a reference to a field in the master section. For example, for each ID in a master EMPLOYEES table, a list of achievements has been recorded in an ACHIEVEMENTS table. There exists a ONE-TO-MANY relationship among the tables. A column name called EMPLOYEEID is a primary key in the EMPLOYEES table and a foreign key in the ORDERS table.

Suppose you want to create a report that displays each employee information, such as name and address, and each employee orders. Because the two sets of data are stored in different tables, you must use a master-detail report design.

You design the master section to contain fields from the EMPLOYEES table and the detail or nested section to contain all records from the ORDERS table that pertains to the particular employee record in the master section. You insert one key field into your parent section. For example, insert a control ID of EMP_ID for EMPLOYEES.EMPLOYEEID.
Typical examples of the queries for the two sections are:

- Master section query:
  
  ```sql
  SELECT EMPLOYEES.FIELD1, ... FROM EMPLOYEES WHERE ... ORDER BY ...
  ```

- Detail or nested section query:
  
  ```sql
  SELECT ORDERS.FIELD1, ORDERS.FIELD2, ... FROM ORDERS WHERE
  ORDERS.EMPLOYEEID = @EMP_ID@ ORDER BY ...
  ```

### Query substitution for nested sections

e.Report Designer Java Edition reserves the `@` symbol for use with query substitution for values from the master section query to be inserted into the detail section query. The substitution occurs dynamically before execution. e.Report Designer Java Edition does a simple string substitution of the value in the WHERE clause of the inner section.

As an example, to make the nested section query and create the dependency between the detail and the master sections the query for SEC_01 inAdvMultiSections.jod uses:

```sql
Orders.EmployeeID = @emp_id@  AND Orders.OrderDate Between #1-1-1994#  AND  #3-31-1994#
```

The e.Report Designer Java Edition reserves the `@` symbol as the delimiter character that specifies the control ID of the master section used to create the dependency. The `@emp_id@` specifies the EMP_ID control in the EMP_DTL zone that displays EMPLOYEES.EMPLOYEEID.

### How to modify a section query

You can modify a section query as follows:

1. Select a section of the report, such as the DTL_00.
2. Choose Tools ➔ Section SQL Query.
   
   The list of available SectionIDs appears.
3. Select the SectionID for the section you want to modify. For example, choose Sec_01.
   
   Usually, the ID is a control inserted into the outer section detail zone and often refers to a Data Field. For example, the EMP_ID refers to the Data Field EMPLOYEES.EMPLOYEEID column in the database.

   Section’s Query appears.
The where clause in the query contains a reference to the control EMP_ID that is in the EMP_DTL zone.

4 Change the last date from #03-31-1994# to #9-1-94#. Choose OK to save the change to the query.

5 Choose OK.

When you preview the report, the nested section executes for each value of EMPLOYEES.EMPLOYEEID that it dynamically receives from the master section.
Formatted string substitution

Depending on the driver or the database, you provide a value in a specific format. This requirement occurs frequently with date and string values. Ensure that the format of the value being substituted is appropriate. Format the values before they are sent to the inner query for string substitution and execution.

Some databases require a string enclosed in single quotation marks ('). In such a case, the WHERE clause of a inner section must be:

```
WHERE Table.Column <condition> '@NAME@'
```

You put a single quotation mark on either side of the @ sign. All formatting characters go outside the @ delimiters.

A complex example is a detail section that depends on a master section Date field. The database not only expects the date field to be delimited by the # sign, but also expects the date field to be in MM-dd-yyyy format.

You specify the # delimiters in the nested section query. We could have the nested section query contain the clause:

```
WHERE Table.Column <condition> #@DATEFIELDID#@
```

How to ensure that the date field format is correct

In AdvMultiSections.jod, specify the format in the DTL_01 section using a hidden control. The SEC_02 query WHERE clause contains a reference to a date control ODATE_ID:

```
`Order Details`.ProductID = Products.ProductID AND `Order Details`.OrderID = @order_id@ AND Orders.OrderDate = #@ODATE_ID#@
```

The control ODATE_ID is an invisible field in the DTL_01 section.

2. In the Report Explorer, navigate to the DTL_01 section and select ODATE_ID.
ODATE_ID properties appear.

3 Choose Properties in the Source field to display Select Field.

4 Check the format next to the Change Format button.
   The date format should be MM-dd-yyyy.

When the value of ODATE_ID passes from DTL_01 to the nested section, it passes in the specified format. This date string is delimited by the # character and sent to the database for execution.
Understanding date field formatting guidelines

Formatting date strings often leads to errors. In general, while formatting date fields, follow these guidelines:

- Determine what date format your data source accepts.
- Set the date format in the master section to that date format.
- Monitor the queries being executed in the console window to see what date formats are being used.
- Determine what character delimiter your data source uses.

Understanding data field formatting guidelines in application data source modules

For database reporting, the formatting requirements for strings depend on your database and the database driver. When using the application data source module to report from your Java application data, the application data source query processor imposes the following rules for formatted strings:

If using the application data source, delimit string values using a single quotation mark. For example, specifying string fields for multiple-sections with application data source, format your inner section query as follows:

```
WHERE Table.Column <condition> '@FIELDNAME@'
```

Understanding three-part SQL

Databases such as Oracle and DB2 support prepending either a table owner name or the schema or the view name to make a unique reference to a column. Instead of referring to a column in the TABLE.COLUMN format, they refer to the column in the SCHEMA/VIEW/OWNER.TABLE.COLUMN format. This technique is often referred to as three-part SQL.

e_Report Designer Java Edition supports three-part SQL if you turn on support for it.

How to turn on three-part SQL support

1. Choose Tools ➜ Options.
   Options appears.
2. Choose Query.
Understanding a MetaData data source

e.Report Designer Java Edition includes a MetaData source that offers some advantages in report design. A MetaData source is a proxy data source for specifying either a JDBC data source or an application data source. You use a MetaData source to create reports offline when the actual data or data source is not available. You can swap a MetaData source with an actual data source at any time while running e.Report Designer Java Edition.

Introducing a MetaData layer

A MetaData layer, called MetaData Source, is a file that defines the structure of your data for prototype reports. It defines structure without having to pass live data to e.Report Designer Java Edition. Before creating the query, e.Report Designer Java Edition queries the database’s MetaData for the table owner’s name. It then prepends that qualifier string to the table name.

Working with parameter fields

Parameter reports enable users to generate reports based on individual criteria which may not be of interest to most users of the same report. Such reports require input from the user. For example, a sales analyst runs a sales report to view data for the entire country, a particular state, or for his own territory. He can query data for specific time periods such as the past quarter or the past
week. The report developer uses parameter fields to create a flexible report that adapts easily to input from each of the sales analysts. The user then enters parameters of his choice to generate the parameter report.

**How to create parameter field reports**

Follow these steps to create a parameter report using ForNorthwind\SingleGrouping.jod.

1. Choose Insert ➤ Report Parameters to define the parameter field.

   New parameter entry appears.

2. Enter MyParam and choose OK.

   Do not enter any spaces such as tabs or space characters.

   Parameter appears with MYPARAM listed as an available parameter.

3. Enter values for the Parameter:

   1. For the Description, an optional field, enter Customer Contact. Use text that functionally describes the field. You see this description when the report runs.

   2. For the Data Type specify Text. Available data types are text, date, time, datetime, boolean, integer or double.

   3. For default value, select Select Value and add two values Sales Manager and Sales Agent. The default value is optional.
4 Choose OK.

This parameter can be used for filtering and sorting inside the WHERE and the ORDER BY clauses of the section SQL query.

4 Select SEC_00 and choose Edit Section Query to edit the section query.

5 In the WHERE clause enter the following clause

Customers.ContactTitle = '{?MyParam?}'
6 Choose Save Changes to save the parameter settings.
7 Preview the report to see the report using the updated query and the parameter.

8 Choose Insert ➜ Report Parameters
9 Change the value of MYPARAM to generate a report containing different data.
10 Choose View ➜ Refresh to update the report preview.

**How to change parameter values**

1 Choose View ➜ Prompt for Parameters.
2 Update the preview.
3 Using Enter Parameters Values, can change parameter values.
How to delete parameter fields

Follow these steps to delete fields from parameter reports.


2. Select the parameter MYPARAM and choose Delete Entry. Delete confirmation? appears.

3. Choose No.
Customizing parameter fields

Often when working with DATE and DATETIME fields, the database requires dates in a specific format. If the default date format does not match your database date format, change the date format.

How to change the date format

Follow these steps to create a parameter report using ForNorthwind\AdvMultiSections.jod.

1. Choose Insert ➤ Report Parameters to define the parameter field.
   New parameter entry appears.

2. Enter End_Date and choose OK. Do not enter any white spaces such as tabs or space characters.
   Parameter Descriptor appears with END_DATE listed as an available parameter.
3 Enter the values for the parameter:
- For the Description, an optional field, enter End Date Range.
- For the Data Type select Date.
- For the Format enter MM-dd-yyyy.
- For default value, enter 6-12-1994.

4 Select SEC_01 and choose Edit Section Query to edit the section query. The WHERE clause is:

```
Orders.EmployeeID = @emp_id@ AND Orders.OrderDate Between #1-1-1994# AND #03-31-1994#
```

5 Change the last date in the Where clause to `#{?End_Date}?`:

```
Orders.EmployeeID = @emp_id@ AND Orders.OrderDate Between #1-1-1994# AND #{?End_Date}#
```

6 Choose OK to save the change and return Parameter Descriptor.

7 Close to dismiss Parameter Descriptor and save changes.

8 Choose Preview to see the report using the updated query and the parameter.
If you enable View ➤ Prompt for Parameters and you update the preview, you can change the parameter date value using the Enter Parameter Values.

**How to create a custom list of values**

If you do not want the user to enter an arbitrary value for the parameter field value, or, for security reasons, you want to limit the possible values that the user can enter, provide a list of values from which the user chooses.

The steps you used to create a parameter report using ForNorthwind\SingleGrouping.jod creates a custom list of values. See “How to create parameter field reports” earlier in this chapter.

**How to suppress prompting for parameters**

How to set delimiters for text and date values

Databases require text and date values to be enclosed in delimiters. As examples, date and text information might need to be enclosed in special characters:

- Dates can require being enclosed in the # symbol. In the query, enclose the parameter between # symbols, as in the following example:
  
  `Orders.EmployeeID = @emp_id@  AND Orders.OrderDate Between #1-1-1994#  AND  #(?End_Date?)#`

- Text can require being enclosed in quotation marks. In the query, enclose the parameter in single quotation marks, as in the following example:
  
  `Customers.ContactTitle = '(?MyParam?)'

How to modify a query when creating a parameter

1. Choose Insert ➤ Report Parameters to define the parameter fields.
2. In Parameter Descriptor select the section and choose Edit Section Query to display the section query.

How to modify the query using the Section Query

1. Choose Tools ➤ Section SQL Query ➤ SectionID. Section’s Query appears.
2 Define parameters in the WHERE section of the query. Enclose the parameter fields between {?...?} delimiters. For example, {?MyParam?}.

3 Select OK to save the changes to the Query.

For information aboard creating a report parameter, see “How to create parameter field reports” earlier in this chapter.

**How to display values entered by the user**

You often want to display parameter values in the report. A simple example is a Sales report that has two parameter fields, Start Date and End Date. You use the runtime parameter values to display the values as follows.

1 Define one or more parameter fields.

   “How to change the date format” earlier in this chapter, describes how to create the date parameter End_Date in ForNorthwind\AdvMultiSections.jod.

2 Choose Insert » Runtime Parameter to enter a parameter.
3 Drag and drop the field into the report.
In the example, the parameter field is dropped into the EMP_SALES_HDR zone above the date.

Runtime Parameter Options appears.

4 To use an existing parameter, select Use Existing Query parameter value and select the parameter. Choose OK to accept the settings.

The parameter field is added.

5 Preview the report to see the value displayed in the report.
Working with stored procedures or functions

You can extract data from stored procedures and functions that have been defined in your database.

Supported types of stored objects

You can insert the following types of stored object definitions into your template:

- **Type 1**: A stored object with input parameters. Specify values as a constant using the wizard or stored objects descriptor. Another option is to dynamically extract from a control in the template.

- **Type 2**: A stored object with output parameters. Values can be retrieved after the stored object is called and executed. This process employs a user defined function called OutParam().

- **Type 3**: A stored object returns a single value after execution. Use a user defined function called StoredProc() to create this object. Used StoredProc() to execute a stored function whose properties are defined by the user defined function arguments. StoredProc() returns a value that is the evaluated user defined function.

- **Type 4**: A stored object that generates a single set of query results that can be mapped into a single section with appropriate column headers.
■ Type 5: A stored object that generates multiple sets of query results, each of which can be mapped to a single section. The section can be either nested or sibling, with column headers.

■ Type 6: An Oracle stored object that returns a single query result set or multiple result sets via one or more ref cursor Out parameters.

Examples of stored objects

This section provides samples of stored procedures and functions. These examples are for demonstration purposes and are not intended to be used in a production environment.

Oracle example

Type #1, #2
PROCEDURE SO_ORA_12(x IN int, y OUT int) IS
BEGIN
  SELECT Count(Emp.EmpNo) INTO y FROM Emp WHERE Emp.EmpNo < x;
END;

Type #1, #3
FUNCTION SO_ORA_13(x IN int) RETURN STRING IS
  y VARCHAR(20);
BEGIN
  SELECT dname INTO y FROM dept WHERE dept.deptno = 40;
  RETURN y;
END;

Type #4

Oracle handles result sets as ref cursor objects. These objects are given as OUT parameters in the stored object signature. OUT parameter variables of the ref cursor type must be defined earlier in some package in the Oracle database. An example of creating a packaged ref cursor variable follows:

CREATE OR REPLACE PACKAGE MyCursor
IS
BEGIN
  type cur_var is ref cursor;
END MyCursor;

After the cursor variable is ready, stored objects can be created as follows:

CREATE OR REPLACE PROCEDURE Proc1 (mycur OUT MyCursor.cur_var)
AS
BEGIN
  OPEN mycur for SELECT empno, ename from emp;
  return;
END;
MULTIPLE ResultSets via different REF CURSOR OUT Parameters
CREATE OR REPLACE PROCEDURE Proc1 (firstresult OUT
   MyCursor.cur_var, secondresult OUT
   MyCursor.cur_var)
AS
BEGIN
   OPEN firstresult for SELECT empno, ename from emp;
   OPEN secondresult for SELECT deptno, dname from dept;
   return;
END;

Sybase SQL Anywhere example
CREATE PROCEDURE DBA.SO_SSA_14(IN custid INTEGER)
RESULT(id INTEGER, fname CHAR(35))
BEGIN
   SELECT Customer.Id, Customer,FName FROM Customer WHERE
   Customer.ID = custid
END

FUNCTION dba.SO_SSA_13(IN x INTEGER)
RETURNS VARCHAR(20)
BEGIN
   DECLARE y VARCHAR(20);
   SELECT fname INTO y FROM customer WHERE customer.id=103;
   RETURN (y)
END

PROCEDURE DBA.SO_SSA_2(INOUT param varchar(30))
ON EXCEPTION RESUME
BEGIN
   SELECT fname INTO param FROM customer WHERE lname='Devlin'
END

PROCEDURE DBA.MULTIRS3
AS
BEGIN
   SELECT employeeid,firstname,lastname FROM employees
   SELECT * FROM customers
   SELECT * FROM orders
END
MS SQL Server 7 example

Type #1, #4
PROCEDURE SO_MSS_14 @param varchar (30) AS
   SELECT * FROM customers WHERE customers.ContactTitle = @param

Type #2
PROCEDURE SO_MSS_2 @maxqty INTEGER OUTPUT AS
   SET @maxqty = 20

Type #5
PROCEDURE MULTIRS3 AS
   SELECT PRODUCTID, PRODUCTNAME FROM PRODUCTS
   SELECT * FROM CATEGORIES

MS Access example

Type #1, #4
PARAMETERS [CatName1] Long, [CatName2] Long;
SELECT [products].[productid], [products].[productname], [products].[supplierID]
   FROM Products
   WHERE [Products].[CategoryID]=CatName1 Or [Products].[CategoryID]=CatName2;
PARAMETERS [category id] Long;

Understanding requirements

In Stored Object Descriptor, you must explicitly resolve all parameters (IN, chooseOUT, choose) that are used in processing the stored object. You must also specify optional columns for query results that you want to appear in your output.

Sometimes the data type for a parameter is incorrectly returned as Other. Change this erroneous value to the appropriate data type as defined by the stored object prototype.

In order to insert a stored object of Type 4, 5, or 6 into e.Repot Designer Java Edition, you must first either create a new section or convert an existing section into a StoredProcedure section. Any result set structure to be inserted into e.Repot Designer Java Edition must be inserted as a section with an appropriate header containing column title values and a footer that contains summary values or output parameter values.
Entering constant values

When you enter constant values in the Descriptor, ensure that the value is specified in the correct format as defined by the data type. Do not change the data type of the parameter if you are a novice user. Before attempting to override the data type, understand the following limitations:

jConnect driver

Incapable of identifying the difference between a return type property and an OUT parameter property, as a result, OUT parameters are not correctly listed in the parameters sheet. Convert the OUT parameter in your database into an choose parameter.

Oracle Thin driver

The following problems can occur with this driver:

- Extremely slow while retrieving metadata information.
- Extremely slow while executing stored objects.
- Incapable of returning the correct data types for parameters. In order to circumvent this problem, override the default data types that are extracted using the jdbc driver metadata information. Use the following Parameter Specification Sheet when defining a stored object.

![Parameter Specification Sheet](image)

You must know the data type of any OUT or INOUT parameters or choose parameters that are defined in your stored object. Those data types must be
correctly specified here. If correct specification is not done, an error occurs when buffering the output parameters after the stored object is executed. This information can be embedded with the data source inside the template and is persistent.

**MSAccess ODBC driver**

Incapable of returning the correct data types for parameters. Use the workaround described for Oracle Thin driver.

**MS SQL Server**

The JDBC ODBC bridge used to interface with a MS SQL server database is incapable of retrieving generic result set information using the ResultSet

getObject( ) method. Data can be retrieved using only the appropriate

getXXXX( ) methods. This limitation results in not being able to detect the required metadata information.

**Output parameters**

Output parameter values are, by default, buffered and stored in the JDBC stored object implementation. They are maintained internally and can be provided when requested. To add a control that is linked to the stored object control’s output parameter, check Create a control attached to the OUTPUT value?. Then, you can specify the control ID for the object whose value is retrieved as the output parameter value.

Internally, the request for the output parameter value is made using a user defined function:

```java
OutParam("soID", "outputParamName")
```

In this function, soID refers to the Stored Object control, which is defined by the following additional line:

```java
StoredProc("soName", "soID", "encodedArgsList")
```

The stored object control is a special user-defined function that you modify only through the Stored Object’s Descriptor.

---

**Inserting a stored object programmatically**

e.Report Designer Java Edition supports stored objects only for an active JDBC data source connection.
How to create a section for a stored object

1. Select any existing section on the left hand side and choose Insert (StoredProc.) Section as SIBLING or NESTED.

   A default Detail zone appears for this section and the Stored Object Descriptor displays.

2. Select a Type 4, 5, or 6 stored object from the available list of objects.

3. Resolve input parameter values by either linking them to an existing control in the report or specifying a constant value.

4. To generate any output parameters as controls, specify their IDs and choose Create a control attached to the OUTPUT value?

How to insert a stored object in a report

1. In the Designer, choose the Stored Procedure tool.

2. Choose the location in the report where you want to place the stored procedure.

   If you have an active connection to the data source, Stored Object Descriptor appears.

3. Choose Specifications.
4 Specify how input and output parameter values are resolved at runtime. Input parameter values can be specified either as a constant value or linked to another control value.

The specifications of the Stored Object Descriptor include the following.

<table>
<thead>
<tr>
<th>Wizard control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter list</td>
<td>Lists the available parameters for the selected stored object. These parameters can be of type IN, OUT, or INOUT. Select the parameter and specify its properties.</td>
</tr>
<tr>
<td>Input properties</td>
<td>Checkboxes resolve the value of an input parameter as a constant value or as a link to an existing control embedded inside the template. A constant must be of the correct data type as indicated by the parameter metadata information; otherwise an error occurs. A link to a field or formula control is substituted in the input parameter to the stored object. It must be of the correct data type.</td>
</tr>
<tr>
<td>Output properties</td>
<td>If Adding an Output Parameter is checked, an output parameter with the specified control ID is added to the template at the appropriate location, depending on the type of the stored object. If the stored object generates a set of results, the output parameter is inserted into the outermost group footer of the corresponding section. If the stored object is inserted as a control that updates the database or has a single return value, the output parameter is inserted as a control within the same zone containing the stored object control. If there is an output parameter control in the template that refers to the same stored object, it is not recreated. If you want to recreate copies of an output parameter, make copies of the same formula by copying and pasting the existing formula.</td>
</tr>
</tbody>
</table>
Certain JDBC drivers do not return the correct metadata information about stored objects. You sometimes must override the output parameter data type for it to be successfully registered internally before it can be executed.

**Working with the return value**

1. In the Descriptor, select Return Value and specify a control ID. This procedure updates the newly inserted Stored Object Control’s ID with the specified ID and ensures that the returned value is the evaluated Stored Object Control result.

2. Choose Finish.

   The new user defined functions are created that represent the stored object and its output parameter links.

3. View the report.

<table>
<thead>
<tr>
<th>Wizard control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultSet column descriptor</td>
<td>Types of stored objects can exist in a variety of databases. These stored objects are capable of returning an entire set of results. When such a stored object is inserted into a template, it is usually inserted as an entire section. When a stored object returns a tabular set of results, the user has the option of selectively choosing the columns to insert into the template. In the ResultSet Column Descriptor, the desired columns can be individually selected and added to the template. These columns get added into the detail zone of the newly created section. A column header can also be added. For example, the title for that column would be inserted into the corresponding group header created above the detail zone.</td>
</tr>
</tbody>
</table>

Return value | Certain databases support executing a stored object and returning a single value as the result. A stored function can accept one or more IN, OUT or INOUT parameters, process them as required by the function, and return a single value result. You insert such a stored function into the template as a single control. A set of formulas can be linked to the output parameter of this control using the OutParam user defined function. |
How to insert a stored object in the new section

As soon as the default Detail zone is created for this section, a Stored Object Descriptor is invoked. The properties need to be defined.

1. Select a stored object from the available list.
2. Specify the resolution of input parameter values by linking them to an existing control in the report or by specifying a constant value.
3. If any output parameters need to be generated as controls, specify their IDs and select Create a control attached to the OUTPUT value?

4. After defining the parameters, describe the columns in the set of results to be extracted.
5. Choose the ResultSet sheet. Select each of the columns listed and specify if they should be added into the template.

You can also add an optional column title to each of the columns.
6 After specifying the columns of results to be added, choose Finish.
   The new section is created in the Designer with the result set structure.

7 View the report.

---

**Inserting stored objects manually**

When a JDBC Driver is not able to extract the correct stored object metadata from the database or is unable to get information regarding the result set(s) of the stored object, you can insert stored objects manually. In Manual mode, you can specify all IN, INOUT, OUT parameters and data types that the stored object takes. Any discrepancies can be rectified by adding or removing stored object parameters. All IN parameters must be resolved by a constant value.

**How to insert a stored object manually**

1 In the Stored Object Descriptor wizard, choose Manual Mode.
   Stored Object Prototype appears and prompts you to resolve the parameters and data types.
2. Resolve the parameters and their INOUT data types.
3. Choose OK.

The result set displays in the ResultSet page.

**How to resolve Oracle result sets manually**

Using Manual mode, you can easily detect Oracle result sets returned via OUT parameters that have the Ref Cursor data type.

1. Choose the data type REF CURSOR for the OUT parameter.
2. Resolve all IN parameter values with constants.
3 Choose OK.

The ResultSet page displays the result.

---

**Working with multiple result sets**

eReport Designer Java Edition supports returning multiple result sets from a stored object. Each result set is identified by the sequence in which it is returned by the stored object, for instance ResultSet_1, ResultSet_2, and so on, as shown in the following illustration.
Each result set has a list of associated Column Names, from which you can select the columns to include in the report.

A new sibling section is created for each of the result sets selected. If you initially select a stored object nested section, a new nested section is created for the first result set. The remaining result sets are siblings of the new section. If you initially select a stored object sibling section, all the result sets you select are separate sibling sections.
Managing an empty result set

If the database returns an empty result set, the preview panel is blank. In such cases, you might choose to print a message, such as No Data Retrieved. To print a message, in the Descriptor, select Print section if ResultSet is empty. Then type the message in Empty ResultSet Message.

Limiting the result set to specific rows

If you want to limit printing or display to specific rows of the result set, identify the rows as a range of integers in ResultSet range From: and To:.

If you limit the result set and no rows are retrieved, the message No Data Retrieved prints in the preview panel.
Saving result set columns in a template

e.Report Designer Java Edition supports saving a result set’s columns in a template. You can load a template containing a stored object section and add additional columns from the result set for that section. To do so, you must first set the template options.

How to set template options to save result set columns

1. In the Designer, choose Tools ➔ Options.
   
2. Choose Template.
3. Select Save Resultsets’ Columns’ information in Template.
4. Choose Save & Apply.

How to save result set columns

1. In the Designer, right-click on a stored object section.
2. Choose Insert object ➔ Data field.

   A dialog displays the columns for the result set.
Chapter 6, Customizing e.Report Designer Java Edition

This chapter contains the following topics:

- Customizing designer functions
- Customizing general report properties
- Customizing report page properties
- Understanding sample ACJDesigner.properties
Customizing designer functions

e.Report Designer Java Edition has many custom design features for report generation, database modification, and output control. User-defined options control how e.Report Designer Java Edition functions, how it appears in design mode, and how it displays output. To help customize your reports and design session, the report designer offers pages of options. Options can be found in the Tools menu.

Options Properties

Every page in Options has many features. Use the General page to customize your design session.

General Page

The following table shows the features of the general page.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal gap between controls</td>
<td>Specifies default spacing between controls when using Alignment→Arranged Horizontally from the Multiple Selection context menu.</td>
</tr>
<tr>
<td>Division per inch</td>
<td>Available range is from 1 to 48</td>
</tr>
<tr>
<td>Look and Feel</td>
<td>Changes the e.Report Designer Java Edition user interface. Change the Look and Feel to match your desktop environment. Available options are Metal, CDE/Motif, and Windows.</td>
</tr>
</tbody>
</table>
### Control Defaults

Use this page to set new default settings for color, style, font, alignment and border style for the controls listed in the window below.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units of measurement</td>
<td>Available options are inches and centimeters</td>
</tr>
<tr>
<td>Snap to grid</td>
<td>Movement and resizing of controls restricted by grid dots. The fixed grid resolution is 1/20th of an inch.</td>
</tr>
<tr>
<td>Show grid</td>
<td>Displays or hides the grid on default template.</td>
</tr>
<tr>
<td>Add label to new fields</td>
<td>Adds a label or data field. By default, the label is added to the zone directly above the data field zone. If the zone above belongs to a different section or is hidden, the label is not added.</td>
</tr>
<tr>
<td>Make image path relative</td>
<td>Sets the image path as relative rather than absolute.</td>
</tr>
</tbody>
</table>

### Workspace

Use Workspace to set up your workspace environment.
### Options

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Working directory</strong></td>
</tr>
<tr>
<td><strong>Output directory</strong></td>
</tr>
<tr>
<td><strong>Browser path</strong></td>
</tr>
<tr>
<td><strong>PDF viewer path</strong></td>
</tr>
<tr>
<td><strong>Mail application path</strong></td>
</tr>
</tbody>
</table>
Query

Use Query to update SCHEMA query. For more information on this, see “How to join tables” in “Querying the database.”

A check in any box causes an automatic update

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto update WHERE</td>
<td>Updates WHERE clause joins based on data field references in the template and the relationships between them. If joins are manually added, then the Auto Update Where mechanism appends all missing joins to ensure proper query execution.</td>
</tr>
<tr>
<td>Auto update ORDERBY</td>
<td>Updates ORDERBY clause based on the data field references grouping criteria setup using the Group Headers.</td>
</tr>
<tr>
<td>Auto prepend SCHEMA</td>
<td>Meets requirements of some SQL engines, such as Oracle8, by prefixing the TABLE name in the FROM clause with the SCHEMA of that table. Other SQL engines, such as Informix, fail if the FROM clause contained TABLES prefixed with the SCHEMA.</td>
</tr>
<tr>
<td>Surround Lowercase Quotes</td>
<td>Meets requirements of some SQL engines to enclose in quotation marks all table names and column names containing any lowercase letters. Typically required by Oracle8i.</td>
</tr>
</tbody>
</table>
**Template**

Templates allow you to store report formats or data sources for later use. Functions stored within the template execute automatically when you load the template into your design editor.

Options | Description
---|---
**Encoding** | Templates can be saved or loaded in either ASCII or UTF8 formats. Selecting this option specifies the default format. Note that templates written in one format cannot be read in a different format.
**Save Datasource in template** | Saves the data source with the template. If this option is not enabled, you must establish a connection to the data source before generating the report.
**Print report footer on new page** | Sets the report footer on a new page.

**Output**

Choose this page to set specification for Table of Contents.
Debug
Use this page to set the debug mode in the appropriate data modules.

This page enables filtering of errors. You can use this page to help you monitor the progress of your activities.

Available debug options

Choosing this bar gives you access to color swatches and attributes.

Output

<table>
<thead>
<tr>
<th>Description</th>
<th>Output Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of contents background color</td>
<td>Sets the background color of the table of contents.</td>
</tr>
<tr>
<td>View table of contents in output</td>
<td>Displays the table of contents on the Preview page to facilitate navigating the report.</td>
</tr>
</tbody>
</table>
Customizing general report properties

Report Properties

This page affects properties that apply to the overall report. To open Report Properties, choose File ➤ Properties.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Generates data for examining the query being executed.</td>
</tr>
<tr>
<td>Application Data Module</td>
<td>Generates data for examining exceptions thrown by the JDBC data module.</td>
</tr>
<tr>
<td>JDBC Data Module</td>
<td>Generates data for examining exceptions thrown by the JDBC data module.</td>
</tr>
<tr>
<td>Engine</td>
<td>Generates data for examining exceptions.</td>
</tr>
<tr>
<td>GUI</td>
<td>Generates data for examining exceptions thrown by Swing or AWT.</td>
</tr>
</tbody>
</table>

For more information about properties that affect the style of the report, see Chapter 2, “Understanding the design process.”
The following table provides detailed information about properties you can set.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report title</td>
<td>Labels the report. The report title is preserved when exporting the report to HTML, PDF, or other file formats.</td>
</tr>
<tr>
<td>Author</td>
<td>Identifies the person responsible for the report, their company, and their department.</td>
</tr>
<tr>
<td>Description</td>
<td>Summarizes report purpose logic used and data structure.</td>
</tr>
<tr>
<td>Modified</td>
<td>Displays the date and time of last save. You cannot change this property.</td>
</tr>
<tr>
<td>Template</td>
<td>Displays the physical location of the report template file on your hard disk. You cannot change this property.</td>
</tr>
<tr>
<td>Form report</td>
<td>Facilitates printing on pre-printed paper. In this report, all controls are suppressed except Fields, Formulas, and Charts. As a result, the printout can be made on the pre-printed page.</td>
</tr>
</tbody>
</table>

**Customizing report page properties**

Report Page Properties are under File ➔ Page Setup.

**Page Setup**

Use Page Setup to customize page properties.
Export properties
You can set export properties for each of the supported file formats. Open Export Properties by choosing File ➤ Export Properties.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Borders       | Displays a border around each page of the report. Checking the box activates the PageBorder Color and PageBorder Style.  
  - Click the area to the left of the PageBorder Color to access the color palette and other settings.  
  - Click the area to the left of the PageBorder Style repeatedly to select a line style for the border. |
| Dimensions    | Sets the page height and width in units of measurement, set in Tools ➤ Options. Default height is 11 inches and default width is 8 1/2 inches. |
| Margins       | Sets left, right, top and bottom margins of the report in inches. |
| Orientation   | Displays your report in portrait or landscape mode. Default is Portrait. If you are using Java 1.1, you must manually set the printer’s orientation to Landscape, before you can print a Landscape mode report. Both modes are supported in Java 2. |
| Wallpaper     | Adds images to the background of a report page. For example, use this property to add the text Confidential as a bitmap watermark to a report. Enter a URL to the image file or click the Browse button to select a local file. By default, the image is tiled beginning at the top left corner. |
The following table describes the available export properties.

<table>
<thead>
<tr>
<th>HTML export properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zipped output</td>
<td>Saves the file compressed. Used for creating and distributing large reports. If you choose this option, save the file with a .zip extension.</td>
</tr>
<tr>
<td>Show table of contents in output</td>
<td>Creates the table of contents in the output. If you choose this option, a series of HTML files is created. The main file contains two frames. Each of those frames contains an HTML file. The frame on the left contains the table of contents in HTML format. The frame on the right contains the report in HTML format.</td>
</tr>
<tr>
<td>Preserve color</td>
<td>Preserves colors specified in the HTML report. Setting this option increases the size of the HTML file.</td>
</tr>
<tr>
<td>Preserve font</td>
<td>Preserves fonts specified in the HTML report. Setting this option increases the size of the HTML file.</td>
</tr>
<tr>
<td>Cache</td>
<td>Specifies whether or not the HTML output header information contains the Cache option. Caching the HTML file can prevent the browser from refreshing the report data when it changes.</td>
</tr>
<tr>
<td>Character set</td>
<td>Specifies which character set to use. Currently, only US and Western European character set (ISO-8459) is supported. Multilingual (UTF8) is not supported.</td>
</tr>
<tr>
<td>Preserve page break</td>
<td>Maintains page footer and header information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PDF export properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zipped output</td>
<td>Saves the file as a compressed file. Used for creating and distributing large reports. If you choose this option, save the file with a .zip extension.</td>
</tr>
<tr>
<td>Show table of contents in output</td>
<td>Creates the table of contents in the exported report. Bookmarks are defined in the generated PDF file.</td>
</tr>
<tr>
<td>PDF export properties</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Subject</td>
<td>Provides a short description of the report.</td>
</tr>
<tr>
<td>Magnification</td>
<td>Specifies the magnification at which your report is displayed when opened by a user.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DHTML export properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zipped output</td>
<td>Saves the file as a compressed file. Used for creating and distributing large reports. If you choose this option, save the file with a .zip extension.</td>
</tr>
<tr>
<td>Show table of contents in output</td>
<td>Creates the table of contents in the output. If you choose this option, a series of HTML files is created. The main file contains two frames. Each of those frames contains an HTML file. The frame on the left contains the table of contents in HTML format. The frame on the right contains the report data in HTML format.</td>
</tr>
<tr>
<td>Preserve color</td>
<td>Preserves colors specified in the HTML report. Setting this option increases the size of the HTML file.</td>
</tr>
<tr>
<td>Preserve font</td>
<td>Preserves fonts specified in the HTML report. Setting this option increases the size of the HTML file.</td>
</tr>
<tr>
<td>Cache</td>
<td>Specifies whether or not the HTML output header information contains the Cache option. Caching the HTML file can prevent the browser from refreshing the report data when it changes.</td>
</tr>
<tr>
<td>Character set</td>
<td>Specifies which character set to use. Currently, only US and Western European character set (ISO-8459) is supported. Multilingual (UTF8) is not supported.</td>
</tr>
<tr>
<td>DHTML Target Browser</td>
<td>Specifies the target browser for the DHTML.</td>
</tr>
</tbody>
</table>
### CSV export properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separator</td>
<td>Specifies the character or string to be used as the separator of the records of data in each line of the file. The separator can either be a single character or a series of characters, but you must specify the correct value. If you are not sure how the separator works, use a comma.</td>
</tr>
</tbody>
</table>

### Text qualifier

Qualifies text data with a character so external applications can understand and parse the data. The text qualifier is often used with text strings that contain the Separator character. The text qualifier can either be a single character or a series of characters. Text qualifier characters are prepended and appended to all string values in the report.

### Email export properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMTP server</td>
<td>Specifies the SMTP mail server used to send email. Check with your system administrator if you do not know what your SMTP server is.</td>
</tr>
<tr>
<td>Sender’s email address</td>
<td>Specifies the return email address to be used.</td>
</tr>
<tr>
<td>Address book of recipients</td>
<td>Specifies one or more email addresses to which you want to email the reports. If you specify more than one email address, use a comma to separate the addresses.</td>
</tr>
<tr>
<td>Default report styles</td>
<td>Specify the file format in which the report is e-mailed to the recipients. Options are HTML, PDF, DHTML, CSV, and JOI.</td>
</tr>
</tbody>
</table>

---

**Understanding sample ACJDesigner.properties**

The defaults you set in Properties are stored in the ACJDesigner.properties file, usually located in the main working directory. This section shows a sample properties file. You can also modify the properties file directly in a text editor:

```properties
#DESIGNER SETTINGS
#Mon Aug 14 14:29:04 GMT+05:30 2000
Designer.Query.PrependSchema.FROM=false
```
Designer.Template.Attrs.LABEL=Dialog,PLAIN,11,0,0,NULL,VERTICALCENTER,HORIZONTALCENTER,U_NONE,false,false,L_NONE,0,0,0,0.4270833333333333, 0.16666666666666666
Designer.LookAndFeel=Windows
Designer.Template.Author=Acutate e.Reports
Designer.Template.Attrs.FORMULA=Dialog,PLAIN,11,0,0,NULL,VERTICALCENTER,HORIZONTALCENTER,U_NONE,false,false,L_NONE,0,0,0,1.0, 0.16666666666666666
Designer.Template.AddLabelsToFIELDS=false
Chapter 7, Presenting data in charts, graphs, tables, and crosstabs

This chapter contains the following topics:
- Types of graphs and charts
- Understanding X-axis and Y-axis
- Sending charts and graphs to an output device
- Creating asymmetrical grids
- Creating tabular reports
- Creating columnar reports
- Creating crosstab reports
Types of graphs and charts

Informative graphic data that is well-integrated with text makes a report more visually appealing and interesting. e.Report Designer Java Edition scales line charts and histograms on the fly based on the available data.

e.Report Designer Java Edition supports the following types of graphs and charts:

- 2D Comparison/column (side by side) Bar Graphs (Horizontal Orientation)
- 2D Comparison/column (side by side) Bar Graphs (Vertical Orientation)
- 3D Comparison/column (side by side) Bar Graphs (Horizontal Orientation)
- 3D Comparison/column (side by side) Bar Graphs (Vertical Orientation)
- 2D Stacked Histogram (Horizontal Orientation)
- 2D Stacked Histogram (Vertical Orientation)
- 3D Stacked Histogram (Horizontal Orientation)
- 3D Stacked Histogram (Vertical Orientation)
- Line graphs
- 2D Pie Charts
- 3D Pie Charts

How to insert charts or graphs into a report

This section discusses how to insert charts or graphs into a report using ForNorthwind\SingleGrouping.jod. Charts and graphs can be inserted only in group footer or report footer zones.

1. Open the report and scroll down to the report footer.
2 Increase the height of the report footer by dragging your cursor on the line between the RP_FTR zone and the PG_FTR zone.

3 Remove any labels from the footer by right clicking on the label. A label menu appears.

4 Choose Delete Label.

5 Choose Chart/Graph or choose Insert ➤ Chart/Graph. Chart/Graph Properties appears.

6 Select a Chart/Graph type.

7 Choose Next to access the types of charts or graphs available. Select a Chart/Graph type appears.
Choose a 2D Vertical Comparison Barchart.
Choose Next.
X-Axis page appears.

Selecting data fields

Select the database fields from which to create the X-axis of the chart or graph. In the example, select FOR000.
Field FIE001 is the control displaying the database Customers.Country field in the report group header.
For more information about creating the X-axis, see “Defining the X-axis,” later in this chapter.

2 Choose Next.

Y-Axis page appears.

3 Select the database fields from which to create the Y-axis of the chart or graph. In the example, select FIE001.

Field FOR000 is the control displaying sum of the number customers in a country in the report group footer.
For more information about creating the Y-axis, see “Defining the Y-axis,” later in this chapter.

4 Choose Next.

Attributes appears.

You can customize fonts, colors, borders, and other properties in charts and graphs. You can include a legend, captions, and title. You can overlap data in a graph, plotting multiple items of data, unless the graph is a pie chart.

5 Choose Finish to save graph properties.

6 In the report design, enlarge the graph control to fill the space in the report footer.

When you preview the report, the graph appears on the last page of the report. You can return to the design and make changes to the graph properties.
Understanding X-axis and Y-axis

The most important task in designing a chart is identifying the data source. The data source must be a control inserted in the template within the scope of the chart. For more information about inserting a data source, see “Inserting a data field” in Chapter 5, “Querying the database.”

You can specify the data source as a single control that changes several times (One Changing Value) or as several static controls within a zone that prints just once (Many Static Values).

The following rules are essential in specifying the data sources for the X- and Y-axis of a chart or graph:

- Use only two axes in chart definition and plotting.
- Attach only numerical data to the Y-axis.

**Defining the X-axis**

Specify a data source of any data type for the X-axis.
Defining the Y-axis

Specify one or more items of data in the Y-axis. The data source for one or more items of Y-axis data must be numerical. If no data source is numerical, an error message displays.

The Y-Axis can plot several series of data for all charts except the 3D Pie Chart. The number of series can be increased using a spin control located above the Y-Axis data source definition tab. In case of the 3D Pie Chart, the currently
selected series being edited in the Y-Axis data source definition tab is used in
determining the magnitude of each slice.

---

**Sending charts and graphs to an output device**

Support for visual elements such as charts and graphs pose certain problems
when transferring them to various file formats. For example, charts and
graphs cannot be embedded inside HTML files. Another issue is that Java 1.1
does not allow for writing image objects to disk. Java 1.1 only supports
reading of images.

e.Report Designer Java Edition implements its own image writer, so you can
save charts and graphs in separate files along with the HTML file. You can also
save them to PDF file format in native PDF API. This process creates high-
resolution graphics that are not distorted when zoomed. The images do print
clearly.

---

**Creating asymmetrical grids**

One of the most common uses of a report writer is to take data from a data
source merged with pre-defined text to create a form. A form can be a
questionnaire, survey, record, or letter. Form reports often have fields for data
entry. For example, when you fill out a form for your health insurance, the
fields are most often grouped inside a grid or table. A table helps organize the
data in an efficient and logical manner.

Creating sophisticated form reports often requires boxes that are almost
always asymmetric. Most often, in form reports, space is very valuable and
form designers try to insert as much data as they can on a single sheet of
paper. This requirement means the form designer must be flexible enough to
allow a user to draw different sized boxes where data can be entered. A
standard rectangle does not work. You need a control with edges that you can
create in a free form manner. Using an asynchronous grid you can create grids,
tables, and other asymmetrical cells.

e.Report Designer Java Edition has such a control—the AsyncGrid
(asynchronous grid). The AsyncGrid resembles a spreadsheet or table that you
can create with symmetrical horizontal and vertical lines or any kind of
structure using lines that are asymmetrical. This grid has regular rows and
columns like those found in spreadsheet applications, but it also allows you to
create a grid in a free-form manner. You can join any two line segments to
create a new cell. You can use this type of grid to create very detailed reports
that do not take up much space on a page.
The following illustration shows an example of the kind of complex AsyncGrid controls you can create.

**Inserting an AsyncGrid into a report**

The basic AsyncGrid consists of the following parts:

- Four outermost bounded lines
- Two vertical lines, left and right
- Two horizontal lines, top and bottom

Each line is bound to the two perpendicular lines, respectively. For example, the left or right line is bound by the top and bottom line, while the top or bottom line is bound by the left and right line.

The rule for adding a new line is to bind it between two perpendicular lines. If you delete a segment, all dependent segments get deleted.

The AsyncGrid control supports the following operations:

- Adding segments
- Deleting segments and their dependents
- Deleting the entire AsyncGrid from the layout
- Modifying segment style and attributes
- Stretching a control to fill the entire space of a cell created by adding segments
- Aligning a segment with either the top or bottom edge of the zone

Suppress the AsyncGrid while printing pre-printed forms.

**How to insert an AsyncGrid**

To insert an AsyncGrid control inside the report template:

1. Choose Insert ➤ AsyncGrid.
   
   A special cursor labelled AsyncGrid appears.

2. Move to the location in the designer pane where you want the grid placed and click to choose that location.
   
   A rectangular control appears.

**How to define grids and tables**

You can define grids and tables of virtually any size and configuration. Use the grids to align controls in a report.

1. Right-click the Async grid line and choose Add Edge.
   
   An arrowhead appears.
2 Move the mouse vertically or horizontally until it reaches the opposite segment. When the mouse reaches the other line segment, your arrow changes color to red. Click the other line segment to choose it.

e.Report Designer Java Edition creates the line segment.

At times it becomes difficult to make the control borders match the edges of the cell of the AsyncGrid surrounding the control. To make it easier to resize the control to fix the cell, use the Stretch to Fit Cell feature for AsyncGrid.

**How to match a control border to a cell edge**

1 Scroll to the group footer or report footer.
2 Right-click the designer window. Choose Insert Object ➔ AsyncGrid.
   A grid appears.
3 Right-click the control inside the cell of the AsyncGrid. Choose Stretch to Fit cell.
   The control resizes to fit the cell.
4 Choose Preview.
   The report design appears.

---

**Creating tabular reports**

Report data is often easier to read and analyze when the data in the report is organized inside a table or column. While you can use the async grid to create tabular reports, e.Report Designer Java Edition also has tabular report styles.

**How to create a tabular report**

1 Choose File ➔ New.
   New report appears.
2 Choose Style.
3 Select Single Table and choose Finish. If you have a report loaded Save Confirmation appears.

4 Choose No.

Creating columnar reports

Unlike row reports, columnar reports can have more than one record in each row. The most common example of a columnar report is a label report. A label report essentially contains a set of labels extracted from a database. You can format the fields in this report in any way you want, subject to the available formatting styles. You can sort the generated report data on multiple keys in ascending or descending order.

A label report is often used for mailing lists. In such a case, the report prints on self-sticking label sheets.

How to create a label report

1 Choose File ➜ New.
   New Report appears.

2 Select Style.
3 Select Label Wizard. Choose Finish.

4 Choose Next.

   Create a Label Report appears.

5 Select Labels.

   Specify either a pre-defined label format or a custom label format.

   Avery and other companies provide a wide variety of pre-designed, self-adhesive paper for printing labels. Only a limited number of Avery labels
are currently supported. If the pre-designed label you use is not listed, simply enter the dimensions manually.

7 Choose Next.

Data source appears.

The pane on the left lists all the available tables.

8 Select the data fields you want to include in the label. Choose Finish.

9 Choose Preview to preview the report.

The labels are vertically arranged in a top-down manner, based on the order in which you select the fields. If the labels won’t fit vertically, asks whether you want to rearrange the fields to fit. You can re-arrange the fields or make additional changes through the report designer.

Creating crosstab reports

A crosstab report summarizes data and presents it in a compact, matrix-like format. The output resembles a spreadsheet. Crosstab reports facilitate comparing data and identifying trends.
A crosstab report typically represents summarized data, which is bound by two or more dimensions. A dimension of a data source refers to the grouping criteria specified for the data source. Most often, data stored in relational tables are connected to each other by means of primary-foreign keys. Building a crosstab report requires some knowledge of these relationships.

Creating crosstab reports is like creating grouping reports. You can specify the grouping criteria vertically and horizontally. In a typical grouping report, you specify the vertical grouping only. Grouping concepts such as group headers, footers, and grouping criteria apply to crosstab reports.

**Before creating crosstab reports**

Creating crosstabs requires knowledge of groupings and query operations. For more information about grouping and queries, see Chapter 5, “Querying the database.”

You insert a crosstab as a complete section. A section is the basic unit that the report engine understands and works with. A crosstab section is a single and complete entity. Each crosstab section has its own query. A report can have both a crosstab and regular sections within a single report. In fact, you can have a crosstab section nested inside a regular section.

The Structure page of the Create a Crosstab Report wizard is a diagram of the crosstab report. See “How to define a row and column,” later in this chapter. In the Structure tab, study the diagram to understand how crosstab reports work and the terminology associated with cross tab reports.

**Building a Cross Tab**

To build a crosstab report, start with the simplest scenario. Define a row and column of the report. The report wizard helps you format the report.

**How to define a row and column**

   
   New report appears.

2. Choose Next.

   
   Create a Crosstab Report appears.

4. Choose Next.
   
   Structure page appears displaying data relationships in a legend.
5 Choose Next.
Groups appears.

6 Choose Next.
Summaries appears.
Chapter 7, Presenting data in charts, graphs, tables, and crosstabs

How to define horizontal grouping

1. To define the horizontal grouping based on a formula, enter the formula in the text box under Specify OR Update the Grouping Criteria here.

2. Choose Add Entry.

3. To define the horizontal grouping based on database columns, choose PASTE Data Field.

Innermost Summary Field appears.

Additional choices
Choose here for the pull-down menu below
These terms summarize the data by summing and averaging
The fields you selected
The tables you selected
Available table names appear in the pane on the left. For the selected table, the corresponding column names appear in the pane on the right.

If you use an application data source, selecting a table and column internally calls the associated method or field from your object to extract the data.

4 Select Add Entry.

5 Choose Next. Choose a table from the list of tables in the left pane and choose a field from the right pane. Choose OK.

Specify Crosstab Groups reappears.

How to define vertical grouping

1 Choose Rows.

2 To define the vertical grouping based on a formula, enter the formula into the text box under Specify or Update the Grouping Criteria here.

3 Choose Add Entry. Choose Next.

Summaries page appears.

Specifying summaries and miscellaneous data

The key to defining a crosstab is understanding the intersecting value of the groupings. This value is always Table.Column with an aggregate function associated with it. Usually, the aggregate value is a sum. This procedure assumes you completed the steps in “How to define vertical grouping.”

How to specify summaries

1 Choose Innermost Summary Field.
Innermost Summary Field appears.

1. Choose the appropriate tables and fields.

2. Choose an aggregate function, such as sum or count. This function is applied to the intersecting value.

3. Define additional summary criteria.

4. Run your report.
This chapter contains the following topics:

- About using formulas
- Passing values to a function
- Using column references in formulas
- Formatting formula results
- Commonly-used formulas
- Pre-defined function list
- Categories of pre-defined functions
About using formulas

Formulas can be used to manipulate and display report data that would not otherwise be available from the data source. Formulas use values, including other formulas, from any data field within the report. You can format formula results. Expressions are evaluated before they are printed.

As an example of using an expression, the e.Report Designer Java Edition can directly access column names from a data source using the formula capability of the product, so you can execute expressions like this:

```
count(SectionID:TableName.ColumnName)
```

Data from controls can also be used in expressions and formulas. To reference data from other controls specify the control identifier (control ID).

Understanding the Control ID

Each object dropped into a report design has a unique alphanumeric identifier. Having unique IDs is critical when working with formulas that use these unique identifiers.

How to insert a formula

1. Choose Insert ➤ Formula.
2. Drag-and-drop the formula to place it in the report.
   Formula specifications appears to prompt you to enter the formula string.
3. Enter a formula, for example, `sum(controlID)` to sum all the values of the field referred to by controlID.

   For example: if `controlID` contained the data field `sales.amount`, the formula sums, and displays all the sales in the table.
Passing values to a function

A function can have a variable number of arguments and which arguments should be passed to a function varies. Each argument for a function could refer to either a pre-defined function or a user-defined function (UDFs).

The following formulas are valid:

\[
\text{sum(controlID1, controlID2, controlID3)}
\]
\[
\text{sum(controlID, 5)}
\]

Functions support nested function calls. The following function is also valid:

\[
\text{sum(controlID, ave(field001))}
\]

Using column references in formulas

Formulas can contain references to other fields in report designs. They can contain direct references to column names residing in a database. For example: if you have a database with a table called Sales and a column inside it called SaleID, you can directly enter a formula such as this one:

\[
\text{count(SectionID:Sales.SaleID)}
\]

You use the following format to handle direct column references:

\[
\text{function(SectionID:TableName.ColumnName)}
\]

You can nest the following operators inside the formula:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>addition</td>
</tr>
<tr>
<td>-</td>
<td>subtraction</td>
</tr>
<tr>
<td>*</td>
<td>multiplication</td>
</tr>
<tr>
<td>/</td>
<td>division</td>
</tr>
</tbody>
</table>

You can nest the following functions inside the formula:

- sum
- count
- ave
Prefixing the section ID to references

Prefix the SectionID to the names of the columns or controls directly to prevent ambiguity in the report generation process. e.Report Designer Java Edition supports multiple sections within a report. That makes it necessary to specify a unique SectionID for each of the available sections. By default, a section identifier is set to the string: "SEC_00."

Setting the reset criteria

With each formula you must specify when to reset its internal state. A reset criteria communicates this information. By default, the reset criteria of a formula is set to the zone in which it is contained. For example: if you are displaying page numbers, you set the reset criteria of that formula to the Report_Footer. On the other hand, if you have a formula in the page footer that sums the values of a certain field on that page only, you would need to set the reset criteria for that formula to the Page_Footer zone.

Setting date, time and duration functions

e.Report Designer Java Edition supports the use of date, time, and duration functions. You can use arithmetic operators + and - to manipulate any of the Date related functions. For example: the following formula displays the date 20 months in the future:

\[ \text{Now()} + 20\text{MM} \]

The following formula displays the quarter of the year 28 months ago:

\[ \text{Quarter(Now())} - 28\text{MM} \]

MM in the preceding formula is the format that specifies the month.

The following table lists some format specifications for formulas using date/time.

<table>
<thead>
<tr>
<th>Format specifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>Day</td>
</tr>
<tr>
<td>MM</td>
<td>Month</td>
</tr>
<tr>
<td>YY</td>
<td>Year</td>
</tr>
<tr>
<td>HH</td>
<td>Hour</td>
</tr>
<tr>
<td>MN</td>
<td>Minute</td>
</tr>
<tr>
<td>SS</td>
<td>Second</td>
</tr>
</tbody>
</table>
Formatting formula results

e_Report Designer Java Edition supports extensive formatting for two types of formula fields:

- Dates
- Numbers

How to format the output of a formula

1. Double-click the formula control.

   Common Properties appears. You can modify the control’s font, colors, style, and alignment.

2. Choose Format button to the right of the Format field.

   Specify the Format appears.
Choose the None tab to remove any existing format from an expression.

Choose OK to save the format changes.

To specify a formula, choose the browse button to the right of Value in Common Properties.

**How to change the formula for a control**

Formula specification dialog appears.

1. Type your formula in the Type in a formula here: field.
2. Choose OK to save the new formula.
   
   Common Properties appears.
3. Choose OK.
Commonly-used formulas

This section discusses commonly used formulas.

**Today's Date**

```plaintext
now() //
displays today's date in format: Apr-3-1998.
Reset Criteria: Report_Footer
```

**Record Number**

The following function is essentially a simple counter mechanism:
```plaintext
this + 1
```
Every time the zone that contains the formula is printed, the counter is increased by one.
Insert the record number in the detail zone along with other controls:
```plaintext
this + 1 //
displays the record number.
Reset Criteria: Report_Footer
```

**Sum of a field**

```plaintext
sum(controlID) //
displays the sum.
Reset Criteria: Based on need
```

**How to use formulas to insert a page number**

Page number is stored as a formula.
```plaintext
format(this + 1), "####")
```
Insert the page number in either the page header or the page footer:
```plaintext
this + 1 //
displays the page number.
Reset Criteria: Report_Footer
```
This formula displays the page number as an integer. You can change the format by double clicking on the date field and selecting the desired expression.

You can also insert a specialized version of the Page Number field, which displays the page number in the format:

Page XX of YY

XX is the current page number and YY is the total page count.

You can insert a Page Number field in either the Page Header or the Page Footer zone, but not in any other zone.

1 Choose Insert ➤ Page Number.
   A sub-menu appears with options for either Page or PageXofY.

2 Choose one of the page numbering formats.

3 Drag the control and drop it into the report.

---

**Pre-defined function list**

The following table lists the available pre-defined functions. Most take a variable number of functions. The parameters of those that do not take a variable number are based on context. These functions can be called in a nested expression.

<table>
<thead>
<tr>
<th>Trigonometry</th>
<th>Arithmetic</th>
<th>String Related</th>
<th>Statistical</th>
</tr>
</thead>
<tbody>
<tr>
<td>pi()</td>
<td>sumsq()</td>
<td>strlen()</td>
<td>aavedev()</td>
</tr>
<tr>
<td>Degrees()</td>
<td>random()</td>
<td>strcat()</td>
<td>var()</td>
</tr>
<tr>
<td>Radians()</td>
<td>sqrt()</td>
<td>substring()</td>
<td>StdDev()</td>
</tr>
<tr>
<td>SinH()</td>
<td>ceiling()</td>
<td>toUpperCase()</td>
<td>average()</td>
</tr>
<tr>
<td>Sin()</td>
<td>floor()</td>
<td>toLowerCase()</td>
<td></td>
</tr>
<tr>
<td>Cos()</td>
<td>exp()</td>
<td>fixStringLength()</td>
<td></td>
</tr>
<tr>
<td>CosH()</td>
<td>round()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cot()</td>
<td>pow()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACos()</td>
<td>abs()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACosH()</td>
<td>count()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASin()</td>
<td>min()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASinH()</td>
<td>max()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Date**

Date()

**Special**

Format()

**DateTime**

DateTime()

**Time**

now()

DayOfWeek()
Categories of pre-defined functions

This section covers the following categories of pre-defined functions:

- Formatting functions
- Trigonometric functions
- Arithmetic functions
- Statistical functions
- String-related functions
- Date-related functions

Formatting functions

Format()

A commonly used function for formatting output. This function corresponds to the format() function found in java.text package.

For example:

Format("www.sun.com",URL)

changes the format of the string and passes it as an argument that is used to construct a URL.

Trigonometric functions

pi()

Returns value of pi. Does not take any arguments.
For example:

pi()
returns 3.141592653589793
Degrees()
Takes a double as argument and returns the equivalent degrees.
For example:
  Degrees(0.5)
returns 28.64788975654116

Radians()
Takes degrees as argument and returns the equivalent in radians.
For example:
  Radians(30)
returns 0.5235987755982988

SinH()
Takes radians as argument and returns the hyperbolic sine as a double value.
For example:
  SinH(0.5235)
returns 0.5477408488380877

Sin()
Takes radians as argument and returns the sine as a double value.
For example:
  Sin(0.5)
returns 0.479425538604203

Cos()
Takes radians as argument and returns the cosine as a double value.
For example:
  Cos(30)
returns 0.1542544988758405

CosH()
Takes radians as argument and returns the hyperbolic cosine as a double value
For example:
  CosH(30)
returns 5.34323729076226E12
**Cot()**
Takes radians as argument and returns the cotangent as a double value. For example:

\[ \text{Cot}(30) \]
returns \(-0.15611995216165922\)

**ACos()**
Takes radians as argument and returns the arc cosine as a double value. For example:

\[ \text{ACos}(0.5) \]
returns \(1.0471975511965976\)

**ACosH()**
Takes radians as argument and returns arc hyperbolic cosine as a double value. For example:

\[ \text{ACosH}(30) \]
returns \(4.0940666686320855\)

**ASin()**
Takes radians as argument and returns arc sine as a double value. For example:

\[ \text{ASin}(0.5) \]
returns \(0.5235987755982989\)

**ASinH()**
Takes radians as argument and returns arc hyperbolic sine as a double value. For example:

\[ \text{ASinH}(30) \]
returns \(4.0946222433053\)

**ATan()**
Takes radians as argument and returns arc tangent as a double value. For example:

\[ \text{ATan}(30) \]
returns \(1.5374753309166493\)
ATanH()
 Takes radians as argument and returns arc hyperbolic tangent as a double value. For example:
   ATanH(0.5)
returns 0.549306144340549

Tan()
 Takes radians as argument and returns tangent as a double value. For example:
   Tan(0.5)
returns 0.5463024898437905

TanH()
 Takes radians as argument and returns hyperbolic tangent as a double value. For example:
   TanH(0.5)
returns 0.4621171572600098

ATan2()
 This formula is used to convert rectangular coordinates (b, a) to polar (r, theta) and to return the theta component of the point (r, theta) in polar coordinates that correspond to the point (b, a) in Cartesian coordinates. For example:
   ATan2(90, 180)
returns 0.4636476090008061

Arithmetic functions

Sumsq()
 Sum of the square of a given series of values. Values can be passed. For example:
   Sumsq(1, 2, 3, 4)
returns 30.0 or any other double value between 0 and 1.

Random()
 Returns a random double value between 0 and 1. For example:
   Random()
returns 0.29789222540788307
**Sqrt()**

Returns the square root of the number being passed as argument.
For example:

Sqrt(144) returns 12

**Ceiling()**

Returns the smallest double value that is not less than the argument and is equal to an integer.
For example:

Ceiling(123.45) returns 124

**Floor()**

Returns the largest double value that is not greater than the argument and is equal to an integer. For example:

Floor(123.56) returns 123

**EXP()**

Returns the exponential number e, that is, 2.718... raised to the power of a double value. For example:

EXP(2.344) returns 10.42284469719284

**Round()**

Using round with a control ID returns the closest integer to the argument. For example:

Round(123.45) returns 123

Round(123.54) returns 124. Round(FIE000) rounds off all the values of the field, where FIE000 is the control ID of a field having double values.

**pow()**

Returns the value of the first argument raised to the power of the second argument. For example:

pow(2,7)
returns 128 that is, the answer of the expression 2 raised to 7.

**ABS()**

Returns the absolute value of the argument. The absolute value of a number is the same whether it is positive or negative. For example:

```
ABS(-123.45)
```

returns 123.45

**Count()**

Returns the number of values that are passed as arguments. For example:

```
Count(FIE000)
```

returns 8 where FIE000 is the control ID of a field having 8 records.

**Min()**

Returns the minimum value out of a set of values are passed as arguments. For example:

```
Min(FIE000)
```

returns 1 where FIE000 is the control ID of a field having values from 1 through 8.

```
Min(10,12,24,36)
```

returns 10 that is, the least of the numbers passed to the function as arguments.

**Max()**

Returns the maximum value out of a set of values which are passed as arguments. For example:

```
Max(FIE000)
```

returns 8 where FIE000 is the control ID of a field having values from 1 through 8.

```
Max(1,12,24,37)
```

returns 37 that is, the highest of the numbers passed to the function as arguments.

**Sum()**

Returns the sum of all values that are passed as arguments. For example:

```
Sum(FIE000)
```

returns 36 where FIE000 is the control ID of a field having values from 1 through 8.
Sum(1,12,24,36)
returns 73 that is, the sum of the numbers passed to the function as arguments.

Fact()
Returns the factorial value of argument this is passed. For example:
   Fact(4)
returns 24

E()
Returns a double value of e, the base of the natural logarithms. For example:
   E(2.344)
returns 2.718281828459045

LN()
Returns a logarithmic value of the argument. Log value is base 10 logarithm. For example:
   LN(30)
returns 3.4011973816621555

Statistical functions

AveDev()
Average Deviation of a given series of values. For example:
   AveDev(FIE000)
returns 3.5 where FIE000 is the control ID of a field having values from 1 through 8.

Var()
Variance of a given series of values. For example:
   Var(34,45,23,45)
returns 110.91666666666667

stdDev()
Standard Deviation of a given series of values. For example:
   stdDev(FIE000)
returns 2.1213203435596424 where FIE000 is the control ID of a field having values from 1 through 8.
Average()
Average of a given series of values. For example:

Average(FIE000)

returns 4.5 where FIE000 is the control ID of a field having values from 1 through 8.

String-related functions

StrLen()
Returns the number of characters that make up the given string.
The following example returns the length of the corresponding strings for each record:

StrLen(FIE001)

If formula is inserted in the detail zone, where FIE000 is the control ID of a field having string type values:

StrLen("Actuate")

returns 7, the length of the string passed in the function as parameter.

StrCat()
Concatenates two strings given as arguments and returns a string with second string appended to the first. For example:

StrCat(FIE000,FIE001)

returns the concatenated string of the corresponding strings of two columns for each record if the formula is inserted in the detail zone, where FIE000 and FIE001 are the control IDs of data fields.

StrCat(FIE000,"acj")

returns the concatenated string of the corresponding string for each record and the string(acj) if the formula is inserted in the detail zone, where FIE000 is the control ID of a field having string type values.

StrCat("Java"," Report"," Designer")

returns the concatenated string of all the three strings passed in the function as parameters.

Substring()
Returns part of a given string. For example:

Substring("Apple", 0, 1)

returns an "A". The first argument is the string to be processed; the second is the start of the index from where the substring should be extracted, and the
third specifies the number of characters to be extracted starting from the given index.

**toUpperCase()**
Takes a string and returns a string with all characters converted to upper case. For example:

```plaintext
toUpperCase("Apple")
```
returns: APPLE

**toLowerCase()**
Takes a string and returns a string with all characters converted to lower case. For example:

```plaintext
toLowerCase("Apple")
```
returns: apple

**FixStringLength()**
Returns the string of some specified size. It takes two parameters: string and integer in the same order. For example:

```plaintext
FixStringLength("ReportDesigner",6)
```
returns the string "Report" that is, the first ten characters from the string passed to the function as an argument.

```plaintext
FixStringLength(FIE000,5)
```
returns the first five characters of the string of the field (FIE000) for each record if the formula is inserted in the detail zone, where FIE000 is the control ID of a field having string type values.

```plaintext
FixStringLength(FIE000,FIE001), if inserted in the detail zone,
```
returns the corresponding number of characters from the field (FIE000) as that of the numbers in the second field (FIE001). In this example, if FIE001 is the control ID of a field having values from 1 through 8, the first record displays the first character from the string of the corresponding value of the field (FIE000), the second record displays two characters and so on.

FIE000 is the control ID of a string; FIE001 is the control ID of an integer.
The record number 8 has the category name “Seafood” (FIE000) (which is 7 characters long) and category ID as “8.” In such cases where the length of the string is less than the second argument passed to the formula, with blank spaces at the end of the displayed string. This formula return value will be ‘Seafood’ with one space at the end.

### Date-related functions

**Date()**

Returns the date. It takes a string argument and converts it into a date. For example:

```
Date("07/04/2000")
```

returns 2000-07-04 depending on the format that is specified.

**DateTime()**

Returns today’s date in specified format. For example:

```
DateTime("07/04/2000 4:20:05")
```

returns 2000-07-04 the date passed in as a parameter.

**Time()**

Accepts a string date and returns time. For example:

```
Time("07/04/2000 4:20:05")
```

returns 4:20:00 the time passed as a parameter.

**Now()**

Returns today’s date in specified format. For example:

```
Now()
```

returns July 04, 2000 the current date/system date.

**DayOfWeek()**

Returns the day of the week of specified date in specified format. For example:
DayOfWeek(now())
returns 2 if the day of week of current/system date is Monday.

WeekOfYear()
Returns the week of the year of the specified date in the specified format. For example:
WeekOfYear(now())
returns 28 2000 the week number and year of current/system date.

Month()
Takes a date as a parameter and returns the month in a numeric format. For example:
Month(now())
returns 07 the month in numeric format of current/system date.

MonthString()
Takes a date as a parameter and returns the month in a string format. For example:
MonthString(now())
returns July the month in words of current/system date.

Quarter()
Returns the quarter of the year of specified date in specified format. For example:
Quarter(now())
returns 2000-3 the year and Quarter number of current/system date.

Year()
Returns the year of the specified date in specified format. For example:
Year(now())
returns 2000 the year of current/system date.
Chapter 9, Conditional formatting

This chapter contains the following topics:

- About using scripts
- Understanding restrictions
- Formatting properties
- Operators
- Commonly used scripts
About using scripts

Users often need to handle report data based on certain rules. A simple example is the need to display a number in red if its value is less than 0. Such a format not only makes the report more readable, but adds value to the business data being presented. In e.Report Designer Java Edition, scripts can handle report data.

A script is a statement that follows certain syntactic conventions. A script is a global feature not specifically associated with any control or a zone. A script consists of two portions, the condition portion and the action portion. A script can have reference to controls and zones in the condition or the action part of the script. A script can also utilize any function available in the expression module.

An object-oriented approach has been used as the mechanism for referring to a control property. The syntax for a simple script is:

```
IF (<Condition>) { <Action(s)> }
```

For example,

```
IF (CtrlID.Value == 5) { CtrlID.FillColor = {255,0,0} }
```

The condition in the above example is CtrlID.Value == 5

This condition can be interpreted as for all Ctrlld controls whose value equals five. The Value property and other properties are described in detail later. This property depends on the context of the control with which it is associated. If CtrlID is a data field, formula or a label, then for every instance of this printed control, the script is processed to determine if its value is 5.

You must enclose the right side in double quotation marks to specify a string literal for LABELs or string data fields.

Supported condition combinations

The condition can contain a standard or a user-defined formula. The following format is the basic format:

```
IF (<LHS> <OPERATOR> <RHS>)
```

For example,

```
IF (CtrlID.VALUE == 5)
```

Combinations of the basic format are also supported.

```
IF (FUNC1(<LHS>) <OPERATOR> FUNC2(<RHS>))
```

For example,

```
IF (strlen(CtrlID1) > 2 + 3)
```
A combination of several expressions concatenated with logical && and || operators is supported:

\[
\text{IF (} <\text{LHS1}> <\text{OPERATOR1}> <\text{RHS1}> \quad && \quad <\text{LHS2}> <\text{OPERATOR2}> <\text{RHS2}> \quad || \\
\text{<LHS3}> <\text{OPERATOR3}> <\text{RHS3}>\text{)}
\]

For example,

\[
\text{IF (CtrlID1.Value}=[07-22-00] \quad && \quad \text{CtrlID2.Value}="\text{Harry}" \quad || \quad \text{strlen(CtrlID3)} > 6)
\]

When a formula is used, the parameter in the formula should not contain the Value property. The following example is not correct:

\[
\text{strlen(CtrlID3.VALUE)} > 6
\]

This script example is not associated with any particular control or zone since it is dependent on values of three different control IDs. Enclose the RHS value in the appropriate delimiters, depending on the LHS expression’s data type.

Use the following delimiters:

IF (IntegerFieldID.Value > 6)
IF (strlen(StringFieldID) > 6)
IF (StringFieldID.Value == "Beverages")
IF (DateFieldID.Value == [01-01-2000])

The delimiters in these examples are quotation marks and brackets. Delimiters are not required for numerical expressions on the LHS. For strings, use double quotation marks. Use brackets for date values.

Some more complex examples of supported scripts are of the following format:

\[
\text{IF (} <\text{CONDITION1}> \quad \{ \quad <\text{ACTION1}> \quad \} \quad \text{ELSE IF (} <\text{CONDITION2}> \quad \{ \quad <\text{ACTION2}> \quad \}
\]

For example,

\[
\text{IF (CtrlID1.Value}=[07-22-00]) \quad \{ \quad <\text{ACTION1}> \quad \} \quad \text{ELSE IF (strlen(CtrlID3)} > 6) \\
\quad \{ \quad <\text{ACTION2}> \quad \}
\]

Another supported condition uses the following format:

\[
\text{IF (} <\text{CONDITION1}> \quad \{ \quad <\text{ACTION1}> \quad \} \quad \text{ELSE IF (} <\text{CONDITION2}> \quad \{ \quad <\text{ACTION2}> \quad \} \\
\quad \text{ELSE (} <\text{ACTION3}> \quad \}
\]

For example,

\[
\text{IF (CtrlID1.Value} = 25) \quad \{ \quad <\text{ACTION1}> \quad \} \quad \text{ELSE IF (strlen(CtrlID3)} > 6) \\
\quad \{ \quad <\text{ACTION2}> \quad \} \quad \text{ELSE (} <\text{ACTION3}> \quad \}
\]
Restrictions

The following restrictions apply:

- The condition can only contain Control IDs that represent a data field, a formula, a label, or a group header.
- The Control ID used in the condition can be checked only against its value property. In this case, the Control ID was CtrlID and the read-only property was Value.

A group header Value property represents the value of evaluated grouping criteria. If a group header has its grouping criteria set as a Group by Column and the grouping column set as TableX.ColumnY, then the script executes every time this zone is processed. The condition is evaluated based on the current grouping criteria evaluated value.

For example, the FillColor property sets the background color for objects including fields, formulas, labels, or zones that are represented by their Control ID. In this case, it sets the background color to the object represented by CtrlID.

```
CtrlID.FillColor = {255,0,0}
```

This action can be interpreted as follows:

- Set its background fill color to R=255, G=0, B=0 (RED)

Supported action combinations

The action contains one or more action expressions separated by a space, a tab, or a new line.

An expression can contain the following:

```
IF (CONDITION) { CtrlID1.Property1={PropertyValue1}
                     CtrlID2.Property2={PropertyValue2} }
```

Or the expression can be written as follows:

```
IF (CONDITION)
{
    CtrlID1.Property1={PropertyValue1}
    CtrlID2.Property2={PropertyValue2}
}
```

For example, this more readable format is preferred:

```
IF (GH_00.Value == 2)
{
    CtrlID1.FillColor = {255, 0, 0}
    CtrlID2.Prefix = {Minus}
    CtrlID3.Font = {Dialog, BOLD, 24}
}
```
Rules

The left side, LHS of an action must contain a valid Control ID that represents any control in the report design. This Control ID can represent a Data Field, Formula, Label or any Zone. If you specify an invalid Control ID, the script editor generates an error and does not accept the changes made to the script. Ensure that the property specified in the LHS is a valid writable property.

- The assignment operator = must be used to set an action LHS property to the RHS value. No other operator is supported in an action clause.
- The RHS of an action expression containing the value to be assigned to the LHS property must be enclosed in curly brackets ({ and }). No delimiters are used in the RHS expression values.

Understanding restrictions

The following restrictions apply when writing scripts:

- One or more action clauses can be applied only to data fields, formulas, labels, and zones. Applying actions on other type of controls has not been validated and can result in an unpredictable behavior in the output.
- Ensure that you use RHS value for a property in the correct format corresponding to the property type.
- The Value for a ControlID cannot be set in the action clause due to cross references.

The script you create is not case sensitive. You can use any combination of lower, upper, or mixed case. This option includes Control IDs, property names and the IF, ELSE keywords. Any number of spaces, tabs and new line characters can be embedded in a script to make it more readable, if required. This inclusion does not affect the parsing and execution of a script.

How to create a script

Follow these steps to create a script.


   If the template does not contain any scripts, Script Editor appears.
2 Specify the ID of the new script. Choose OK.

3 Enter script details into the panel on the right. Choose Save Changes.

   Internal validation occurs to verify the script.

   Your new script added to the list of available scripts

If your template already contains scripts, you see a list of scripts being used in
the template. You can rename or delete existing scripts.

You create a new script by choosing New Entry.
Using the syntax preceding, you can create scripts. e.Report Designer Java Edition performs some basic validations for syntax. It cannot validate the logic of your script. It assumes that the script is correct.

Creating Zone-level scripts

In Zone-level scripts you can have the conditions based on the GroupHeaders values. Depending on the GroupHeader value you can change the appearance of the detail zone or you can hide the detail zone.

For example,

\[
\begin{align*}
\text{IF (GH}_00\text{.VALUE > 100 ) \{ DTLZONE.VISIBLE = {false} \quad \text{GH}_00\text{.VISIBLE = {false}} \}} \\
\text{IF (GH}_00\text{.VALUE == "Beverages") \{ DTLZONE.FILLCOLOR = {255,255,0} \}} \\
\end{align*}
\]

Handling different data types

The scripting module handles the following data types in different ways.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Handled as . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers</td>
<td>Integer and double</td>
</tr>
<tr>
<td>Dates</td>
<td>&quot;mm/dd/yy&quot; (any java supported date format)</td>
</tr>
<tr>
<td>Strings</td>
<td>&quot;xxxxyyyyzzz&quot;</td>
</tr>
<tr>
<td>Color</td>
<td>rrr/ggg/bbb</td>
</tr>
</tbody>
</table>

Chapter 9, Conditional formatting 213
### Formatting properties

This section lists properties that you use in a condition along with a brief description.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td>Represents the control's or zone's group header current value. Read only.</td>
</tr>
<tr>
<td>FILLCOLOR</td>
<td>Changes the background color of the control or zone. Write only.</td>
</tr>
<tr>
<td>TEXTCOLOR</td>
<td>Changes the foreground color of the control or zone. Write only.</td>
</tr>
<tr>
<td>VISIBLE</td>
<td>Determines whether or not a control or zone should be displayed. Write only.</td>
</tr>
<tr>
<td>STRIKE</td>
<td>Strikes out the control text. Write only.</td>
</tr>
<tr>
<td>PREFIX</td>
<td>Prefixes the specified text to the control's value. Write only.</td>
</tr>
<tr>
<td>POSTFIX</td>
<td>Postfixes the specified text to the control's value. Write only.</td>
</tr>
<tr>
<td>FONT</td>
<td>Changes the font used to display the control's value. Write only.</td>
</tr>
</tbody>
</table>

### Operators

This section lists the operators that you can use in a condition.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>Tests for an equality condition.</td>
</tr>
<tr>
<td>!=</td>
<td>Tests for an inequality condition.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Tests for a greater than condition.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Tests for a greater than or equal to condition.</td>
</tr>
</tbody>
</table>
The last three operators are used to compare the numerical and string data, as illustrated by the following examples:

\[
\begin{align*}
\text{IF (CONTROLID01.VALUE < 3)} \\
\text{IF (CONTROLID02.VALUE == "Beverages")} \\
\text{IF (CONTROLID01.VALUE != 5)}
\end{align*}
\]

**Arithmetic operators**

A script supports the following arithmetic operators.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Plus</td>
</tr>
<tr>
<td>-</td>
<td>Minus</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
</tr>
<tr>
<td>%</td>
<td>Modulus or remainder after division</td>
</tr>
</tbody>
</table>

**Logical operators**

A script supports the following logical operators:

- || or || or OR can be used to perform a logical OR between two condition subexpressions.
- & or && or AND can be used to perform a logical AND between two condition subexpressions.
- ! or NOT can be used to perform a logical NOT when prefixed before an AND or OR operator.

With the help of arithmetic and logical operators you can embed complex expressions in the script.

For example,

\[
\text{IF ((CONTROLID01.VALUE \% 2) == 0 OR CONTROLID02.VALUE CONTAINS 'Con')}
\]
String operators

A script supports the following string operators.

- **CONTAINS**

  This operator is used for testing whether or not the control value contains the specified substring or character. CONTAINS is not case sensitive.

  For example,
  
  ```
  IF (CONTROLID01.VALUE CONTAINS 'b')
  IF (CONTROLID01.VALUE CONTAINS 'beverages')
  ```

- **STRINGLENGTH**

  This operator is used for comparing the string length. You must enclose the integer value in single quotation marks when you use this operator.

  For example,
  
  ```
  IF (CONTROLID01.VALUE STRINGLENGTH '7')
  IF (strlen(CONTROLID01) == 7)
  ```

- **STLEN_GT**

  This operator determines whether the string length is greater than the specified value. For using this operator, you must enclose the integer value in single quotes as above.

  For example,
  
  ```
  IF (CONTROLID01.VALUE STLEN_GT '7')
  IF (StrLen (CONTROLID01) > 7)
  ```

- **STLEN_LT**

  This operator determines whether the string length is less than the specified value. You must enclose the integer value in single quotation marks when you use this operator.

  For example,
  
  ```
  IF (CONTROLID01.VALUE STLEN_LT '7')
  IF (StrLen(CONTROLID01) < 7)
  ```

- **STARTSWITH**

  This operator is used for testing whether or not the control value starts with the specified string or character. STARTSWITH is not case sensitive.

  For example,
  
  ```
  IF (CONTROLID01.VALUE STARTSWITH 'b')
  IF (CONTROLID01.VALUE STARTSWITH 'Con')
  ```

- **ENDSWITH**
This operator is used for testing whether or not the control value ends with the specified string or character. ENDWITH is not case sensitive.

For example,

\[
\text{IF (CONTROLID01.VALUE ENDSWITH 't')}
\]

\[
\text{IF (CONTROLID01.VALUE ENDSWITH 'ges')}
\]

The only operator that can be used in an action of a script is the assignment operator (=).

For example,

\[
\text{IF (CONDITION) \{ CONTROLID01.FILLCOLOR=(255,0,0) }
\]

\[
\text{CONTROLID02.VISIBLE=(false) } \}
\]

---

**Commonly used scripts**

This section contains typical scripts that show how to handle different data types, how to refer to controls and zones, and use various operators:

- Change a control's text color to red if its value is less than 0.
  \[
  \text{IF (CONTROLID.VALUE < 0) \{ CONTROLID.TEXTCOLOR = \{255,0,0\} } \}
  \]

- If a control's value is null, do not display it.
  \[
  \text{IF (CONTROLID.VALUE == 'null') \{ CONTROLID.VISIBLE = \{False\} } \}
  \]

- Change the font for a control if its value is odd.
  \[
  \text{IF ((CONTROLID.VALUE % 2) == 0) \{ CONTROLID.FONT = \{COURIER, BOLD|ITALIC, 16\} } \}
  \]

- Change the background color of the detail zone to red if the control value is even.
  \[
  \text{IF ((CONTROLID.VALUE % 2) != 0) \{ DTL_00.FILLCOLOR = \{255,0,0\} } \}
  \]

- If a group header zone's value is "Beverages," do not print the details of that group. This feature is conditional printing.
  \[
  \text{IF (GROUP_HDR.VALUE = "Beverages") \{ DTL_00.VISIBLE = \{false\} } \}
  \]

- If a group header zone's value is negative, display the details of that group in red color. This feature is conditional printing.
  \[
  \text{IF (GROUP_HDR.VALUE < 0) \{ DTL_00.FILLCOLOR = \{255,0,0\} } \}
  \]
Symbols

! operator 215
!= operator 214
# sign 120
& operator 215
&& operator 209, 215
< operator 215
<= operator 215
= operator 211
== operator 214
> operator 214
>= operator 214
@ symbol 118
| operator 215
|| operator 209, 215

Numerics

3-part SQL 122

A

abs function 200
absolute paths 55
absolute values 200
access controls 26
accessing
  color palette 31, 160
databases 73
Java objects 82
ACJDemo report 3
ACJDesigner.properties file 163
ACJjdbc.bat
  running on non-Windows platforms 69
  running on UNIX platforms 69
  running on Windows platforms 67
ACJMain class 80
ACJMain command line options 81
ACos function 197
ACosH function 197
action expressions 210, 211
form 211
actions
  defined 210
  entering in conditional expressions 209
Actuate LRX xiii
Actuate product summary x
Actuate web site xv
Add Edge command 176
adding
  asynchronous grid controls 176
  background colors 29
  borders 29, 30, 160
  controls 6, 23, 59
  current dates 6
dates 49, 50
formulas to reports 188
graphics objects 53, 54
graphs 168
hyperlinks 58
images 54
labels to controls 46
page breaks 33
stored objects 143
Address Book of Recipients property 163
Advanced e.Reporting Server xii
Advanced e.Reporting Server Progress
  Edition xiii
AdvMultiSections.jod 116, 118
aggregate functions 185
alignment
  asynchronous grid controls 177
  controls at top edges 29
currency 59
text 59
alignment tools 44, 59
analysis tools (third-party) x
analyzing data 2
analyzing report designs 22
AND keyword (SQL) 107
AND operator 215
applets 2
Application data command 82
application data sources
  creating crosstabs with 184
  specifying 123
application programming interfaces  x
applications 2
  as data sources 64, 82
  query processor for 122
  setting text qualifier for external 163
arguments 189
  See also parameters
arithmetic functions 198
arithmetic operators 215
ASC keyword (SQL) 107
ascending sort order 107
ASCII data source wizard 74
ASCII data sources
  connecting to 74
  overview 64
  retrieving records from 163
ASCII files 64
ASCII formats 156
ASCII-CSV file formats 2
ASin function 197
ASinH function 197
assignment operator 211
asymmetrical grids 175
  creating 176
AsyncGrid command 176
AsyncGrid control 175
  adding 176
  aligning 177
ATan function 197
ATan2 function 198
ATanH function 198
attributes
  changing dynamically 62
  setting from toolbar 44
Attributes dialog 59
Author property 159
auto update options 98
AutoFormatWizard command 28
automatically running reports 40
automating reporting tasks xi
avedev function 201
average deviation 201
average function 202
averages 202
axes values
  defining 173, 174
  overview 173
  retrieving data for 170
B
background color icon 44
background colors
  applying to labels 49
  applying to reports 29
  setting in scripts 210
  specifying 29, 49, 157
backgrounds 53, 160
batch files 40, 80
blank IDs 61
Blank Report option 36
blank reports
  creating 36
  creating crosstabs from 181
  creating mailing labels with 178
  creating tabular reports with 177
  viewing zones in 23
bookmarks 161
border styles 30
borders 29
  adding 30, 160
  selecting line styles for 31
  setting graph 172
Borders property 160
bounds 60
Bounds command 60
Bring In Front of Others command 61
building reports
  from sample report 1, 3
  planning process described 18
C
Cache property 161, 162
calculations 24
case conversions 203
ceiling function 199
cells (asymmetrical grids) 175, 177
center alignment icon 44, 59
Change Label’s Caption dialog 46
changing
  data types in stored procedures 138
  default date formats 128
defaults 152
graphics objects 54
changing (continued)
  label properties 47
  page number formats 194
  page orientation 160
  queries 131
  report margins 160
  reports 6
  reports at runtime 62
  RichText controls 57
  user interface 152
channels xi, xii
Character Set property 161, 162
character sets 161, 162
character strings 216
  See also strings
Chart template 4
Chart/Graph command 169
Chart/Graph Properties dialog
  creating y-axis 171
  selecting graph type 169
charts. See graphs
ChartTemplate.acj 4, 5
choosing
  border styles 30
  database fields 38, 94
  database tables 38, 71, 83
  database tables for crosstabs 184
  fields for graphs 170
  multiple controls 28
  report styles 35
code
  configuring connections with 82
  cutting and pasting 85
  location of source 21
color palette 31, 160
colors
  adding background 29, 157
  applying to labels 48, 49
  applying to rectangles 54
  applying to report pages 29
  preserving 161, 162
  setting fill 45, 46
  setting graph 172
  setting in scripts 210
column headings
  adding to result sets 143
  database fields as 34, 37
  referencing in formulas 112
  with lowercase letters 155
columnar reports
  creating 178
columns
  See also fields
  as links to tables 102
  defining for crosstabs 181, 183
  defining for result sets 143
  embedding in RichText controls 57
  referencing 122, 189
  selecting for crosstabs 184
  selecting from application data sources 83
  specifying as fixed-length 76
  specifying for query results 137
  suppressing duplicate 61, 62
  uniquely referencing 122
command line options
  ACJMain 81
  startup 40
comments 85
Common Properties dialog 51, 191
company names and logos 24
comparing data 180, 215
comparison operators 215
compressing files 161
concatenation 202, 209
conditional expressions
  actions in 210, 211
  adding to scripts 208, 210
  entering operators in 214
  formatting properties for 213
Conditional Formatting command 211
conditions 107, 208
  restrictions for 210
Confidential text string 160
configuration parameters 81
configurations
  application data sources 82
  JDBC data sources 81, 86
  JDBC drivers 81
Connect command 70
Connecting Application data dialog 82
connection parameters 74
connections
  ASCII data sources 74
  configuring JDBC 81
  defining 64
  establishing 70, 156
  JDBC data sources 73, 78
  ODBC data sources 73
  overview 64, 80
  remote databases 74
  sample database 67
  setting up in source code 82, 85, 87
  specifying user name and passwords 81
  testing 71
constants 134, 141
  entering 138
containers 23
CONTAINS operator 216
contents page
  creating 9, 161, 162
  designing 11, 22
  disabling display of 81
  displaying on Preview page 157
  drilling down through 10
  setting defaults for 157
context-sensitive help. See help
Control Bounds dialog 60
Control Defaults page (Options) 153
Control ID command 61
control IDs
  entering in conditions 208
  entering in expressions 188
  entering in scripts 211
  in scripts 208
  overview 61
  setting master section 118
  specifying 61
  updating for stored procedures 142
cross Tab Report command 181
controls
  adding labels 46
  adding to designs 6, 23, 59
  aligning top edges 29
  aligning with cell edge 177
  as data source for graphs 173
  embedding in RichText controls 57
  for asymmetrical grids 175
  formatting dynamically 62
  formatting options for 42, 44
  hiding 26
  identifying 61
  linking input parameters to 141
  linking to output parameters 139
  moving 28, 153
  overlapping 61
  placing bounds on 60
  redisplaying 27
  referencing 190, 208
  resizing 57, 58, 153
  selecting multiple 28
  setting default spacing between 152
  setting properties for 45
  testing string values in 216
  types listed 42
  variable dimensions in 57
coordinates 61
Cos function 196
CosH function 196
Cot function 197
counter 193
Create a Crosstab Report wizard 181, 182
Create a Label Report dialog 179
creating
  batch files 40
  columnar reports 178
  contents page 9, 161, 162
  crosstab reports 180
  forms 175
  grouped reports 108
  joins 102, 105
  mailing labels 178
  master-detail reports 113, 117
  parameterized reports 124, 131
  queries 90, 96, 122
  query import sections 99
  reports 6, 34
  reports offline 123
  running headers 33
  scripts 211, 212, 213
  stored procedures 140, 143
  tabular reports 177
templates 36
222 Designing e.Reports Java Edition
crosstab reports
  creating 180
  defining columns and rows 181
  defining groups 183, 184
  overview 181
  retrieving data for 184
CSV file formats 2
CSVSource.txt 4
CtrlID variable 208
currency controls 59
Current Export Properties dialog 14
customizing
  designer functions 152
  report properties 158

D
data
  adding to crosstabs 184
  adding to graphs 170
  analyzing 2
  comparing 180, 215
  embedding 57
  entering in forms 175
  grouping 108, 112
  grouping into logical sets 25
  importing 74
  in hierarchical format 113
  plotting multiple series of 174
  retrieving specific formats 120
  sorting 125
  uniquely identifying 95
  with varying lengths 57, 58
Data Browser 93, 104
data extraction 90, 95, 134
Data Field command 93, 94
data fields 24, 93
  user-defined options for 153
data filters 107, 125
Data menu 64
data source options 64
data sources
  configuring application 82
  connecting to 70, 156
  creating crosstabs with 181
  embedding columns from 57
  extracting data from 90, 95, 134
  overview 64
  retrieving records from ASCII 163
  saving 156
  selecting tables from 71
  specifying for graphs 173, 174
  supported types 2
  supporting stored procedures 139
  swapping 123
data types
  constants and 138
  entering in scripts 213
  returning incorrect 137
Database Connection Wizard 78
database connections. See connections
database drivers 73
databases xiii
  accessing 73
  as data sources 2
  creating ODBC entries for 64
  displaying schemas for 71
  fields as column headings 34, 37
  fields as page headings 33
  fields as report headers 33
  referencing fields in 189
  remote 74
  sample 3, 64, 67
  selecting tables 38, 71, 83
Datasource Property Sheet 70
date controls 42
date formats 49, 50
date function 204
date functions 190, 193, 204
dates 42
  adding to headers 6
  adding to reports 24, 49, 50
  changing default formats 128
  entering in conditional expressions 209
  entering in expressions 190
  formatted string substitution for 120
  formatting 122, 191
  reformatting 50
  returning current 193, 204
  setting delimiters for 131
  verifying formats for 120
datetimestamps 159
datetime function 204
DayOfWeek function 204
debug mode 157
debug options 157
Debug page (Options) 157
Default File Formats property 163
default values 124
defaults
  changing 152
date formats 128
language 81
margins 14
defining
  connections 64, 82, 85, 87
  filters 107
  relationships 104
Degrees function 196
deleting
  parameter fields 127
delimiter character
control IDs 118
query string substitution 120
delimiters 76, 131, 163
  numerical expressions and 209
delimiting string values 120, 122
Demo folder xv
dependencies 113
DESC keyword (SQL) 107
descending sort order 107>Description property 159
design phase 18
designer 2
customizing 152
  formatting toolbar 44
loading templates 4
panes described 6
resizing panes 10
sections in 22
zones in 23
designer window 19
designing report layouts 29
designing reports 8
  overview 18, 22
  user-defined options for 152
  with asymmetrical grids 175
designing Table of Contents 11, 22
detail groups 25
detail sections
  creating 114
  defining dependencies 113
  inserting data into 118
details zone
  adding images to 54
  creating scripts for 213
  overview 23, 24
Detected Drivers list 70
deveiation 201
dimensions (data sources) 181
Dimensions property 160
directories
  absolute vs. relative paths 55
  specifying image paths 153
directory
  setting workspace defaults 154
Display Datasource Schema command 71
displaying
  border colors 32
database schemas 71
  formatting options 42, 44
  online documentation xvi
  query statements 90
reports 10, 21, 26
Table of Contents 9, 81, 157
user-entered values 132
zones 23
documentation
  online xvi
  syntax conventions xx
  typographical conventions xix
Drill Down Options command 12
drilling down 25
designing reports for 11, 13
  in Preview page 10
  overview 10
driver specifications 70
drivers
  availability of ODBC 73
  configuring JDBC 81
  displaying registered JDBC 70
  getting database 73
  specifying JDBC 81
  stored procedures and 138, 139
duration 190
dynamic formatting 62
dynamic HTML. See DHTML
E
e function 201
e. Analysis application xiii
  e. Report Designer xi
  e. Report Designer Java Edition xi
    changing user interface for 152
    creating startup scripts for 80
    customizing 152, 158, 159
    overview xiv, 2, 90
    reading data sources with 64
    starting 3, 40
  e. Report Designer Java Edition Viewer 2
  e. Report Designer overview xv
  e. Report Designer Professional x
  e. Reporting Server xi
  e. Reporting Server Java Edition xii
  e. Reporting Server Progress Edition xii
  e. Reporting Suite ix
    product summary x
  editors
    report 6
    rich text 56
  EJB data sources 64
  EJB servers 2
  email applications 154
  embedded modules 2
  embedding data 57
  embedding graphs 175
  enabling Rich Text Editor 56
  encoding 156
  End User Desktop xiii
  ENDSWITH operator 216
  Enter Parameters Values dialog 126
  Enterprise Java Beans. See EJB
  equality 214
  errors 157
  establishing connections 70, 156
  executing reports 40
  exp function 199
  Explorer. See Report Explorer
  exponentiation 199
  export properties 161
  Export Properties dialog 160
  exporting reports 14
  Expression command
    setting dates with 50
  expressions 24
    containing references 189
    defining groups with 112, 183, 184
  delimiters in 209
  embedding in RichText controls 57
  embedding in scripts 215
  entering conditions in 208, 210
  entering format specifiers in 190
  entering operators in 214
  formatting results 191
  inserting 188
  overview 188, 193
  removing formats from 192
  resetting 190
  supported action combinations for 210
  external applications 163

F
  F1 key xvii
  fact function 201
  factorial 201
  fields
    See also columns
    adding labels to 153
    adding to queries 93
    adding to reports 24
    as column headings 34, 37
    as page headings 33
    creating for parameterized reports 124
    defining relationship to tables 104
    referencing in columns 189
    referencing names in formulas 112
    removing for parameterized reports 127
    selecting 38, 94
    selecting for graphs 170
  file formats
    ASCII data sources 74
    email applications 163
    graphs and 175
  files
    compressing 161
    creating batch 40
    generating HTML 161
    linking to HTML or PDF 58
    loading from URLs 36
    loading image 55
    setting up text 64
    supported types 2, 14
  Fill Color command 54
fill colors 29, 45, 46, 54
FillColor property 210, 214
filters 107, 125
fixed-length columns 76
fixStringLength function 203
flat files 24
floor function 199
Font property 214
font size attribute 44
font styles 48
Font type attribute 44
Font/Colors command 46, 47
fonts
  preserving 161, 162
  setting for graphs 172
  supported 48
footers
  adding to reports 26
  inhibiting 32
  numbering pages with 32
  printing 26, 156
  zones specific to 23
foreground color icon 44
foreground colors 48
  applying to controls 44
foreign keys 102, 104, 181
form letters 57, 175
Form report property 159
format function 195
format/specifiers 190
formatted string substitution 120
formatting
  controls dynamically 62
  dates 50, 122, 191
  graphs 172
  output 195
  strings 122
  text 56
  time values 191
formatting options
  displaying 42, 44
  overview 42
  setting template 156
formatting properties 213
formatting toolbar 44
forms 159

creating 175
Formula command 188
formula controls 191
Formula specification dialog
  resetting record counts 52
  setting dates with 50
formulas
  as fields 24
  containing references 189
  defining groups with 112, 183, 184
  delimiters in 209
  embedding in RichText controls 57
  embedding in scripts 215
  entering conditions in 208, 210
  entering format specifiers in 190
  entering operators in 214
  formatting results 191
  inserting 188
  overview 188, 193
  removing formats from 192
  resetting 190
  supported action combinations for 210
ForNorthwind directory 67
frames 23
FROM clause (SQL) 97
functions
  adding to reset criteria 190
  aggregate 185
  categorized 195
  entering in scripts 208
  nesting 189
  overview 134
  passing values to 189
  predefined listed 194
  samples of 135

G
gap 152
General page (Options) 152
generating reports 8
  with specific criteria 123
generic queries 96
GIF images 54
GotoPage command 8
graphical user interface 152
graphics objects
See also images
adding 53, 54
changing 54
graphs
adding 168
defining axes values for 173, 174
formatting 172
overview 168, 173
plotting multiple data series 174
retrieving data for 170
saving 175
sending to output devices 175
setting properties 172
specifying type 169
greater than operator 214
greater than or equal to operator 214
grids
creating asymmetrical 175, 176
displaying report 153
sizing 176
group footer zone 23
group header zone
adding page headings to 33
adding to reports 109
formatting options for 44
overview 23
selecting fields for 38

group headers
repeating 33
restrictions for grouping criteria in 210
grouped reports
creating 108
crosstabs as 181
grouping criteria 11, 97, 108
for nested groups 112
restrictions for 210
Grouping Criteria command 112
grouping data 108, 112
grouping details 25
grouping element
crosstab reports 181
rectangles as 53
groups
applying reset criteria to 51
contained in multiple sections 113
creating 108
creating with formulas 112, 183, 184
defined 108
defining in crosstabs 183, 184
displaying record counts for 51
nesting 25, 111, 112
tracking number of records in 50
GUI
customizing 152
headers
adding current date 6
adding to reports 24
creating running 33
database fields as 33
inhibiting 32
numbering pages with 32
repeating group 33
zones specific to 23
headings 56
help xvii–xviii
usage overview xvii
help topics xvii
hiding
page numbers 32
parts of a report 26
Table of Contents 9
zones 23
high-resolution HTML 2
horizontal gap 152
HTML files 2
embedding graphs in 175
generating 161
linking to 58
HTML reports
displaying Table of Contents in 161
preserving colors for 161
preserving fonts for 161
viewing 10
hypertext links
adding to reports 58
identifiers 61, 188
image controls 42
image writer 175
images 42
  adding 54
  as backgrounds 160
  loading 55
  setting relative paths for 153
Images tool 54
Import ASCII Data command 74
import query sections 99
importing
  data 74
  SQL queries 99
–Indexed option 81
inequality 214
inhibiting headers and footers 32
input 123, 130
input parameters 134, 141
Insert Grouping Above command 111
Insert GroupingAbove command 109
installation requirements 64
installing online documentation xvii
intranets 74

J
Java applications 2
Java Database Connectivity. See JDBC
Java objects 82
Java Product Demo file 3
Java Runtime Environment 64
Java Virtual Machines 2
Java web reporting tool 2
jConnect driver 138
JDBC Code Wizard 86
JDBC Connectivity options 79
JDBC data command 86
JDBC data sources
  configuring 81, 86
  connecting to 70, 73, 78
  specifying 123
  support for 64
  testing connections 3
JDBC drivers
  configuring 81
  displaying registered 70
  specifying 81
  stored objects and 142
JDBC URLs 73, 74, 81
–JdbcDRIVER option 81
JDBC-ODBC bridge 64, 73
–JdbcPASSWORD option 81
–JdbcURL option 81
–JdbcUSERNAME option 81
.JOD filename extensions 3, 20
joins
  creating 102, 105
  removing 104
JPEG images 54
JRE 64
justification 44

K
key columns 102

L
label controls 42
label format menu 47
label formats 179
label reports 178
Label Wizard 179
labels
  adding to controls 46
  adding to data fields 153
  as web links 58
  changing properties of 47
  overlapping 61
  resizing 47
  setting colors 48, 49
  setting properties for 159
landscape orientation 13
–Language option 81
large reports 10, 108, 161
launching reports 40
layout pane 6
layouts. See page layouts
left alignment icon 44, 59
less than operator 215
less than or equal to operator 215
letters 57, 175
line controls 42
Line Edge Attributes dialog 53
line styles 31
lines 25
  adding 53
  changing 54
linking to web sites 58
links
  adding to reports 58
Linux platforms 2
ln function 201
Load file from URL dialog 36, 55
loading
  image files 55
  reports 40
  sample database 64
  templates 4, 36
localization 81
logarithmics 201
logical operators 209, 215
logos 24
lowercase characters 203
LRX (Live Report Extension) xiii

M
magnification 162
Magnification property 162
magnifying glass icon 10
mailing labels 178
  printing 179
Make Imagepath relative option 56
Make Vertical command 53
Manuals directory xvii
margins
  changing 160
  setting 13, 15
Margins property 160
Master Detail Query Wizard 113, 114
master sections
  creating 114
  defining dependencies 113
  retrieving data from 118
  setting control IDs for 118
master-detail reports 113, 116
  creating 117
mathematical functions 198
mathematical operators 215
matrix reports. See crosstab reports
max function 200
maximum values 200
measurements 153
memory 54
MetaData data sources 64, 123
metadata files 96
min function 200
minimum values 200
modification dates and times 159
Modified property 159
modules 2
month function 205
monthstring function 205
moving controls 28, 46, 153
moving through reports 7
MS SQL Server
  problems retrieving data from 139
  stored procedures and 137
multi-level contents page 11
multi-level reports 9
multiline text controls 57, 58
multipage reports 24, 33
Multiple Selections command 28
multi-table reports 102

N
naming
  zones 25
navigating through reports 7
nesting
  functions 189
  groups 25, 111, 112
  sections 117
New parameter entry dialog 124
New Report wizard 35
non-Windows platform 69
Northwind database
  connecting to 67
  loading 64
Northwind.mdb 3, 64
Not on First Page option 32
NOT operator 215
now function 193, 204
numbering report pages 32, 193
numerical comparisons 215
O
Object Attributes dialog 46
objects
accessing Java 82
adding stored 143
inserting in templates 142
inserting stored 139, 140
types of stored 134
ODBC Data Source Administrator 65
ODBC data sources
connecting to 64, 73
setting up 64
ODBC drivers 73, 139
ODBC manager 64
offline report creation 123
online documentation xvi
syntax conventions xx
typographical conventions xix
online help xvii–xviii
usage overview xvii
online reports ix, 2
See also reports
Open Database Connectivity. See ODBC
open server xii, xiii
opening
Crosstab wizard 181
Rich Text Editor 56
operators 189, 215
options 152
Options dialog
enabling 3-part SQL in 122
setting debug mode 157
setting relative paths in 55
setting template defaults 156
updating query schemas 155
OR keyword (SQL) 107
OR operator 215
Oracle drivers 138
Oracle stored procedures 135
ORDER BY clause (SQL) 98, 107
creating grouped reports with 108
report parameters in 125
updating 155
orientation
changing 160
setting 13
Orientation property 160
Other type 137
OutParam function 134, 139
output
displaying query results in 137
formatting 195
output devices 81, 175
output directory 154
output files 81
output parameters 134, 141
creating 139
–OutputDEVICE option 81
–OutputFILE option 81
overlapping controls 61
overlays 53
owner 159
P
page breaks
setting 33
page dimensions 160
page footer zone 23
page footers 26
page header zone
adding page headings to 33
overview 23
selecting fields for 38
page headers
database fields as 33
overview 24
page layouts
adding background colors to 29
adding borders around 160
adding controls 59
changing margins for 160
designing 18
inhibiting headers and footers 32
overlapping controls in 61
overview 29
selecting report styles for 35
setting dimensions for 160
Page Number command 194
page numbers 24
adding 32, 193
applying reset criteria to 51
changing formats for 194
page orientation
  changing 160
  setting 13
page properties 159
Page Setup command 13, 159
Page Setup dialog 159
paging through reports 7
palettes
  accessing color 31, 160
panes (designer)
  described 6
  resizing 10
Parameter Descriptor dialog 124, 127
parameter fields 124
  creating 124, 131
  creating list of values for 130
  customizing 128
  removing 127
parameter list 141
parameterized reports
  deleting fields from 127
  overview 123
parameters
  changing data types for 138
  creating input/output 141
  entering values for 124, 132
  returning incorrect data types 137
  specifying exact values for 130
  suppressing display of 130
passwords
  setting from the command line 81
paths
  absolute vs. relative 55
  setting for images 153
  templates 154
.PDF files xvii
PDF files
  embedding graphs in 175
  exporting to 2
  linking to 58
PDF viewer 154
pi function 195
pie graphs
  plotting data series for 174
  planning process for reports 18
  plotting multiple data series 174
portrait orientation 13
Postfix property 214
pow function 199
predefined functions 194, 195
predefined templates 4, 21
Prefix property 214
Preserve Color property 161, 162
Preserve Font property 161, 162
Preview page
  current changes not displaying 8
  displaying contents page on 157
  drilling down in 10
  overview 7
  selecting 19
previewing reports 7, 8
primary keys 102, 104, 181
printed documentation
  syntax conventions xx
  typographical conventions xix
printing
  mailing labels 179
  page footers 26
  page headers 24
  report footers 26, 156
  report headers 24
  reports 2, 13, 159
product summary x
product update information xv
programming interfaces x
Prompt for Parameters command 126, 130
properties
  changing label 47
  customizing 158
  displaying for ASCII files 78
  formatting in conditional expressions 213
  referencing 208
  setting control 45
  setting export 161
  setting for stored procedures 141, 143
  setting graph 172
  setting report page 159
properties file 163
proportional spacing problems 27
prototyping reports 123
proxy data sources 123
Q
quarter function 205
queries 8
  adding fields 93
  adding filters to 107
  adding SELECT FROM clause 96
  adding WHERE and ORDERBY clauses 98, 108
  changing 131
  containing joins 102
  creating 90, 122
  creating generic 96
  creating master-detail reports from 113
  displaying 90
  enabling 3-part SQL 122
  enclosing strings in 120
  format of 90
  importing 99
  in nested sections 117, 118
  overview 90, 95, 122
  removing joins from 104
  report parameters in 125, 130, 131
  sorting results 107
  specifying columns for results 137
  specifying values in specific formats 120
  updating 93, 102
  updating schemas for 155
Query page (Options) 155
query processor 122
query substitution 118
  formatted strings for 120
questionnaires 175

R
Radians function 196
random function 198
record count 25
  applying to groups 51
  applying to reports 50
  resetting 51
  restarting 50
Record Count command 51
record count controls
  adding 51
  as formatting option 42
  inserting in page headers 24
record numbers 193
records
  applying reset criteria to 50
  as forms 175
  grouping 25, 108
  retrieving from ASCII data sources 163
  tracking number of 50
rectangle controls
  adding 53, 54
  as formatting option 42
  changing 54
  filling 54
Rectangle tool 54
referencing
  columns 122, 189
  controls 190, 208
  properties 208
Refresh command 8
Refresh tool 9
registering tables 85
relational databases 73
  See also databases
relationships 104
Relationships page
  overview 104
  removing joins 105
relative paths 55, 153
release notes xv
remote databases 74
removing borders 31
renaming zones 25
Repeat on Every Page command 33
Repeat on Every Page property 33, 34
repeating group headers 33
report designer 2
  customizing 152
  formatting toolbar 44
  loading templates 4
  main window 19
  panes described 6
  resizing panes 10
  sections in 22
  zones in 23
report designs. See designing reports
report editor 6
report examples xv
Report Explorer 21
report files
  compressing 161
report footer zone 23
report footers 26, 156
report header zone
  inserting RichText controls 57
  overview 23
report headers 24
Report Parameters command 124, 127
report parameters. See parameters
Report Properties dialog 158
report server 2
report server API xi
report styles 35, 42
Report title property 159
report titles 24
ReportCast channels xi, xii
reporting solutions ix
reporting tool 2, 90
reports ix, 2
  adding descriptions 159
  analyzing designs 22
  building with sample demos 1, 3
  changing 6
  changing at runtime 62
  controlling access to 26
  controlling amount of detail in 25
  creating 6, 34
  creating offline 123
  customizing properties for 158
  design process for 8, 18
  generating 8
  generating with specific criteria 123
  launching 40
  laying out 29
  moving through 7
  numbering pages 32, 193
  planning processes for 18
  previewing changes in 7, 8
  printing 2, 13, 159
  prototyping 123
  running automatically 40
  saving in compressed formats 161
  setting email addresses for 163
  spacing problems with 27
  spanning multiple pages 24, 33
  viewing 10, 21, 26
  viewing zones in 23
  with defined dependencies 113
Requester API x
reserved characters and symbols 118
reset criteria 50, 51
  specifying 53, 190
Resize to Fit Content property 57, 58
resizing
  controls 57, 58, 153
  designer panes 10
  grids 176
  labels 47
  rectangles 54
  zones 27
restarting record counts 50
result sets
  defining columns for 143
  displaying values 95
  generating 142
  with single values 134, 142
ResultSet sheet 143
return addresses 163
return values 142
Rich Text command 56
Rich Text Editor 56
  toolbar for 57
RichText controls
  adding 57
  as formatting option 42
  changing 57
  formatting 56
  resizing 57, 58
  variable dimensions in 57
right alignment icon 44, 59
round function 199
rounding 199
rows
  defining for crosstabs 181
rulers 15
RunACJ.bat 4, 80
running headers 33
running on non-Windows32 platforms 4
running reports 40
Runtime Parameter field 132
Runtime Parameter Options dialog 133
runtime parameters 132
Sample application xiii
Sample database
  connecting to 67
  loading 64
  testing JDBC connectivity with 3
Sample report templates 4, 21
Sample reports xv, 3
Saving
  data sources 156
  graphs 175
  report parameters 126
  reports in compressed formats 161
  source code 85
  templates 20, 36, 40
Schemas
  displaying database 71
  prepending to queries 122
  updating query 155
Script Editor 211
Scripts
  adding conditional expressions to 208, 210
  creating 211, 212, 213
  embedding expressions 215
  entering control IDs in 211
  entering data types in 213
  entering operators in 215
  overview 208, 217
  restrictions for 210, 211
  validating 213
Scrolling 8
Search extension API x
Searches (help topics) xviii
Section IDs 95, 96, 190
  displaying 118
Section queries
  changing 96
  creating 131
  overview 95
Section Query dialog 131
Section SQL Query command 92, 96, 118
SectionID variable 190
Sections
  adding to designs 113
  containing multiple groups 113
  creating dependencies between 114
Creating query import 99
  nesting 117
  overview 22
  query substitution for nested 118
Security 26
Select a color dialog 48
Select Field dialog 94
Select object attributes dialog 47
SELECT statements
  creating 90, 96
  creating joins with 102
Select Tables button 71
Select Tables dialog 71
Selecting
  border styles 30
  database fields 38, 94
  database tables 38, 71, 83
  database tables for crosstabs 184
  fields for graphs 170
  multiple controls 28
  report styles 35
Send Behind Others command 61
Sender’s property 163
Separator property 163
Separators 76, 131, 163
  numerical expressions and 209
Series
  plotting multiple 174
Servers xi, xii, xiii, 2
  specifying SMTP mail 163
Setting margins 13, 15
Setting page orientation 13
Settings. See properties 45
Settings. See properties
Show Table of Contents property 161
  –ShowDESIGNER option 81
Sibling sections 113
Sin function 196
Single grouping template 91
SingleGrouping.jod 124, 168
SinH function 196
SMTP server property 163
Snap 153
Software Development Kit x
Software requirements 64
Sort criteria 108
sort order 107
sorting data 125
sorting query results 107
source code
   configuring connections with 82
cutting and pasting 85
location of 21
spaces in names 25, 61
spin control (graphs) 174
splitter bars 27
spreadsheets 24
   asymmetrical grids as 175
crosstabs as 180
SQL engines 155
SQL queries 8
   adding fields 93
   adding filters to 107
   adding SELECT FROM clause 96
   adding WHERE and ORDERBY clauses 98, 108
   changing 131
   containing joins 102
   creating 90, 122
   creating generic 96
   creating master-detail reports from 113
displaying 90
   enabling 3-part 122
   enclosing strings in 120
   format of 90
   importing 99
   in nested sections 117, 118
   overview 90, 95, 122
   removing joins from 104
   report parameters in 125, 130, 131
   sorting results 107
   specifying columns for results 137
   specifying values in specific formats 120
   updating 93, 102
   updating schemas for 155
SQL Server
   problems retrieving data from 139
   stored procedures and 137
sqrt function 199
square root 199
standard deviation 24, 201
starting e.Report Designer Java Edition 3, 40
STARTSWITH operator 216
statistical functions 24, 201
stdDev function 201
stored object descriptor 143
Stored Object Descriptor wizard 137, 140
stored objects
   adding to reports 143
   examples of 135
   inserting in reports 139, 140
   inserting in templates 142
   returning incorrect metadata 142
   types supported 134
   updating control IDs for 142
Stored Procedure tool 140
stored procedures
   changing data types 138
   creating 140, 143
   linking controls to output parameters for 139
   overview 134
   samples described 135
   setting properties 141, 143
   viewing available parameters 141
StoredProc function 134
strcat function 202
Stretch to Fit Cell option 177
Strike property 214
string functions 202
string substitution 118
preformatted 120
STRINGLENGTH operator 216
strings
   as identifiers 61
   as text qualifier 163
   as web links 58
   comparing 215, 216
   concatenating 202
   entering in conditional expressions 209
   formatted string substitution for 120
   formatting 122
   getting number of characters in 202
   returning substrings for 202
   testing for 216
   testing length of 216
   with varying lengths 57
strlen function 202
STRLEN_GT operator 216
STRLEN_LT1 operator 216
Today's Date command 6, 50
toLowerCase function 203
toolbars
  formatting 44
  Rich Text editor 57
Tools menu 64
totals 25
toUpperCase function 203
Transparent option 49
trends 2, 180
trigonometric functions 195
typestyles 48
typographical conventions xix
  syntax xx

U
units of measurement 153
UNIX platforms
  creating startup scripts for 80
  running e.Report Designer Java Edition on 4
  running sample database on 69
updating
  queries 93, 102
  query schemas 155
  report parameters 126
uppercase characters 203
URLs
  adding to reports 58
  loading files from 36
  loading images with 55
  specifying JDBC 73, 74, 81
user interface
  customizing 152
user names
  grouping alphabetically 25
  specifying for connections 81
user-defined formulas 208
users
  inputting report parameters 124
  setting passwords for 81
UTF8 formats 156

V
validating scripts 213
Value property 208
entering in conditions 214
entering in formulas 209
restrictions for 210
values 188
  creating custom list of 130
  displaying user-entered 132
  getting absolute 200
  passing to functions 189
  returning single 134, 142
  specifying default 124
  testing string 216
var function 201
variance 24, 201
verifying date formats 120
Viewer xiii, 2
viewing
  border colors 32
  database schemas 71
  formatting options 42, 44
  online documentation xvi
  query statements 90
  reports 10, 21, 26
  Table of Contents 9, 81, 157
  user-entered values 132
  zones 23
views
  prepending name to queries 122
Visible property 214

W
Wallpaper property 160
web browsers xiii, 10
  setting paths for 154
Web link settings dialog
  inserting web links with 58
web links controls
  adding 58
    as formatting option 42
    setting browser paths 154
web reporting tool 2
web site (Actuate) xv
web sites
  linking to 58
WeekOfYear function 205
WHERE clause (SQL) 98
  report parameters in 125
string substitution in 118
  updating 155
Windows platforms 2
  creating startup scripts for 80
running e.Report Designer Java Edition on 4
  running sample database on 67
  wizards 64
working directory 154
Workspace page (Options) 153
X
  x-axis
    creating 170, 173
    overview 173
Y
  y-axis
    creating 171, 174
    overview 173
year function 205
Z
  zip files 161
Zipped Output property 161
Zone Border command 30
Zone Fill Color command 29
zone-level formatting options 44
zones
  adding graphs to 168
  as running header 33
  changing dynamically 24
  creating scripts for 213
  defined 6, 22
  displaying 23
  hiding 26
  moving controls between 46
  moving controls in 28
  naming 25
  overview 23
  redisplaying 27
  resizing 27
  types described 24, 25, 26
  zoom values 8