

Crystal Reports™ 5.0 User's Guide

Seagate Software Information Management Group, Inc.
1095 West Pender St., 4th Floor
Vancouver, B.C., Canada V6E 2M6

Copyright ©1991-1996 (manual and software) by Seagate Software, Inc., All Rights Reserved.

Crystal Reports for Windows is a trademark of Seagate Software, Inc. or one of its subsidiaries. Borland and Delphi are trademarks of Borland International. Microsoft, MS, Visual C++, and Visual Basic are trademarks of Microsoft Corporation. Windows, when used in this manual, refers to the Microsoft Windows operating system. Portions of this program Copyright 1982-1995 Btrieve Technologies, Inc. All Rights Reserved. All other product names referenced are believed to be the registered trademarks of their respective companies.

Manual written by
Gary Carter, John Morse, Lisa Barry, Kirsten Sutton, and Mojgan Sami
(INSTRUCTIONS, ETC.), 1992-1996.

C O N T E N T S

Welcome to Crystal Reports

Welcome	2
Two kinds of hands-on tutorials	2
Command, button, key, and control conventions	3
Commands and Buttons	3
Dialog box Controls	3
Key combinations	3
Other Conventions	4
Mouse actions	5
Using Crystal Reports documentation	5
Online Help features	6
If you need more help... ..	8

Chapter 1 - Installation and Quick Start

Installation Requirements	12
16-bit version	12
32-bit version	12
Installing Crystal Reports	12
Windows 3.1 or NT 3.51	13
Windows 95 or NT 4.0	13
Installing on a network workstation	13
Windows 3.1 or NT	14
Windows 95 or NT 4.0	14
Upgrading from a previous version of Crystal Reports	15
Quick Start	15

Chapter 2 - What's New

Sensational new interface gives you power and control	24
Placing objects anywhere gives you new flexibility	25
Multiple Section Reports offer new reporting alternatives	25
Subreports expand report usefulness	25
Query tool adds ad-hoc querying capabilities	26
Conditional formatting adds new intelligence to reporting	26
Parameter Fields mean multi-purpose reports	27
Text Objects give you text with intelligence	27
One-stop form letters make custom mailings a snap	28
Preprinted-form reports easier than ever	28

Enhanced Cross-Tab capabilities help identify trends	29
Underlay Sections add stunning new effects	29
More powerful formulas extend your capabilities	30
In-place editing makes it easy to edit OLE objects	30
Dragging objects between reports cuts creation time	30
HTML exporting simplifies Web activities	31
Sample reports demonstrate professional techniques	31
New developer features make development easier, faster	32
New database support improves data access	32

Chapter 3 - Learning Crystal Reports

Learning Crystal Reports	34
User's Guide	34
Online Help	37
Books Online	38
Sample Reports	39
Sample Applications	39
Glossary	40
Sample Data - CRAZE.MDB	40
Suggested Learning Paths	43
New user (business)	43
Business user upgrading from earlier version	43
New power user	44
Power user upgrading from earlier version	44
New MIS user	45
MIS user upgrading from earlier version	45
Developer using Crystal Reports for the first time	46
Developer upgrading from earlier version	46

Chapter 4 - Getting to Know Crystal Reports

The Crystal Reports Window	48
The Menu Bar	48
File menu	48
Edit menu	49
View menu	49
Insert menu	49
Format menu	49
Database menu	50
Report menu	50
Window menu	50

Help menu	50
The Toolbar	51
The Format Bar	53
The Status Bar	54
Shortcut menus	56
Crystal Reports cursors	58
Crystal Reports Design Tab	60
Design Tab Areas	61
Area printing characteristics	62
Identifying and working with areas and sections	64
Other Design Tab capabilities	65
Crystal Reports Preview Tab	65
Similarities/Differences from Design Tab	66
The Data Age indicator	67
Other fundamentals	68
Grid	68
Freeform	68
Freeform with guidelines	69
Sections and objects	69
Underlying Objects	71
HANDS-ON (Report Design Environment)	72
How to move the toolbar and format bar	72
Moving the bars	72
Resizing the bars	72
How to add, delete and move guidelines	73
To add guidelines	74
To move a guideline	75
To delete a guideline	75
How to move and position objects using guidelines	75
To position objects with guidelines	75
To resize an object using guidelines	77
Snapping the Object(S) with Two Guidelines	77
How to turn the grid on/off	78
How to zoom your report in and out	79
How to undo/redo activities	80
How to drill down on summarized data	81
Drill down triangles	81
Drill down cursor	82

HANDS-ON (Sections and Areas)	83
How to add, delete, move, and merge sections	83
How to split and resize sections	86
HANDS-ON (Report Creation and Design)	89
How to select data and begin creating a report	89
How to add and link multiple tables	92
How to insert database fields	93
How to insert special fields	94
How to insert a Page n of N field	95
How to insert text objects	95
How to use a database field in a text object	96
How to insert a picture	97
How to select, move and resize objects	98
How to select objects	98
How to select multiple objects	98
How to make an object underlay a following section(s)	100
How to hide parts of the report	103
To hide sections	104
To set section properties	105
To hide objects	106
HANDS-ON (Finishing Your Report)	107
How to insert page headers and footers	107
How to add a title page to your report	108
How to add summary information to your report	109
HANDS-ON (Distributing Your Report)	110
How to export reports	110
Exporting to e-mail	112
Exporting to Lotus Notes	113
Exporting to an Exchange Folder	113
Exporting to HTML	114
Exporting to an ODBC data source	115
How to fax a report	117

Chapter 5 - Tutorial - Customer List

Overview	120
Before you begin	120
Getting Started	121
Creating Your Report	121
Selecting a database to use	123
Report sections	124
Inserting a field	125
Understanding fields	127
Handles, selecting fields	128
Adding additional fields	128
Reviewing your work	128
Displaying Field Names	129
Combining database fields in text object	130
Adding titles using text objects	132
Saving your report	134
Record Selection	135
A first look at selecting records	135
Entering your selection criteria	136
Deleting the Country field	138
Balancing field spacing	138
Grouping and Sorting	140
Grouping your report	140
Sorting Records	142
Understanding "Live" group headers	144
Completing the Report	145
Inserting company logo	145
Printing a hard copy	146

Chapter 6 - Reporting 101

Basic Report Design	148
Deciding on the content of your report	148
Purpose	150
Readers	151
Title	151
Do you know the data you want to use in your report?	151
Header information	152
Header information sources	152
Data types in the header	152

Footer information	152
Footer data sources	153
Data types in the footer	153
Report body data	153
Body data sources	153
Existing or calculated?	153
Data types in the body	154
Record or group selection	154
Groups	154
Group values	154
Group value positions	154
Grand totals, subtotals, averages, etc.	155
Flags	155
Flag options	155
Highlights	155
Sorting	155
Developing a prototype on paper	156
How to design a paper prototype with Crystal Reports	156
Concepts in Reporting	157
Report Expert, Custom Report or Custom?	157
Selecting Your Data	158
Linking	160
Placing data on your report	160
Formatting Data	161
Record Selection	162
Grouping, sorting and summarizing your data	162
Formatting your report	163
Report Distribution	163
Beyond Basic Reports	163

Chapter 7 - Multiple Section Reports

Using Multiple Sections in Reports	166
HANDS-ON	167
How to prevent variable length objects from overwriting each other	167
How to work with text objects	168
How to create a form letter using a text object	170
How to print conditional messages in form letters	177
How to format objects conditionally	178
How to alternate background colors for rows	179

How to eliminate blank lines	180
How to add blank lines conditionally	181

Chapter 8 - Formatting

Formatting Concepts	184
Properties	185
Absolute Formatting	185
Types Of Formatting Properties	186
Conditional Formatting	188

HANDS-ON (Absolute Formatting) 190

How to add color, shading, and borders	190
How to add/edit lines and boxes	191
How to add lines	191
How to edit lines	192
How to add boxes	193
How to edit boxes	193
How to change margins	194
Using the Page Margins dialog box	194
How to add/delete white space between rows	195
How to delete white space by resizing	195
How to add white space by resizing	196
How to set page orientation and paper size	197

HANDS-ON (Conditional Formatting) 198

How to create a footer that appears on all pages but the first	198
How to flag values that meet certain conditions	199

Chapter 9 - Sorting, Grouping, and Totaling

Sorting, grouping, and summarizing overview	202
Sorting	202
Sorting by record	202
Single field sorts	204
Multiple field sorts	204
Sorting and grouping together	204
Results using different sorting and grouping operations	205
Creating Custom Groups	207

HANDS-ON	209
How to do a single field sort	210
How to do a multiple field sort	213
How to group data	215
How to sort the records within groups	217
How to summarize grouped data	219
How to subtotal grouped data	221
How to sort based on summarized group values	223
How to extend prices and then subtotal the extensions	223
How to create multiple levels of subtotals	227
How to group based on first letter of company name	231
How to group data in intervals	235
How to calculate a percentage of the grand total	238
How to create Group Headers	241
STANDARD HEADERS	242
LIVE HEADERS	243

Chapter 10 - Formulas 101

What are Formulas?	250
How formulas are created -	
An Introduction to the Formula Editor	254
Fields box	255
Functions box	256
Operators box	257
Text box	257
Other formula conventions	258
Other Formulas	258
Group Field values	259
Formula Comments	259
Formula comment considerations	260
Formula Syntax	261
Text	261
Numbers	261
Fields	261
Operators	262
Functions	262
Brackets in formulas	262
How formulas are evaluated - Order of precedence	264

How to delete formulas from your report	264
To delete individual working copies of a formula	265
To delete the specification	265
How to copy formulas from Crystal Reports online Help	266
To copy formulas from online Help	266
How to copy formulas from one report to another	267
To copy the contents of a formula from one report to another	267

Chapter 11 - Advanced Formulas

How to create If-Then-Else formulas	270
Example	270
How to print Time or dateTime values conditionally	271
How to create multi-condition If-Then-Else formulas	272
How to format text with formulas	273
Example	273
How to use variables in formulas	273
Using variables to streamline formulas	273
Using variables to expand formula capabilities	275
Declaring the variable	276
Naming the variable	276
Variable Data Types	276
How to declare a variable	277
How to assign a value to a variable	278
Example assignment statement	279
How to combine a variable declaration and assignment expression	280
How to declare and assign values to multiple variables	280
How to conditionally assign values to variables	281
How to use an array in a formula	282
Using arrays with summary functions	283
A formula example	283
How to use a range in a formula	284
Extracting a range of values	284
Extracting a range of characters	285
How to use semicolons in formulas	286
How to debug a formula	287
Example	287
Solution	287
Debugging Steps	287
To create the formula field (@Formula1)	287
To create the formula field (@Formula2)	288

To create the formula field (@Formula3)	289
To create the formula field (Formula4)	289
To create the formula field (@Formula1+2)	290
To create the formula field (@FinalFormula)	290

Chapter 12 - Advanced Totaling

Introduction	292
How to maintain running totals in a list	292
How to maintain/subtotal running totals within groups	295
How to subtotal without grouping	298
How to subtotal true A to B, A to C reports	301

Chapter 13 - Parameter (prompting) Fields

Parameter Field Objects Overview	306
Multiple Parameter Fields	307
Parameter field considerations	307

HANDS-ON 308

How to create a parameter field	308
How to set record selection using parameter fields	309
How to use a parameter field in a formula	309
How to respond to parameter field prompts	310
How to use wildcards with parameter fields	311
How to do conditional formatting using parameter fields	312
How to set a report title using parameter fields	313
How to set sort order using parameter fields	314

Chapter 14 - Record and Group Selection

Record Selection	316
Specifying records/groups to be included	316
Setting up record selection	316
Interaction of the Expert and the Selection Formula Editor	319
Group Selection	320
Select Expert	320
Formula language	321
Top N	321
Record selection formula templates	322
Formula templates	322
Record selection templates	322
Selection Performance Tips	325

Doing record selection with a group selection formula	327
HANDS-ON	328
How to set up record selection using the Expert	328
How to set up group selection using the Expert	329
How to create a record or group selection formula	331
How to use record/group selection templates	332
How to select the top or bottom N groups	333
Chapter 15 - Subreports	
What are subreports?	336
Unlinked versus linked	336
Database links versus subreports in one-to-many situations	339
HANDS-ON	339
How to insert a subreport	340
How to link a subreport to the data in the primary report	342
How to combine unrelated reports using subreports	344
How to use subreports with "unlinkable" data	347
Linking to/from a formula field	347
LINKING UNINDEXED TABLES	349
How to show different views of the same data in one report	349
Chapter 16 - Cross-Tab Objects	
Cross-Tab Overview	352
Cross-Tab capabilities in Crystal Reports	357
Chapter 17 - Queries	
The Crystal Query Designer	360
Why Use A Query?	360
Using the Query Designer	361
How to use an SQL query that you designed elsewhere	362
How to create a new query	364
How to add tables and fields	365
How to link tables and specify a join type	367
How to add fields	368
How to identify unique values	369
How to summarize data with aggregate functions	370
How to sort records according to field values	372

How to specify records to be included	373
How to select groups to be included	374
How to create an SQL expression	377
How to create a query from another Crystal Query	379
How to select a Crystal query for a report	381

Chapter 18 - Dictionaries

Crystal Dictionaries Overview	384
Why use a Crystal Dictionary?	384
How to create a new Dictionary	385
How to open a data file	386
How to open an SQL or ODBC data source	388
How to link multiple tables	389
How to select tables and fields for users	390
How to add formulas	392
How to rename field headings, fields, and formulas	393
How to update the location of a database table	394
How to move fields/field headings within the list	395
How to add a new field heading	396
How to add Help text	396
How to add a graphic	397
How to create sample data for users to browse	399
How to edit an existing Dictionary	400
How to convert a Crystal Reports 3.x or 4.x Dictionary file	400
How to select a Crystal Dictionary for a report	401

Chapter 19 - Working With Databases

Relational Database basics	404
Aliases	406
Locating files	408
Indexed tables	409
Linking tables	411
Using SQL and SQL databases	435
For additional information	444

HANDS-ON 444

How to open Access queries through DAO	444
How to open Access queries through ODBC	446
How to open Access parameter queries	448

How to set up an ODBC data source	453
How to check settings for an ODBC data source	455
How to log on to an ODBC data source	457
How to add an ODBC database table to a report	457
How to log on to MS SQL Server via ODBC	459
How to log off an ODBC data source	461
How to change the ODBC data source accessed by a report	461
How to select a stored procedure from an SQL database and change stored procedure parameters	465
How to set up an A to B, A to C link	469
How to edit an SQL query	470
How to use an ACT! database	471
How to open the NT Event Log	473

Chapter 20 - Data Sources

Introduction	476
Why you should read this chapter	476
A note to developers	476
Four types of data	477
Direct access database files	478
Advantages	478
Disadvantages	478
Three layers	478
Common database formats	480
ODBC data sources	498
Advantages	499
Disadvantages	500
Five layers	500
Common ODBC database formats	504
Crystal Query data sets	514
Advantages	515
Disadvantages	516
Crystal Dictionaries	516

Chapter 21 - Graphing

Graphing Overview	522
Before you create your graph	522
The Graph/Chart Expert	522
Types Tab	523

Data Tab	523
Text Tab	523
Options Tab	524
Where to place your graph	524
Drill down with graphs	524
The data you can graph on	524
Editing your graphs - PGEEditor	525
Using the underlay feature with graphs	525

Chapter 22 - OLE

OLE Objects Overview	528
OLE and the Insert Picture command	529

HANDS-ON 530

How to insert a graphic/picture as an OLE object	530
--	-----

Chapter 23 - Commands

File Menu Commands	534
Edit Menu Commands	539
View Menu Commands	544
Insert Menu Commands	546
Format Menu Commands	553
Database Menu Commands	557
Report Menu Commands	560
Window Menu Commands	564
Help Menu Commands	565

Chapter 24 - Crystal Reports Functions

Introduction	570
--------------------	-----

Chapter 25 - Operators and Variables

Introduction	610
--------------------	-----

Chapter 26 - Application Development with Crystal Reports

Introduction To The Crystal Report Engine	632
Special Features Of The Report Engine	633
Before using the Report Engine in your application	633
Using The Crystal Report Engine	634
Creating reports	634

Creating the interface for printing the reports	634
Adding the Report Engine to your application	635
The Crystal Custom Control	636
Adding the Custom Control to your project	636
Using the Custom Control	637
The Crystal ActiveX Control (OCX)	639
Adding the ActiveX Control to your project	640
Using the ActiveX Control	641
The Crystal Visual Component Library for Delphi	642
Adding the VCL to your project	643
Using the Crystal VCL	644
The Report Engine Class Library	645
The Report Engine API	647
Declarations for the Report Engine API	647
Using the Report Engine API	648
Establishing A Print-Only Link	649
Establishing A Custom-Print Link	653
Report Engine API Variable Length Strings	665
Sample Code	668
Report Engine API Structures	668
Distributing Report Engine Applications	669
Additional Sources of Information	670

Chapter A - Report Processing Model

Multi-pass Reporting	672
What is a "Pass"?	672
Pre - Pass #1	672
Pass #1	672
TopN/BottomN/Group Sorting	673
Pass #2	673

Chapter B - Product Support, Upgrades, and Licensing

Product Support	678
Fax On Demand	678
Fax Support - FREE	679
Mail-in Support - FREE	679
Telephone Support	680
Extended Technical Support Policy	681
Product Registration	681
Crystal Reports Upgrade Plan	682

Product Return Policy	682
Product Replacement Policy	683
License Agreement And Limited Warranty	683
Restricted Rights Legend	688

Glossary

Index

Welcome to Crystal Reports

What you will find in this chapter...

Welcome 2

Two kinds of hands-on tutorials 2

Command, button, key, and control conventions 3

Using Crystal Reports documentation 5

Online Help features 6

If you need more help... 8

Welcome

Welcome to Crystal Reports, the most powerful Windows reporting tool available today.

- If you are new to the product, we are confident that you will soon understand why more than a million users worldwide turn to Crystal Reports for all their reporting needs.
- If you are upgrading, thanks for your vote of confidence. We have made every effort to make this release the best ever, offering major improvements over earlier versions.

The User's Guide has been totally redesigned to fit the way our users tell us that they work. The emphasis is on getting jobs done in the easiest way possible. To eliminate confusion, we have used a show-as-you-go style that uses pictures liberally to demonstrate procedures.

The User's Guide contains instructions for most of the typical reporting tasks and for some very sophisticated tasks as well. It also contains a great deal of conceptual information to help you better understand databases, reporting, and Crystal Reports itself.

There are many topics in this manual that deal with general areas of interest such as placing fields, formatting your report, and sorting records, as well as topics that deal with more specific areas of interest such as advanced formula creation and accessing different types of data.

In most chapters, the Hands-On topics flow from general to more specific areas of interest. Consult *What you will find in this chapter...* (first page of each chapter) to target the information you need in a hurry.

Two kinds of hands-on tutorials

Hands-On sections in this manual contain two types of tutorials.

- Detailed tutorials demonstrate core procedures, procedures that are used again and again. For example, even though you may use subreports in many different situations, the two core procedures you need to understand are:

- How to insert a subreport, and
- How to link a subreport to the data in the primary report.

Each of these is explained in step-by-step fashion.

- The other kind of hands-on tutorial is a process tutorial. Process tutorials deal with more specialized topics. These tutorials give you an overview of a process so you understand the concepts involved, and they give you cross references to core tutorials (in the current chapter and in other chapters) that provide the details. Following our subreport example, a typical process tutorial might discuss how to place three unrelated reports in a single report. One of the process steps is, "Create the first subreport." No details on how to do this are provided in this tutorial. But the tutorial does include a cross-reference to the core tutorial that will explain in a detailed fashion how to create a subreport.

Command, button, key, and control conventions

This User's Guide uses the following conventions:

Commands and Buttons

For easy recognition, command names and button names from the toolbar and format bar are displayed in small caps. For example: NEW command, PRINT button, etc.

Dialog box Controls

Buttons, lists, and other dialog box elements are displayed in italics. For example, *Visible* check box, *OK* button, etc.

Key combinations

Keyboard shortcuts appear in the following forms:

- Delete means the Del key (either the Delete key or the Del key on your numeric keypad).
- Escape means the Escape or Esc key on your keyboard.
- Enter means the Enter, Return, CR or ↵ key, depending on which of these keys appears on your keyboard.
- Ctrl-Key, Shift-Key, and Alt-Key are examples of the notation for two key combinations. Press the first key in the

combination (Control, Shift, or Alt), and, at the same time, press the second key in the combination (designated above as Key). For example: Ctrl-C means to hold the Control key down and then press the letter C on your keyboard (Windows Copy command).

Other Conventions

- Text enclosed in double brackets (for example, «information») is intended to expand or explain the information that it follows.
- `Computer type` indicates data that you are to enter using the computer keyboard. It is also used to show example formulas.
- Field names appear in the following format:
 - `{file.FIELD}`
 - *file* represents the alias of the table the field comes from.
 - *FIELD* represents the name of the field in that table.
- Text that you should pay special attention to within normal body text is underlined.
- **NOTES** are used to provide extra or special information regarding the preceding topic.
- `\WINDOWS\SYSTEM` refers to the System subdirectory of the Windows directory on your computer or network server. By default, this directory is `C:\WINDOWS\SYSTEM` in Windows 3.1 and Windows 95 and `C:\WINNT35\SYSTEM32` in Windows NT 3.51 or later. This directory may be different on your system. If you are unsure, contact your network administrator, or refer to your Windows documentation.
- *Related topics* lists point to other topics in the User's Guide and online Help that deal with related procedures, additional uses for the same command or dialog box, or alternative methods for accomplishing the same thing.
- Not all of the pictures used in the Hands-On sections reflect exactly what you will see on screen.
 - Some of the example reports have been designed to illustrate concepts only, not the actual look of your finished report.

- Some menus have been shortened by removing some of the commands that are unrelated to the current discussion.
- The data in some lists has been abbreviated to focus attention on the specific items of interest.
- In some cases, an individual screenshot illustrates both a before and after state of a dialog box when such an illustration can be done without confusion.
- In the Hands-On tutorial sections, be sure to pay attention to the callout text accompanying all screenshots and graphics. Often the callouts are steps necessary to complete the tutorial and to pass by one inadvertently may cause undesirable or incorrect results.

Mouse actions

This manual uses the following words to describe mouse actions: (Use the left mouse button unless instructed otherwise.)

- **Click**
Click the left mouse button once.
- **Right-click**
Click the right mouse button once.
- **Double-click**
Click the left mouse button twice rapidly without moving the mouse.
- **Shift-click**
Hold down the Shift key on your keyboard and click the mouse button.

Using Crystal Reports documentation

In addition to this User's Guide, Crystal Reports includes a comprehensive set of online and printed learning tools to help you when you are getting started with the program and when you need answers fast in your day-to-day reporting.

Whether you are a beginner or an expert, Crystal Reports documentation provides a clear and easy path to productivity. For a complete description of learning tools and suggested learning paths, see Chapter 3, *Learning Crystal Reports*, Page 33.

PROFESSIONAL EDITION FEATURES

This User's Guide is used with both the Standard and Professional Editions of Crystal Reports. The following list identifies those features and capabilities that are discussed in the Guide but that are available only in the Professional Edition.

- Data Dictionary Builder
- Crystal Query Designer
- Report Engine Class Library
- The ability to read SQL databases
- Access to Microsoft Exchange data files
- The ability to export to ODBC data sources

Online Help features

The online Help is full of useful information. Here is just a sampling of things you find there:

- Explanations of error messages and formula compiler errors. Search for *Error Messages and Formula Compiler Warnings*.
- Runtime information so you know which files to include when you include the Crystal Reports Engine with your application. Search for *Runtime File Requirements*.
- Tips and Tricks: dozens of helpful hints for working with Crystal Reports. Search for *Tips and Tricks*.
- Formulas for study: a series of complex formulas that showcase the use of Crystal Reports functions, operators, and formatting language in solving a number of real-world reporting problems. Search for *Formulas in Action Index*.
- Specialized formulas - a collection of formulas that address specific reporting needs. Search for *Specialized Formulas*.
- Sample formulas: a number of topics contain sample formulas that can be cut and pasted directly into the Formula Editor to save you time. For pointers to those topics, search for *Copying formulas from Crystal Reports online Help*.

- Formula functions and operators: a complete and detailed list of all functions and operators you can use when creating formulas. Search for *Functions* or *Operators* or for the individual function or operator by name.
- Full documentation of the Report Engine API. Search for *Report Engine API Functions*.
- Sample code for making calls to the Report Engine from your C, Visual dBASE, Visual Basic, or Delphi application. Using the Edit capabilities built into the Windows help facility, you can cut this code and paste it as ASCII text into any editor that supports Windows cut and paste procedures. Search for *Copying sample code* in online Help.
- Full documentation of the Report Engine Class Library, a C++ class library addition to the Microsoft Foundation Class Library. Use the class definition in the Report Engine Class Library to access the Crystal Reports Engine from your C++ application. Search for *Report Engine Class Library* in Crystal Reports online Help.
- Full documentation for the Crystal Custom Control, the Crystal ActiveX control, and the Crystal VCL you can use when working with development environments that support them. Search for *The Crystal Custom Control*, *The Crystal ActiveX Control*, or *The Crystal VCL*.
- Dissections of sample applications included with the program. Search for *Report Engine Sample Applications*.

Online Help is a warehouse of information that can make your reporting more productive and enjoyable. A comprehensive indexing system and hundreds of search terms provide you a variety of avenues for finding the help you need, right from your computer. Once you see what's in the help system, we are confident you will return to it often.

If you need more help...

WEBSITE

<http://www.img.seagatesoftware.com>

INTERNET

- **Report Creation and Design**
tsrcad@img.seagatesoftware.com
- **Delphi/VCL**
tsvcl@img.seagatesoftware.com
- **dBase and Paradox**
tsxsbse@img.seagatesoftware.com
- **Report Engine API**
tscrpe@img.seagatesoftware.com
- **Crystal Info Product Support**
tscinfo@img.seagatesoftware.com
- **Visual Basic, VBX, OCX Support**
tsvbocx@img.seagatesoftware.com
- **PC database connectivity Support**
tspcdata@img.seagatesoftware.com
- **SQL/ODBC connectivity Support**
tssql@img.seagatesoftware.com

COMPUSERVE

GO REPORTS

FAX ON DEMAND

Fax On Demand (604) 681-3450

FAX SUPPORT

Fax Support (604) 681-7163

MAIL-IN SUPPORT

Seagate Software Information Management Group, Inc.
1095 West Pender Street, 4th Floor
Vancouver, BC, Canada V6E 2M6

TELEPHONE SUPPORT

Telephone Number (604) 669-8379

For more information on these services, please see Appendix B, Product Support, Upgrades, and Licensing.

1

Installation and Quick Start

What you will find in this chapter

Installation Requirements 12

 16-bit version 12

 32-bit version 12

Installing Crystal Reports 12

 Windows 3.1 or NT 3.51 13

 Windows 95 or NT 4.0 13

Installing on a network workstation 13

 Windows 3.1 or NT 14

 Windows 95 or NT 4.0 14

Upgrading from a previous version of Crystal Reports 15

Quick Start 15

Installation Requirements

16-bit version

The 16-bit version of Crystal Reports has the following installation requirements:

- Microsoft Windows 3.1 or higher
- 8 MB RAM (16 MB for Windows NT 3.51 or higher)
- Minimum hard drive space required: 14 MB
- Maximum hard drive space required:
 - 82 MB for Crystal Reports Standard Edition
 - 87 MB for Crystal Reports Professional Edition

32-bit version

The 32-bit version of Crystal Reports has the following installation requirements:

- Microsoft Windows 95 or Microsoft Windows NT 3.51 or higher
- 8 MB RAM (16 MB for Windows NT)
- Minimum hard drive space required: 10 MB
- Maximum hard drive space required:
 - 69 MB for Crystal Reports Standard Edition
 - 75 MB for Crystal Reports Professional Edition

Your system must meet these requirements for the version of Crystal Reports that you have. If your system does not meet these requirements, Crystal Reports may not run correctly.

Installing Crystal Reports

Insert the Crystal Reports CD or diskette in the appropriate disk drive.

Windows 3.1 or NT 3.51

1. Make certain that the Window Program Manager is active. Choose the RUN command from the File menu.
2. When the Run dialog box appears, type:
`x:\setup`
where x represents the appropriate drive letter. For example if your CD is in drive d:, type:
`d:\setup`
3. Click *OK* or press Enter to activate the Crystal Reports installation program. The Installation dialog box appears.
4. Follow the directions on the screen to set up Crystal Reports.

Windows 95 or NT 4.0

1. Choose RUN from the Start menu.
 2. When the Run dialog box appears, type:
`x:\setup`
where x represents the appropriate drive letter. For example if your CD is in drive d:, type:
`d:\setup`
 3. Click *OK* or press Enter to activate the Crystal Reports installation program. The Installation dialog box appears.
 4. Follow the directions on the screen to set up Crystal Reports.
- During the installation procedure, you will be given the choice to have Setup do an automatic installation or a custom installation. If you select *Automatic*, Setup installs all of the Crystal Reports files to your hard drive. If you select *Custom*, you will be given the opportunity to select which components of the Crystal Reports application files are installed on your system.

Installing on a network workstation

If you are operating a network workstation that runs Windows locally and Crystal Reports is already installed on your network server, you do not need to install Crystal Reports on your workstation. Instead, you must run the workstation setup application that was installed with Crystal Reports.

Workstation setup (SETUP.EXE) is installed in the
\CRW\WKCSETUP (16-bit) or \CRW\WKCSTP32 (32-bit)
directory on the network drive on which Crystal Reports was
installed. To set up your workstation:

Windows 3.1 or NT

1. From Program Manager, choose RUN from the File menu. The Run dialog box appears.
2. In the Run dialog box, select the network drive and directory in which Workstation Setup resides and run the SETUP.EXE application.
3. Follow the directions on the screen to set up your workstation.

Windows 95 or NT 4.0

1. Choose RUN from the Start menu. The Run dialog box appears.
2. In the Run dialog box, select the network drive and directory in which Workstation Setup resides and run the SETUP.EXE application.
3. Follow the directions on the screen to set up your workstation.

Workstation setup configures your workstation and copies several files to your local drive. When Workstation Setup is finished, you can access Crystal Reports as you would any other application on your network.

NOTE: When you perform an automatic installation (vs. a custom installation), the Crystal Reports installation program will ask you if you want to install the Workstation setup files. If you are installing on a stand-alone machine or will not be using the Workstation Setup on your network, these files are not needed and will take up extra disk space if installed.

If you convert to a workstation at a later time, you can once again perform a custom installation and install only the Workstation install files.

Upgrading from a previous version of Crystal Reports

If you are upgrading to Crystal Reports from a previous version (4.0, 4.5, Crystal Reports for Visual Basic, etc.), the installation routine ensures that there will be no conflict between different versions of the program running on the same machine. When it finds a previous version of Crystal Reports on your system, the setup application:

- installs Crystal Reports to the directory you specify,
- installs the new CRPE.DLL into the \WINDOWS\SYSTEM directory,
- renames DLLs in the WINDOWS\CRYSTAL directory, installed by your previous version of Crystal Reports, with *.OLD extensions. If, for some reason, you need to use the older versions of the files later on, you only need to rename them back to a *.DLL extension.
- installs the PD*.DLLs and UX*.DLLs (16-bit) or the P2*.DLLs and U2*.DLLs (32-bit), and several other DLL files required by Crystal Reports into the WINDOWS\CRYSTAL and WINDOWS\SYSTEMS directory and changes their extensions from *.DLL to *.OLD.
- installs any common third party DLLs such as CTL3DV2.DLL or WBTRCALL.DLL to the WINDOWS\SYSTEM directory.

This upgrade procedure makes it unnecessary to change your AUTOEXEC.BAT file.

Quick Start

If you are an experienced Windows user who wants to get right into the program, follow these steps to set up a report for the first time.

NOTE: If you are not an experienced user, please refer to Chapter 3, Learning Crystal Reports, Page 33.

1. In Windows 3.1 and NT 3.51, start Crystal Reports by double-clicking the Crystal Reports icon in the Program Manager. In Windows 95 and NT 4.x, click Start and then select the program from the Crystal Reports program folder.



2. Click the NEW button on the toolbar. The Report Gallery appears.
3. Select one of the eight Experts to build a report with the help of an Expert, click the *Another Report* button to use a template for building your report, or click the *Custom* button to build a custom report. If you click *Custom*, the Report Gallery expands, and you can then select a *Report Type* and *Data Type* for your custom report.
4. If you choose *Custom* and select:
 - *Data File*, the Choose Database File dialog box appears. Select the first database you want to activate for your report.
 - *SQL/ODBC*, the Log On Server dialog box appears. Select the data source you want and then select the first table you want to use from the Choose SQL Table dialog box when it appears.
 - *Dictionary*, the File Open dialog box appears. Select the dictionary you want to use for your report.

NOTE: If you want to use additional database tables for your report and match them up on a record-by-record basis, click the Link Expert button and then select the table(s) and set up the links in the Visual Linking Expert when it appears. Search for Visual Linking Expert in Crystal Reports online Help.

The Crystal Reports Design Tab appears with Report Header, Page Header, Details, Page Footer and Report Footer areas. You create your report by inserting and formatting items in each of these areas.



5. Each of the default report areas contains a single section. If you want to add additional sections, click the SECTION EXPERT button and add the sections you want in the Section Expert.
 - Once you have added sections to an area, you can move, merge, and delete them in the Section Expert. See *How to add, delete, move and merge sections*, Page 83.
6. If you want to toggle the grid on and off, choose the OPTIONS command on the File menu and make your changes in the File Options dialog box when it appears. See *How to turn the grid on/off*, Page 78.

7. If you are working with the grid off and you want to use snap-to guidelines for positioning objects, click the top or left ruler wherever you want guidelines to appear.
 - Move a field to a guideline until it snaps to the guideline.
 - Move the guideline arrow to move the guideline (and any objects that are snapped to it).
 - Drag the guideline arrow away from the ruler to remove the guideline. See *How to add, delete and move guidelines*, Page 73, and *How to move and position objects using guidelines*, Page 75.



8. If the Insert Fields dialog box is not visible, click the INSERT FIELDS button. The Insert Fields dialog box appears with the Database Tab active. This dialog box displays a list of all of the fields in the active database table(s). To speed the entry of multiple fields, the box remains on screen until you click the Close button. You can move the dialog box to a new location or resize it if you wish. See *How to database fields*, Page 93.
9. Select the field(s) you want to appear on the report. You can select and place them one at a time, or you can use the Shift-click combination to select a number of contiguous fields or the Ctrl-click combination to select fields from the list at random. Drag and drop is also active. Place the fields in the Details section where you want them to appear.
10. If you place multiple fields, they will appear next to each other in the order they appear in the Insert Fields dialog box. The program marks the position of each field with a rectangular frame. The characters in the frame indicate whether the field is text (xxx...), number (555...), currency (\$555...), date (12/31/99), time (00:00:00), dateTime (12/31/99 00:00:00), or Boolean (T/F).

NOTE: You can see the actual field names and field types by toggling the *Show Field Names* check box on in the *File Options dialog box (Layout Tab)*. Search for *Setting up Crystal Reports in Crystal Reports online Help*.

NOTE: The program automatically places field titles in the *Page Header* section unless you have toggled the *Insert Detail Field Titles* check box off in the *File Options dialog box (Layout Tab)*. Search for *Setting up Crystal Reports in Crystal Reports online Help*.

NOTE: If you add additional Details sections to your report, please note that field titles will only be placed in the Page Header section for fields in the Details A (the original) section of your report.

11. Once you have objects in place, you may want to adjust the report sections somewhat. You do this using the shortcut menu that appears when you right-click the shaded area to the left of the section ruler.
 - If you want to expand the section to accommodate an additional line, choose the INSERT LINE command.
 - If you want the program to automatically align the objects in the section horizontally, choose the ARRANGE LINES command.
 - If you want to reduce the size of the section to eliminate unnecessary white space above and below objects, choose the FIT SECTION command. See *How to add/delete white space between rows*, Page 195.



12. To create a report title, click the INSERT TEXT OBJECT button on the toolbar. A rectangular placement frame appears when you move the cursor over your report. Click once in the Report Header section to place the text object and type in your title. See *How to insert an object - Text object*, Page 93, and *How to add a title page to your report*, Page 108.



13. To see how your results will print, click the PRINT PREVIEW button on the toolbar.
 - If you want to speed processing time while building your report, you can preview your report using only a small subset of the available data. To do this, choose the PREVIEW SAMPLE command from the File | Print menu. See *Crystal Reports Preview Tab*, Page 65.

In either case, Crystal Reports takes you to the Preview Tab. You can fine tune your report in the Preview Tab if you wish while seeing the results as actual report data. You can also close the Preview Tab and continue working on your report in the Design Tab.

14. If you want to:

- change the placement or width of a field,
- format a field, or
- insert a subtotal or grand total,

click the field to select it. Handles appear on the top, bottom, and both sides of each selected field.

- To change the placement of the field(s), drag the field placement frame to its new position using the mouse.
- To change the width of the field, drag the right or left handle using your mouse.
- To format or summarize the field, right-click the field. A shortcut menu appears listing various commands for formatting the field.
- To change the font, format (alignment within field, number, currency, and date display, etc.), border, color, indentation, etc., choose the **FORMAT FIELD** command. The **Format Editor** appears with specific to the field you have selected.



- If you want the formatting to apply only under certain conditions, click the *Formula* button next to the formatting property, and create a formula that defines those conditions. Use a Boolean (True/False) formula to set conditions for check box (on/off) properties. Use an If-Then-Else formula to set conditions for attribute properties (where you specify values). See *Conditional Formatting*, Page 188.

NOTE: Many of the font and formatting options are available on the format bar just below the toolbar.

- To insert a grand total, right-click the field you want to summarize and choose the **INSERT GRAND TOTAL** command from the shortcut menu and refine your selection using the **Insert Grand Total** dialog box when it appears. Crystal Reports places the grand total in the Report Footer section.

- To insert a subtotal, right-click the field and choose the INSERT SUBTOTAL command from the shortcut menu and refine your selection using the Insert Subtotal dialog box when it appears. See *How to subtotal grouped data*, Page 221.

NOTE: The program automatically sorts the data (based on the field that triggers the subtotals) before it subtotals. You do not have to manually sort data that has been subtotaled.



15. If you want to create a formula to make data calculations or comparisons, click the INSERT FIELDS button on the toolbar. When the Insert Fields dialog box appears, click the Formula Tab to make it active.
 - Click the *New* button. The Formula Name dialog box appears. Enter a name for your formula and click *OK*. The Formula Editor appears.
 - Enter the formula in the Formula Editor. Enter fields, operators, and functions by selecting them from their respective scroll lists or type them in. You can check your formula syntax via the *Check* button. When finished editing, click the *Accept* button to return to the Insert Fields dialog box. Click the *Insert* button to place the formula just like you do a database field. See Chapter 10, *Formulas 101*, Page 249, Chapter 11, *Advanced Formulas*, Page 269, Chapter 24, *Functions*, Page 569, and Chapter 25, *Operators and Variables*, Page 609.



16. To insert a subreport (a report within a report), click the INSERT SUBREPORT button on the toolbar and choose an existing report to import as a subreport or use the Create Report Wizard to create a new subreport. See *How to insert a subreport*, Page 340.
 - If you want the records in your subreport to match up with the records in your primary report, click the Subreport Link button and specify the link in the Subreport Links dialog box when it appears. See *How to link a subreport to the data in the primary report*, Page 342.



17. To insert a Cross-Tab object in your report, click the INSERT CROSS-TAB button and set up the Cross-Tab in the Cross-Tab dialog box when it appears. See Chapter 16, *Cross-Tabs Objects*, Page 351.



18. To create a parameter field (a field that prompts you for a value whenever you retrieve data for your report), click the INSERT FIELDS button on the toolbar, then click the Parameter Tab. Click the *New* button to set up a parameter field. Once created, you can insert the parameter field in your report like a database field or select it from the *Fields* list in the Formula Editor.

- You can use parameter fields in your report (as title or label prompts), in selection formulas (as selection criteria prompts), and in formulas (for a variety of purposes including specifying sort fields). See Chapter 13, *Parameter (prompting) Fields*, Page 305.



19. To add a graph, click the INSERT CHART button on the toolbar. See Chapter 21, *Graphing*, Page 521.

20. To insert a spreadsheet, picture, or other OLE object that you can edit from within Crystal Reports using the tools from the object's native application, choose the OBJECT command from the Insert menu. See Chapter 22, *OLE*, Page 527, and *How to insert a graphic object*, Page 97.



21. To change the record sort order, choose the SORT ORDER button on the toolbar. The Record Sort Order dialog box appears. Select the field(s) you want Crystal Reports to use for sorting the report data and the sort direction. See *How to do a single field sort*, Page 210.



22. If you want to limit your report to specific records (for example, the records of California customers that have YTD sales greater than \$10,000), click the first field on which you want your selection to be based and click the SELECT RECORDS button from the toolbar. When the Select Records Expert appears, set up your record selection criteria. See Chapter 14, *Record and Group Selection*, Page 315.



23. To print your report, click the PRINT button.

That's it! It is that easy to build a report with Crystal Reports. To practice using Crystal Reports by completing step-by-step lessons, proceed to Chapter 5, *Tutorial - Customer List*, Page 119.

2

What's New

What you will find in this chapter...

- Sensational new interface gives you power and control 24
- Placing objects anywhere gives you new flexibility 25
- Multiple Section Reports offer new reporting alternatives 25
- Subreports expand report usefulness 25
- Query tool adds ad-hoc querying capabilities 26
- Conditional formatting adds new intelligence to reporting 26
- Parameter Fields mean multi-purpose reports 27
- Text Objects give you text with intelligence 27
- One-stop form letters make custom mailings a snap 28
- Preprinted-form reports easier than ever 28
- Enhanced Cross-Tab capabilities help identify trends 29
- Underlay Sections add stunning new effects 29
- More powerful formulas extend your capabilities 30
- In-place editing makes it easy to edit OLE objects 30
- Dragging objects between reports cuts creation time 30
- HTML exporting simplifies Web activities 31
- Sample reports demonstrate professional techniques 31
- New developer features make development easier, faster 32
- New database support improves data access 32

This is Crystal Reports as you've never seen it before. The program has been redesigned to improve performance and offer an incredible list of new capabilities.

Sensational new interface gives you power and control

The report design environment maintains all of the capabilities of the old environment while offering greater flexibility than ever before.

- The Design and Preview Tabs are broken into areas and each area can contain multiple sections, each with different data and/or properties.
- The new Format Section dialog box enables you to insert, delete, move, merge, and format sections from one place. See *Chapter 4, Getting to Know Crystal Reports*, Page 47.
- Vertical and horizontal guidelines make it easy to place and align objects with precision. See *How to add, delete and move guidelines*, Page 73, *How to move and position objects using guidelines*, Page 75.
- Multiple levels of undo and redo make it easy and safe to experiment, and they provide a fail-safe mechanism against inadvertent errors. See *How to undo/redo activities*, Page 80.
- New zoom capabilities enable you to zoom in and out on your data, viewing it at magnifications from 25% to 400% of its actual size. See *How to zoom your report in and out*, Page 79.
- The toolbar and format bar can be moved and resized in Crystal Reports 32-bit version, enabling you to customize your environment for maximum efficiency. See *How to move the toolbar and format bar*, Page 72.
- The Preview Tab now lets you expand and collapse report sections so you can review data with different levels of detail. See *How to drill down on summarized data*, Page 81.

These are just a few of the new features in the design environment. Many of the other major program changes are described below.

Placing objects anywhere gives you new flexibility

In previous versions of Crystal Reports, object placement was always controlled by an underlying grid. In this version Crystal Reports, the grid is still available if you need it, but you can also construct reports in a freeform environment similar to that of a drawing program. You can place objects wherever you want them and resize them to your precise specifications.

You can also create reports that print perfectly on preprinted forms, stagger fields, and create unusual alignments. Basically, if you can conceive it, you can probably create it with Crystal Reports. See Chapter 4, *Getting to Know Crystal Reports*, Page 47.

Multiple Section Reports offer new reporting alternatives

Now you can include multiple sections in any area of your report and you can format each section independently of the others. Using this capability you can create custom form letters that deliver different messages to different readers, format numbers and dates differently depending on the audience, create unusual printing effects, force objects to print in the order you specify, and more. See Chapter 7, *Multiple Section Reports*, Page 165.

Subreports expand report usefulness

Crystal Reports enables you to create subreports and place them in your report. You can create freestanding reports based on tables that are not related to those in the primary report, or you can bind the subreports to primary report data.

With subreports, you can create one report that provides all the information you need instead of having to create multiple reports. You will find it easier to create reports and analyze report data with the new subreport feature.

Subreports can be inserted into most sections of the main report and the main report can contain numerous subreports.

Existing report (*.RPT) files can be used as subreports. Once a report file is inserted as a subreport, it is an object in the main

report and is not linked to the original report file (i.e., no automatic updating). Subreports can have any report format; they can be Cross-Tabs or contain graphs. Thus, you can present columnar data from one report right next to a graph from another report.

Subreports let you present different types and views of data all on one report for convenient viewing of information and efficient analysis. See Chapter 15, *Subreports*, Page 335.

Query tool adds ad-hoc querying capabilities

Crystal Reports now includes a powerful Query tool that allows processing of all aggregate functions on the database server. A complete SQL Editor is included to fine tune your queries. And, as you would expect, you can create reports from query result sets. Ad-hoc queries, reports, or both? Crystal Reports has it all. See Chapter 17, *Queries*, Page 359.

Conditional formatting adds new intelligence to reporting

Crystal Reports has always been known for its flexibility in formatting report data. This version takes a giant step beyond earlier versions by allowing you to format data conditionally based on whatever criteria you want to use.

- Do you want sales figures to print in red if more than 10% under quota, in green if more than 10% over quota, and in black in all other cases?
- When your data contains domestic and international data, do you want to print dates and currency values in the expected format for every country?
- Do you want a section to print only if it contains a record from a Canadian company?
- Do you need to customize your report colors to meet corporate standards or design requirements?

You can do these and countless other things using conditional formatting.

Virtually all field, object, and section formatting (color, font, border, alignment, visibility, and more) can be controlled by creating simple, straightforward formulas. See *Conditional Formatting*, Page 188, Chapter 10, *Formulas 101*, Page 249, and Chapter 11, *Advanced Formulas*, Page 269.

Parameter Fields mean multi-purpose reports

Now you can create parameter fields. Parameter fields are fields that prompt for a value when the report is run for the first time or the data is refreshed. With parameter fields, you can change report titles or labels, modify record selection criteria, specify sort fields, set threshold levels for flagging purposes, all by responding to simple prompts when you run the report. Using parameter fields you can create a single report that satisfies a number of needs. It makes your work easier and more efficient. See Chapter 13, *Parameter (prompting) Fields*, Page 305.

Text Objects give you text with intelligence

Text objects give you flexibility and control when you insert text in your report.

- Text objects replace text fields. A text object can contain a character, a word, a paragraph, or even an entire document, making them perfect for creating customized form letters (see the section on Form Letters below).
- Each text object contains its own mini word processor, making it easy to insert and edit text.
- Text objects and elements within text objects can be independently formatted with fonts, colors, line breaks, tabs, and more.
- Database fields can be inserted into a text object in-line with automatic trimming to fit properly within text. Text objects can also contain formula fields.
- Text objects can be inserted and edited in-place on the report, making report design easier. Now you can readily see how a text object looks on the page in relation to other objects.

- Presentation quality is improved with total control and flexibility over text object formatting. See *How to insert text objects*, Page 95.

One-stop form letters make custom mailings a snap

Creating form letters has never been easier. By creating multiple form letters in text objects and setting them each to print only if specific criteria are met, you can send customized letters tailored to the needs of targeted clients or customers.

Assume, for example, that you have customers that are over their credit limits and others that have available credit. You can create one letter to over-limit customers asking them to reduce their balance and another letter to under-limit customers encouraging them to buy more. Crystal Reports can:

- pull the data for both letters out of a single table or set of tables,
- identify the correct letter for each customer,
- customize the letter with such things as contact, company name, balance, and over/under limit amounts, and
- print the letters for you.

You can boost your marketing efforts, improve your receivables collections, and satisfy other business objectives with Crystal form letters. See *How to create a form letter using a text object*, Page 170.

Preprinted-form reports easier than ever

The ability to design reports that print on forms is greatly enhanced now that you can place objects anywhere in the new freeform, drawing program-type environment. New rulers and guidelines facilitate the precise placement of objects on the form, and the built-in tab and ruler capabilities in text objects gives you more flexibility and control. There's even a new Forms Expert that speeds the design process. For fast, efficient reporting on preprinted forms, you can not beat Crystal Reports. See *How to make an object underlay a following section(s)*, Page 100.

Enhanced Cross-Tab capabilities help identify trends

Cross-Tabs are now objects that you can insert into your report.

- You can insert as many Cross-Tabs as you need, and you can even insert Cross-Tabs in subreports.
- You can use formulas that are defined elsewhere on the report in Cross-Tabs.
- You can include multiple summary fields or calculations in your Cross-Tabs. For example, for a single item you can summarize both quantity and price information.
- You can print Cross-Tabs that extend beyond the width of a page.
- You can independently format rows and columns with background colors, borders, and fonts for better looking and easier-to-understand Cross-Tabs.
- You can create Cross-Tabs that are longer than a single page and the program will automatically repeat the column headings at the top of each page.
- You can design and edit Cross-Tabs easily with the ability to call up the Cross-Tab Expert from both the Design and Preview Tabs. See Chapter 16, *Cross-Tab Objects*, Page 351.

Underlay Sections add stunning new effects

Crystal Reports enables you to underlay multiple sections when you print bitmap objects. This allows you to:

- print an object so it appears one time in the Details section, beside a number of details,
- print a company watermark centered on the page, flowing through multiple sections, and
- print Cross-Tabs or graphs so they contain the data that you want while appearing in a section that would normally generate different data.

You also have the ability to optionally print or not print the background image with the report for maximum flexibility. This enables you to place bitmaps and use them as design tools yet not print them when you print the rest of your report. (See *How to make an object underlay a following section*, Page 100.

More powerful formulas extend your capabilities

Crystal Reports includes new functionality that lets you create even more powerful formulas than ever before. With new time and data types, you can now create date formulas, time formulas, and date/time formulas. You can place arrays and ranges in variables, and you have a number of new functions that give you more control over your reporting than ever before. See Chapter 24, *Functions*, Page 569.

In-place editing makes it easy to edit OLE objects

Crystal Reports draws upon OLE technology to add solid new functionality to your reporting environment.

- All pictures are converted to static OLE objects. Double-click an object and the program displays the tools to edit that object, all while staying in Crystal Reports.
- While working in the Design Tab, you can convert OLE objects to a format that is supported by your editing tools. For example, if you have a *.gif picture and no *.gif editor, you can convert the picture to a Paintbrush format that can be edited in the paint program that ships with Windows. See Chapter 21, *OLE*, Page 527.

Dragging objects between reports cuts creation time

Now you can open multiple reports and drag many kinds of objects between them. Now there is no need to "reinvent the wheel" if you have an object in one report and you need to use it in a different report. Just drag them where they are needed.

You can also drag objects into Crystal Reports, from any application that is an OLE server application. Microsoft Word and Excel are examples of this kind of applications. Just highlight the object (such as text or worksheet cells) and drag it into the report. The object becomes part of your report. Double-click it and you can edit it in place using the application you created it with. See Chapter 22, *OLE*, Page 527.

HTML exporting simplifies Web activities

With Crystal Reports, you can now export your reports directly to HTML format. With this capability, you can publish your reports on the World Wide Web and on organizational intranets. Combining this capability with other Crystal Reports features, you can add dynamic reporting to your Web applications. Some benefits of exporting to HTML include:

- HTML report can be viewed by any Web browser.
- Specific Microsoft, Netscape and Oracle Web extensions are available when exporting to HTML.
- Reports are automatically converted to HTML, saving you time and tedious effort when preparing database information for the Web.

As the Internet and Intranets become more and more important to many businesses, Crystal Reports is providing the tools you need to take advantage of the new opportunities. See *HTML* in Crystal Reports online Help, and *How to export reports*, Page 110.

Sample reports demonstrate professional techniques

The program comes with professionally designed sample reports that demonstrate a wide range of reporting techniques. You can create your own reports using these reports as templates. Simply select a template, change the data pointers to your data, make whatever other modifications you see fit, and run the report. That's all there is to it. See *Crystal Reports Sample Reports* in online Help and the *Reports at a Glance* in your Crystal Reports program group.

New developer features make development easier, faster

Developers will find a lot to like in Crystal Reports. The product comes with a new OLE / ActiveX Control with 90 properties, more than 15 of them new. The Report Engine DLL has been redesigned to enhance performance, and it has an API with 110 calls, more than 20 of them new. There's also a VBX, a VCL for 16-bit Delphi development, an MFC Class Library for the Report Engine, and an NewEra Class Library for developing in that environment. Online Help has been restructured so you can find the information you need more efficiently, sample reports have been rewritten and expanded to help you better understand the interaction of your application and the Report Engine, and the Developer Manual is presented in a full text-searchable, books-online format. See Chapter 26, *Application Development with Crystal Reports*, Page 631, and search for *Developer Topics* in Crystal Reports online Help.

New database support improves data access

Crystal Reports now supports more databases than ever, and it provides new ways to access previously-supported data for enhanced performance. The program now ships with drivers for INFORMIX and Microsoft Exchange, and it includes more 32-bit drivers than ever before. It has native drivers for Sybase, SQL Server, and Oracle, and it supports the PeopleSoft PS Query. Connecting to your data has never been easier. See Chapter 19, *Working with Databases*, Page 403, and Chapter 20, *Data Sources*, Page 475.

3

Learning Crystal Reports

What you will find in this chapter...

- Learning Crystal Reports 34
- User's Guide 34
- Online Help 37
- Books Online 38
- Sample Reports 39
- Sample Applications 39
- Glossary 40
- Sample Data - CRAZE.MDB 40
- Suggested Learning Paths 43

Learning Crystal Reports

The purpose of this chapter is to:

- explore the various learning chapters in this manual,
- introduce the other tools that come with the program, and
- suggest learning paths based on your background.

Crystal Reports comes with a wide variety of tools and with a comprehensive sample database to help you learn the program and use it efficiently. This chapter describes each of those tools and the sample data as well.

User's Guide

While everything in this manual is provided to help you learn and use Crystal Reports, there are certain sections and chapters that are more useful than others as training tools. The chapters we do not discuss here may be more valuable as reference material, troubleshooting suggestions, and performance enhancements than for learning the program.

CHAPTER 1, INSTALLATION AND QUICK START GUIDE

The *Quick Start Guide* (Page 15) is a "shorthand" explanation to getting up and running with Crystal Reports. Intended for experienced users and new users who like to "learn by doing," the *Quick Start Guide* covers all of the key elements of working with Crystal Reports in a few short pages.

CHAPTER 2, WHAT'S NEW

Crystal Reports has changed dramatically for this release. This chapter identifies and describes the main new features, and it points you to sections of the manual where they are discussed in depth. Users who have used earlier versions of Crystal Reports will benefit the most from this chapter.

CHAPTER 4, GETTING TO KNOW CRYSTAL REPORTS

This chapter introduces you to Crystal Reports. In this chapter you will learn about the tools, the pointers, and the tabs you will use to design, preview, and analyze your reports. This chapter

explains what to do, and then shows you how to do it. Since the design environment has changed dramatically with this release, all levels of users will benefit from reading this chapter.

CHAPTER 5, TUTORIAL - CUSTOMER LIST

Chapter 5 is the primary tutorial for Crystal Reports. It is a product introduction that leads you step-by-step through the creation of a report and introduces you to many of the key features of the program in the process. The tutorial has been written for the new user. No prior knowledge of reporting is expected. By the time you finish the tutorial, you should have enough understanding of the program to feel comfortable getting started on your own report.

HANDS-ON SKILLS TUTORIALS

While the tutorial in Chapter 5 gives you an overview of the program and some hands-on experience, your real learning begins when you start working on your own project. You will not be working alone. The manual contains over 200 hands-on tutorials covering all of the reporting tasks you might typically need to perform. Most of the tutorials are show-as-you-go; they teach with pictures as well as words.

The tutorials are not cluttered with conceptual information or endless explanations. They are concise and to the point, showing you the easiest way to get something done. However, these tutorials will not leave you with unanswered questions either. Each tutorial points you to related topics and to background information you can read if you want to know more about what you are doing.

While the Hands-On sections have been written so that new users can understand them, they will benefit anyone who is trying to learn a specific skill.

CHAPTER 6, REPORTING 101

Reporting 101 concentrates on the basic concepts of report design, and then it explains, in easy terms, the things you should understand and consider in order to:

- select your database table(s),
- select fields,

- place objects on your report,
- sort, group, and total data,
- select records to be included in the report, and
- output your report to a printer or export it electronically.

This chapter is written for people new to reporting, but it contains a lot of background information that users at all skill levels may find useful. Armed with that kind of information you will be well-prepared to create powerful reports that run efficiently and that provide exactly the information you need.

CHAPTER 9, SORTING, GROUPING, AND TOTALLING

No other program has the sorting, grouping, and totalling capabilities of Crystal Reports. This chapter, written for the new user, provides a visual tour of the kinds of sorting, grouping, and totalling you can do, and then it shows you how. This chapter is an excellent overview for beginners who may not understand sorting, grouping, and totalling as well as advanced users who want to know more about sorting, grouping, and totalling capabilities.

CHAPTER 10, FORMULAS 101

You can do many things in Crystal Reports without using the formula language. But once you know how to create formulas, your reporting capabilities are virtually endless. *Formulas 101* gets you started creating simple formulas. It familiarizes you with the Formula Editor tools and leads you step-by-step through the formula creation process. This chapter has been written for people new to formulas and uses a lot of pictures to show you exactly how to perform each step. It is a chapter that is intended to get you beyond the mystery of formulas and into using them for your everyday reporting needs.

CHAPTER 11, ADVANCED FORMULAS

Advanced Formulas gets into the realm of "what is really possible" with Crystal Reports. It shows you how to create and use variables, how to "tweak" formula evaluation times so the formula is evaluated against the "right" data, and how to do some pretty fancy conversions. Written for the advanced user, this chapter shows you how to program Crystal Reports when you need it to do something out of the ordinary.

CHAPTER 19, WORKING WITH DATABASES

Understanding database concepts, relational database design, and performance considerations can help you get the most out of Crystal Reports. This chapter leads you through the basics, and it provides a detailed explanation of the way the program accesses linked data. The information in this chapter will help you optimize your reporting for maximum efficiency. This chapter contains information both for the beginner and the advanced user.

CHAPTER 20, DATA SOURCES

Crystal Reports works with all kinds of data, from simple text files to advanced client-server SQL databases. This chapter shows you how Crystal Reports connects to various data sources and what files have to be in place to make the connection. This is an in-depth chapter for advanced users who need to know how Crystal Reports works beneath the surface.

CHAPTER 26, APPLICATION DEVELOPMENT WITH CRYSTAL REPORTS

Crystal Reports provides developers with a free runtime license for its Report Engine (CRPE.DLL). This chapter provides an overview of how to use the Report Engine with your application, and it provides pointers to the many other sources of developer information. This chapter is written for those individuals who need to know how to use the Report Engine with applications they are developing.

APPENDIX A, REPORT PROCESSING MODEL

Crystal Reports uses a sophisticated multi-pass reporting model for processing reports. Understanding when different parts of the reporting process take place can help you design more efficient reports and solve reporting problems. This appendix is written for the advanced user.

Online Help

Online Help included with Crystal Reports is a superset of the manual. That is, it includes all of the key information from the manual and a lot more as well. Sample formulas, sample record and group selection formulas, sample code, and hundreds and

hundreds of topics on virtually anything related to reporting - online Help has it all.

If you want help to learn to use menu commands for example, online Help gives you an explanation of what the command does, then a thorough explanation of any dialog boxes the menu command activates, and finally, tutorials for performing tasks using the menu command. By working through the topics for any menu command, you can find out why you would use it and become an expert in its use.

You can traverse online Help by using a contents "tree", you can use the Index to look up topics by key words, or you can use the full-text search facility to find topics that contain the word of interest. It is suggested that you use online Help often while learning Crystal Reports.

Books Online

Books Online contains the full text of the User's Guide and a complete Developer's Reference in electronic format. Both online books are included with the CD version; only the Developer's Reference is included with the diskette version of Crystal Reports. These manuals have been created with Adobe Acrobat and Acrobat Reader has been included on the CD and setup disks to allow you to view or print these documents.

Books Online are hypertext documents with hundreds of jumps to take you to topics of interest, related topics, and technical information. The books are fully indexed with hypertext jumps to each of the indexed topics. Additionally, Acrobat Reader has a full text search capability so you can find any topic of interest if you can think of just one unique word it might contain. With electronic search and jump capabilities, Books Online give you the tools to find the information you need in a hurry.

You can set up Books Online in a variety of ways to suit your needs. Consult the Acrobat Reader Help System for a complete explanation of the options.

Sample Reports

Crystal Reports comes with many professionally designed sample reports in two categories:

- **General departmental reports**
cover a wide range of reporting needs for many of the departments in an organization.
- **Financial reports**
address the specific needs of financial institutions and corporate finance departments. (Financial reports are available with the Professional version of Crystal Reports only.)

The sample reports are valuable learning tools.

- By studying the content of the reports, you can see the kind of information that is needed and how it is presented.
- By studying the layout and design of the reports, you can see how the information is arranged for clarity.
- By studying the formatting of the reports, you can see how key information is emphasized and what was done to make the reports visually pleasing. See the *Crystal Reports Sample Reports* topic in online Help.

Sample Applications

Crystal Reports comes with a number of sample applications that show you how to incorporate the capabilities of the Crystal Report Engine. The following sample applications are included:

- One written in Visual Basic and making direct calls to the Report Engine.
- One written in Visual Basic and accessing the Report Engine using the Crystal Reports VBX.
- One written in Visual Basic and accessing the Report Engine using the Crystal Reports OCX.
- One written in Delphi and accessing the Report Engine using the Crystal Reports VCL.

By studying these applications, you can learn how the code blocks can be structured and where they should be located to generate the report output capabilities you want for your application. (See the *Crystal Reports Sample Applications* topic in online Help.)

Glossary

Crystal Reports comes with a comprehensive glossary. The glossary explains basic database and reporting concepts as well as issues specific to Crystal Reports. Unlike some glossaries which offer little substance, this glossary gives in-depth explanations of terms from the Crystal Reports perspective.

- All glossary words are printed in green the first time they appear in an online Help topic. When you click the word, the definition appears in a pop-up window.

Reading the glossary is an excellent warm-up before you start working with the other training aids.

Sample Data - CRAZE.MDB

Crystal Reports comes with CRAZE.MDB, a database of sample data you can use when learning the program. CRAZE.MDB is a Microsoft Access 2.0 database and all of the necessary drivers are included with the program. You should be able to open the database directly and begin designing reports. Virtually all of the examples in this manual are based on CRAZE.MDB data.

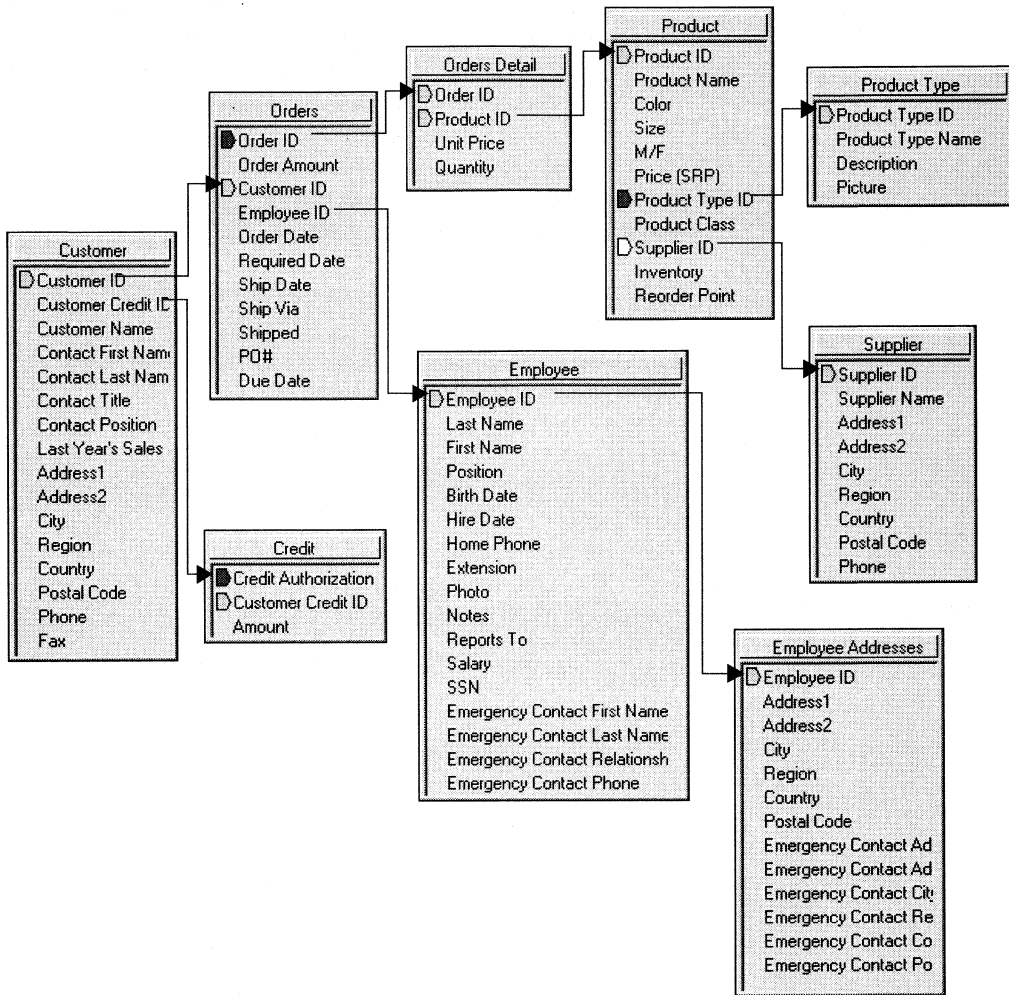
CRAZE.MDB is a database that contains data for Craze Mountain Bikes, a fictitious manufacturer of mountain bikes and accessories.

The database includes the following tables:

- **Craze Info**
Company data on the Craze Mountain Bikes company including the company logo in both color and black and white.
- **Credit**
Information on customer credit memos.

- **Customer**
Data on the customers served by the company.
- **Employee**
Company-oriented data on the employees of Craze Mountain Bikes.
- **Employee Addresses**
Personal data on company employees.
- **Orders**
Identifying and tracking data for company orders.
- **Orders Detail**
Line item data for company orders.
- **Product**
Descriptive data on company products.
- **Product Type**
Category data on company products, including product pictures.
- **Supplier**
Data on the suppliers who serve the company.

The links (relationships) between the tables are as follows:



CRAZE.MDB also includes a query (Top Customers) and a parameter query (Credit Limits) that you can use to learn how to report on those kinds of data sets.

NOTE: The sample data also contains the *Craze Info* table (not pictured) which is not linked to any other table. This table contains company information on the Craze Mountain Bike Company.

NOTE: The sample data has been designed to illustrate various reporting concepts in a training environment, not to teach database design. While there are alternative ways of designing a database, this design was selected to keep the tutorials and examples focused on reporting, not on data manipulation.

Suggested Learning Paths

There is no "correct" learning path for everyone; we all have our own needs, and we all learn in our own way and at our own speed. What follows are simply "suggested" learning paths for several kinds of users.

The order of the elements in each path is a suggested order, yet one that has been carefully structured to speed the learning process. Feel free to modify any of these paths at your discretion.

New user (business)

The following learning path is suggested for new users who expect to use Crystal Reports on a casual basis.

- Give the *Glossary* a quick read.
- Read Chapter 4, *Getting to Know Crystal Reports*.
- Work through Chapter 5, *Customer List Tutorial*.
- Read Chapter 6, *Reporting 101*.
- Scan Chapter 9, *Sorting, Grouping, and Totaling*.
- Read Chapter 10, *Formulas 101*.
- Review the *Sample Reports*.
- Refer to online Help and the Hands-On skills tutorials as needed.

Business user upgrading from earlier version

The following learning path is suggested for users who are upgrading from an earlier version of Crystal Reports.

- Scan the *Glossary* for new terms.
- Read Chapter 2, *What's New*.
- Read Chapter 4, *Getting to Know Crystal Reports*.

- Read Chapter 6, *Reporting 101*.
- Scan Chapter 9, *Sorting, Grouping, and Totaling*.
- If you have limited formula experience, read Chapter 10, *Formulas 101*.
- If you have a working knowledge of formulas, see Chapter 11, *Advanced Formulas*.
- Review the *Sample Reports*.
- Refer to Hands-On skills tutorials as needed.

New power user

The following learning path is suggested for new users who expect to use many of the sophisticated features of Crystal Reports.

- Read the *Quick Start Guide* (starting on Page 15).
- Read Chapter 4, *Getting to Know Crystal Reports*.
- Scan Chapter 6, *Reporting 101*.
- Scan Chapter 9, *Sorting, Grouping, and Totaling*.
- Scan Chapter 10, *Formulas 101*.
- Read Chapter 11, *Advanced Formulas*.
- Read *Performance Considerations* in Chapter 19, *Working with Databases*, Page 419.
- Read Appendix A, *Report Processing Model*.
- Review *Specialized Formulas* in online Help.
- Refer to the *Glossary* and Hands-On tutorials as needed.

Power user upgrading from earlier version

The following learning path is suggested for power users who are upgrading from an earlier version of Crystal Reports.

- Read Chapter 2, *What's New*.
- Read Chapter 4, *Getting to Know Crystal Reports*.
- Read Chapter 11, *Advanced Formulas*.
- Read *Performance Considerations* in Chapter 19, *Working With Databases*, Page 419.

- Read Appendix A, *Report Processing Model*.
- Review *Power Formulas* in online Help.
- Refer to the *Glossary* and Hands-On tutorials as needed.

New MIS user

The following learning path is suggested for MIS (Management Information Systems) professionals who are using Crystal Reports for the first time.

- Read the *Quick Start Guide* (starting on Page 15).
- Read Chapter 4, *Getting to Know Crystal Reports*.
- Scan Chapter 6, *Reporting 101*.
- Scan Chapter 9, *Sorting, Grouping, and Totaling*.
- Scan Chapter 10, *Formulas 101*.
- Read Chapter 11, *Advanced Formulas*.
- Read Chapter 18, *Dictionaries*.
- Read *Performance Considerations* in Chapter 19, *Working with Databases*, Page 419.
- Read Chapter 20, *Data Sources*.
- Read Appendix A, *Report Processing Model*.

MIS user upgrading from earlier version

The following learning path is suggested for MIS users who are upgrading from an earlier version of Crystal Reports.

- Read Chapter 2, *What's New*.
- Read Chapter 4, *Getting to Know Crystal Reports*.
- Scan Chapter 9, *Sorting, Grouping, and Totalling*.
- Read Chapter 11, *Advanced Formulas*.
- Read Chapter 18, *Dictionaries*.
- Read *Performance Considerations* in Chapter 19, *Working with Databases*, Page 419.
- Read Chapter 20, *Data Sources*.
- Read Appendix A, *Report Processing Model*.

Developer using Crystal Reports for the first time

The following learning path is suggested for developers who are integrating Crystal Reports with their applications for the first time.

- Read Chapter 26, *Application Development with Crystal Reports*.
- Review the *Sample Applications*.
- Read the *Quick Start Guide* (starting on Page 15).
- Read Chapter 4, *Getting to Know Crystal Reports*.
- Read *Performance Considerations* in Chapter 19, *Working with Databases*, Page 419.
- Read Appendix A, *Report Processing Model*.
- Refer to the Hands-On tutorials, online Help for Developers, and the online Developer's Manual as needed.

Developer upgrading from earlier version

The following learning path is suggested for developers who have previously integrated Crystal Reports with their applications and who are upgrading from an earlier version.

- Read Chapter 26, *Application Development with Crystal Reports*.
- Review the *Sample Applications*.
- Read *Upgrading from an earlier version of Crystal Reports* (starting on Page 15).
- Read Chapter 2, *What's New*.
- Read Chapter 4, *Getting to Know Crystal Reports*.
- Read *Performance Considerations* in Chapter 19, *Working with Databases*, Page 419.
- Read Appendix A, *Report Processing Model*.
- Refer to the Hands-On tutorials, online Help for Developers, and the Developer's Manual online as needed.
- Refer to *Runtime File Requirements* in online Runtime Help to identify the files you need to distribute with your applications.

4

Getting to Know Crystal Reports

What you will find in this chapter...

The Crystal Reports Window 48

The Menu Bar 48

The Toolbar 51

The Format Bar 53

The Status Bar 54

Shortcut menus 56

Crystal Reports cursors 58

Crystal Reports Design Tab 60

Crystal Reports Preview Tab 65

Other fundamentals 68

How to move the toolbar and format bar 72

How to add, delete and move guidelines 73

How to move and position objects using guidelines 75

How to turn the grid on/off 78

How to zoom your report in and out 79

How to drill down on summarized data 81

How to make an object underlay a following section(s) 100

How to undo/redo activities 80

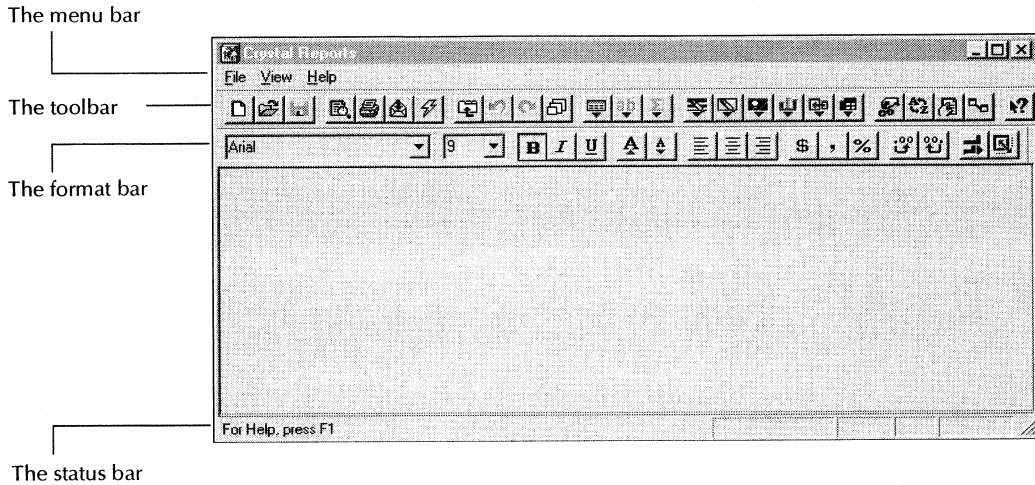
How to add, delete, move, and merge sections 83

How to split and resize sections 86

How to hide parts of the report 103

The Crystal Reports Window

The Crystal Reports window is clear and easy to understand.



- The title bar identifies the current report and it contains the standard Windows buttons.
- The menu bar appears just below the title bar.
- The toolbar appears just below the menu bar.
- The format bar appears just below the toolbar.
- The status bar appears at the bottom of the window.

The Menu Bar

The menu bar is the command center of Crystal Reports. Each option on the menu bar activates a drop down menu of commands that you can use to create, modify, print, and save your reports.

File menu

The File menu includes commands you can use to open, close, and save files, to save files under a different file name, and create new reports, Mailing Labels, and Cross-Tabs, as well as several other

kinds of reports. It also includes a command you can use to exit Crystal Reports. Additionally it contains commands that enable you to preview your report before printing, to export your report to a disk file in a variety of formats, to send your report to a printer, to change page margins, and to select a printer if you want the report to print on something other than the default printer. You can also add summary information to help you identify your report, and you can change the default settings so the program works the way that is the most efficient for you. See *File menu Commands*, Page 534.

Edit menu

The Edit menu allows you to modify aspects of your report. It includes commands to undo and redo actions; to edit fields, formulas, summaries, OLE objects, and OLE links; to view a sample of the data in a selected field; and to cut, copy, and paste objects and OLE objects. You can also use the Edit menu to show, hide, move, merge, or delete report sections and to convert static OLE objects to editable bitmaps. See *Edit menu commands*, Page 539.

View menu

The View menu includes commands you can use to modify the user interface of Crystal Reports. It enables you to show or hide the toolbar, format bar, and status bar, to zoom in and out on your report to view it at different magnifications, and to turn guidelines and the grid on and off. See *View menu commands*, Page 544.

Insert menu

The Insert menu is the central menu you use for creating reports. It includes commands you can use to insert database fields, text objects, formula fields, parameter fields, Cross-Tab objects, subtotals, grand totals, summaries (counts, averages, etc.), groups, sections, and several special fields such as print date and page number. The Insert menu also includes commands that enable you to insert group name fields, subreports, graphics, lines, boxes, graphs, and OLE objects into your report. See *Insert menu commands*, Page 546.

Format menu

The Format menu includes commands for changing the look of the elements in your report. It includes commands for changing fonts and for adding field borders, background color, and drop shadows. The Format menu has commands for formatting fields

that are embedded in text objects and for formatting individual paragraphs in those text objects as well. Finally, using this menu, you can have the program automatically arrange report objects, apply professionally designed styles to the entire report, and move objects in a stack of objects forward and backward. See *Format menu commands*, Page 553.

Database menu

The Database menu is used to add and delete tables for use with your reports, to change the alias you use to identify a table, and to link and unlink tables. It has commands for logging on and off SQL and ODBC servers, for showing and editing SQL queries, and modifying parameters for stored procedures. The Database menu also has commands that direct the program to look for tables in new locations, to remove tables from a report and to adapt your reports appropriately if there are minor changes in table structure. See *Database menu commands*, Page 557.

Report menu

The Report menu includes commands that let you select the records or groups to be included in your report, select the order in which report data is to be sorted (by record or by group), specify subreport links, and specify a print date for your report. It has commands for updating the data used in a report and for gathering all the files you need for distributing your reports. You can also use one of the Report menu commands to create an executable version of your report that you can share with others that do not have Crystal Reports. See *Report menu commands*, Page 560.

Window menu

The Window menu includes commands that let you rearrange icons and windows. It also lists the report windows that are open and includes a command that lets you close all report windows at once, if desired. See *Window menu commands*, Page 564.

Help menu

The Help menu includes commands that take you to Crystal Reports online Help index and search facility that provide context sensitive Help. It has commands for registering your version of Crystal Reports, for calling up technical information about your computer system, and for creating a technical support request. The Help menu also has commands that provide information on Crystal Reports LANPAKS, the Professional version of the

program, and Upgrade Express, the most efficient way to stay on top of program upgrades. One final command, ABOUT CRYSTAL REPORTS, gives you information about the Crystal Reports version you are using. See *Help menu commands*, Page 565.

The Toolbar

Crystal Reports groups several commonly used commands on a toolbar that remains on screen at all times (unless you choose to turn it off using the TOOLBAR command on the View menu). Each button displays a graphic that visually describes the command it represents. You activate a toolbar command by clicking the appropriate button.

The toolbar eliminates some of the steps needed to activate the included commands and can thus greatly speed your work in creating reports. Each command available via the toolbar is discussed in Crystal Reports online Help. (Search for each command by name.)

NOTE: In the 32-bit version of Crystal Reports, you can move the toolbar to another fixed location in the window or set it up as a floating palette.

The buttons on the toolbar perform the following functions:



Create a new report.



Open an existing report.



Save your report.


















Preview your report in the Preview Tab.



Print your report to a printer.



Export your report to a file or e-mail.

-  Refresh report data.
-  Activate the Report Expert.
-  Undo an action.
-  Redo an action.
-  Zoom in and out on your report.
-  Insert database, formula, parameter and group name fields.
-  Insert a text object.
-  Insert a summary.
-  Insert a line.
-  Insert a box.
-  Insert a picture.
-  Insert a graph/chart.
-  Insert a subreport.
-  Insert a Cross-Tab object.
-  Set selection criteria.



Set record sort order.



Format any section of your report.

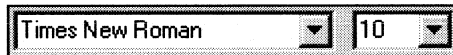


Define links between tables.

The Format Bar

The format bar enables you to select many popular formatting options with the click of a button or a selection from a drop-down list box. You simply select the data you want to format, then click the appropriate button or list option to format the data.

Two drop-down list boxes appear at the left side of the format bar.



Use these drop-down list boxes for selecting new fonts and font sizes if you want to use something other than the default. To make a selection from either box, click the arrow to reveal your options and then click the option you want.

The buttons on the format bar perform the following functions:



Change the selected data to boldface.



Italicize the selected data.



Underline the selected data.



Increase the font size of the selected data one point each time you click the button.



Decrease the font size of the selected data one point each time you click the button.



Align the selected data flush left.



Center the selected data.



Align the selected data flush right.



When a number field is selected, places a currency symbol with the number.¹



When a number field is selected, places a thousands separator in the number.¹



When a number field is selected, places a percentage sign with the number.¹



When a number field is selected, adds one decimal place to a number.¹



When a number field is selected, subtracts one decimal place from a number.¹



Applies a professionally-designed style to the entire report.



Arranges field objects automatically on the entire report.

¹The program refers to your setting in the International section of the Control Panel (Windows 3.x, Windows NT) or the Regional section of the Control Panel (Windows 95).

The Status Bar

The status bar at the bottom of the Crystal Report window displays valuable information to help you use the program more efficiently.

TOOLBAR/FORMAT BAR FUNCTIONS

When the cursor is over a toolbar or format bar button, the status bar displays a short description of the button's function.

MENU COMMAND DESCRIPTIONS

When you highlight a menu command, the status bar displays a short description of the command.

CURRENT SELECTIONS

When you select or place an object, the status bar displays the name of the object (or its object type) plus its location and sizing information.

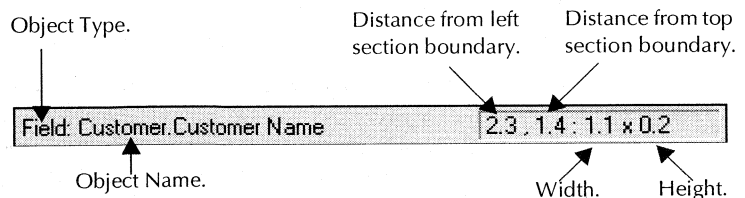
To identify objects, it displays:

- the word *OLE* for a picture or other OLE object,
- the alias and field name for a field,
- the words Line for a line, Box for a box, and Text for a text object,
- the field type for special fields (Print Date, Record Number, and so forth),
- the summary name for a summary or subtotal,
- the formula name for a formula, and
- the parameter field name for a parameter field.

It also displays

- how far the object is from the left and top boundaries of a section, and
- how big the object is.

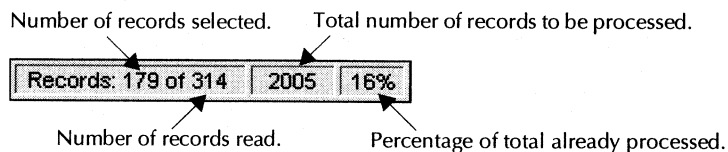
Using this information, you can easily move and resize objects with precision.



DATA RETRIEVAL INFORMATION

When you preview your report for the first time or refresh the data in the Preview Tab, the status bar displays three figures that describe the data retrieval processing.

- The first number defines the number of records selected out of the number of records read.
 - If your report has a selection formula, the records selected figure will typically be smaller than the records read figure.
 - If your report does not have a selection formula, records selected will always be equal to records read.
- The second number displays the total number of records that will be processed.
 - If your report is based on a single table, the figure that is displayed should remain constant.
 - If your report is based on linked tables, and if any one-to-many situations exist, the figure will typically increase as the program identifies all of the linked records.
- The final number is the percentage of the total records that have been processed.



Once the report has finished processing, Crystal Reports displays only the number of records selected and the percentage processed.

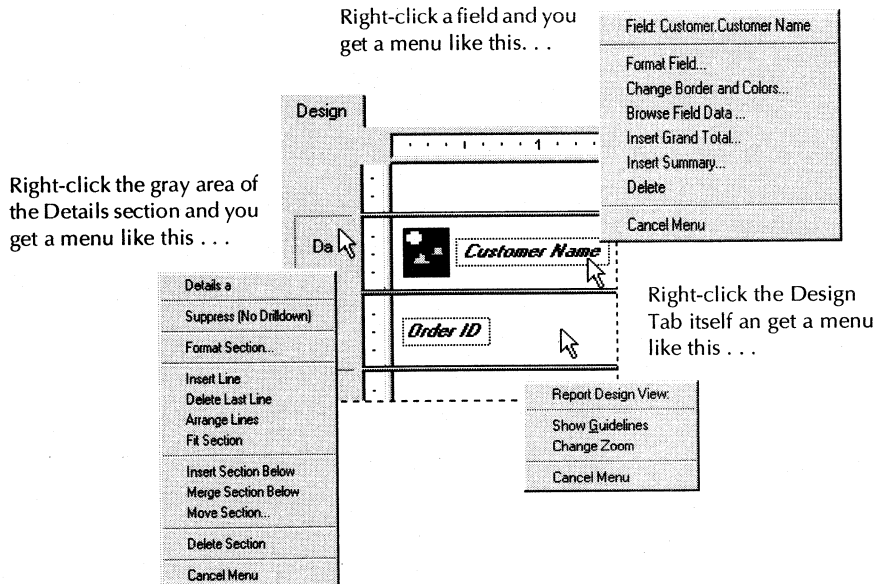
Shortcut menus

When you are working in either the Design or Preview Tab, you can speed up your work considerably using Crystal Reports' shortcut menus. When you right-click a report element (a picture, section, field, etc.), Crystal Reports displays a shortcut menu next to the element. Unlike the program's standard menus that group

commands by function (editing, inserting, etc.), these shortcut menus are element-specific: that is, they contain only those commands from Crystal Reports' primary menus that are available for use with the selected element.

The shortcut menus are valuable because:

- They display the name and source (alias) of the element at the top of the menu (also on the status bar) so you can identify the elements on your report with a single click.
- They make it easier to learn Crystal Reports because they eliminate the need to remember where to find a command.
- They make this more efficient because you are dealing with only a compact list of commands which make it easier to pick the right one.
- They spotlight the things you can do with an element making an easier system to use when you are under pressure or distracted.



Crystal Reports cursors

Crystal Reports uses a number of different cursors at different points in its operation:



The Arrow cursor is the primary cursor. It is used for making menu selections, selecting options from dialog boxes, working with scroll bars, clicking buttons and objects, and so on.



The Double-Arrow cursor is a resizing cursor. It changes to one of a number of different double-arrow cursors whenever it is over a resizing handle on a bit-mapped graphic, a graphic box, a graphic line, or a field.



The I-beam cursor is active whenever you are working with text (editing text objects, creating formulas, typing text in a dialog box). It is the cursor you use to highlight text and to set the position of the insertion point.



The Pencil cursor is a drawing cursor. It appears whenever you choose the **BOX** or **LINE** command from the **Insert** menu. The point of the pencil marks the spot where the drawing begins and is used to define the size and shape of the object drawn.



The Drag (or Stop) cursor. It appears whenever the item you are dragging is over an area in which it cannot be dropped. For example, dragging a **Cross-Tab** object into a section where it cannot be placed.



The single-unit Drag and Drop cursor is available whenever you are dragging a single item over an area where it can be dropped.



The multi-unit Drag and Drop cursor is available whenever you are dragging multiple items over an area where they can be dropped.



The Tiny Hand cursor is available only in the online Help facility. The Arrow cursor changes to the Tiny Hand cursor whenever it is positioned over text or a graphic that you can use to jump to another topic in online Help.



The Section Sizing cursor. The Arrow cursor changes to the section sizing cursor whenever it is positioned over the bottom boundary of any of the report sections. Using this cursor you can drag a section boundary line to expand or reduce the size of a section.



The Insertion Point cursor identifies the location for entering text. It is available in text objects, the Formula Editor, and some of the dialog boxes. You set the insertion point by positioning the cursor and then clicking.



The Hourglass cursor is the cursor that appears whenever Crystal Reports is processing a command you have chosen. Whenever the hourglass is visible, you cannot select any other commands or proceed further with your report.



The Frame cursor is the cursor you will use for placing fields and formulas in the Design Tab. It is a graphic approximation of the size of the object that Crystal Reports uses to represent objects in the Design Tab.



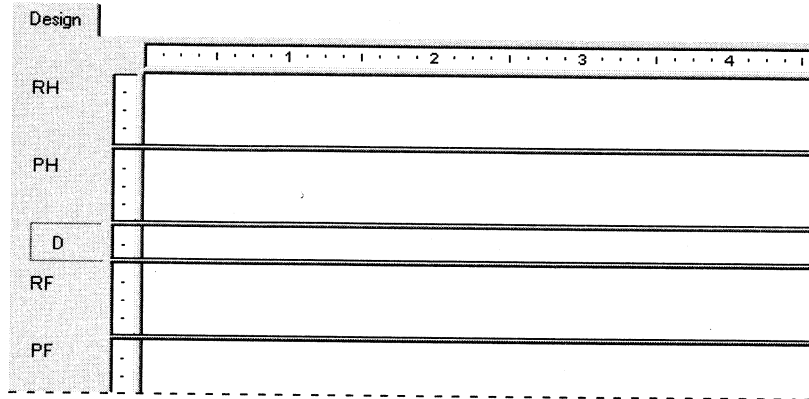
The Drill Down cursor appears when the pointer is positioned over a summary value or a graph in the Preview Tab. When you double-click it on a summary value or graph element, the program displays the details behind the summary.



The Section Splitting cursor. It appears when you position the pointer over the left boundary of a report section. When you click, a horizontal line appears that you position where you want to split the section.

Crystal Reports Design Tab

When you are working with Crystal Reports, you will probably find yourself using the Design Tab more than any other part of the program.



The Design Tab is the place you do most of your initial work when creating a report. It provides you with simulated page width bands that designate the various areas of your report. You can place objects in the sections where you want them to appear, specify your sorting, grouping, and totalling needs, do your initial formatting, and so forth.

The Design Tab provides the most efficient environment for designing your report because you work on the tab with data representations, not data itself. When you place a field on the report, the program uses a frame to identify the field on the tab; it does not retrieve the data. Thus, you can add and delete fields and other objects, move them around, set up complex formulas and more, without tying up the computer time or network resources it takes to gather the data.

The report you create in the Design Tab is a kind of virtual report; it has the structure and the instructions for creating the final report, but it is not the report itself. To turn the Design Tab report into a final report or into a report that you can fine tune, you "just add data." You do this whenever you preview the report, print it, or output it in any other way. The actual data will now appear in the report.

Design Tab Areas

When you first begin creating a report, Crystal Reports automatically creates five areas in the Design Tab.

- **Report Header**

This section is generally used for the report title and other information you want to appear at the beginning of your report. It can also be used for graphs and Cross-Tabs that include data for the entire report.

- **Page Header**

This section is generally used for information that you want to appear at the top of each page. This can include such things as chapter names, first/last topic to appear, the name of the document, and other similar information. You can also use this section to display field titles above the fields on your report.

- **Details**

This section is used for the body of the report. The bulk of your report data will generally appear in this section.

- **Report Footer**

The Report Footer section is used for grand totals, for information you want to appear only once at the end of the report, and for graphs and Cross-Tabs that include data for the entire report.

- **Page Footer**

This section usually contains the page number and any other information you want to appear on the bottom of each page.

If you add a group, a summary, or a subtotal to your report, the program creates two additional sections:

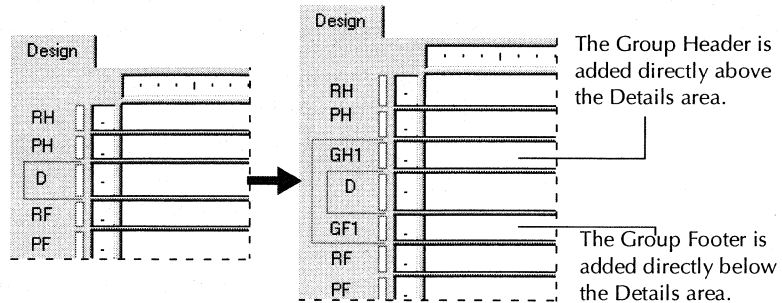
- **Group Header**

This section typically holds the group name field, and it can be used to display graphs or Cross-Tabs that include data specific to the group.

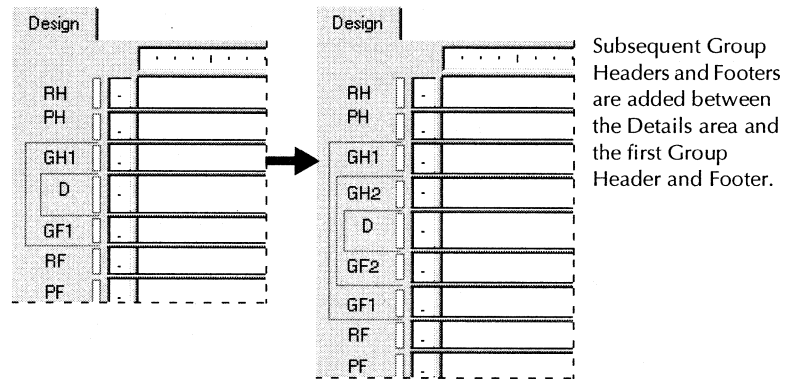
- **Group Footer**

This section generally holds the summary value, if any, and it too can be used to display group-specific graphs or Cross-Tabs.

When you add a group, a summary, or a subtotal, the Group Header area appears directly above the Details area and the Group Footer area appears directly below the Details area. See Chapter 9, *Sorting, Grouping, and Totaling*, Page 47.



If you set up additional groups, the program creates new group areas between the Details area and the existing Group Header and Group Footer area(s).



Like the original areas, each of these areas can contain one or more sections. By default they each contain a single section.

Area printing characteristics

Each report area has its own printing characteristics. It is important to understand these characteristics because they affect *when* and *how often* different report objects get printed.

WHEN AREAS PRINT

Areas print in the order they appear on the Design Tab (top to bottom). If there is more than one section in an area, the sections print in the order they appear within the area. Thus, if you have three Report Header sections, all three of those sections will print, in order, before the section(s) in the Page Header area begin to print.

HOW OFTEN OBJECTS PRINT

Your decision on where to place objects on the Design Tab is made easier if you understand how often each of the areas prints. Once you understand this, most of your reporting decisions are straightforward. This information becomes most useful, however, when you are trying to decide where to place graphs, Cross-Tabs, and formulas to get specific results.

Objects print in the following ways:

- Objects placed in the Report Header area print once, at the beginning of the report.
 - Graphs and Cross-Tabs placed in this area contain data for the entire report.
 - Formulas placed in this area are evaluated once, at the beginning of the report.
- Objects placed in the Page Header area print at the beginning of each new page.
 - You cannot place graphs or Cross-Tabs in this section.
 - Formulas placed in this area are evaluated once per page, at the beginning of each new page.
- Objects placed in the Group Header area print at the beginning of each new group.
 - Graphs and Cross-Tabs placed in this area contain data just for the group.
 - Formulas placed in this area are evaluated once for each group, at the beginning of the group.
- Objects placed in the Details area print with each new record.
 - You cannot place graphs or Cross-Tabs in this area.
 - Formulas placed in this area are evaluated once for each record.
- Objects placed in the Group Footer area print at the end of each group.
 - Graphs and Cross-Tabs placed in this area contain data just for the group.

- Formulas placed in this area are evaluated once for each group, at the end of the group.
- Objects placed in the Report Footer area print once at the end of the report.
 - Graphs and Cross-Tabs placed in this area contain data for the entire report.
 - Formulas placed in this area are evaluated once, at the end of the report.
- Objects placed in the Page Footer area print at the bottom of each page.
 - You cannot place graphs or Cross-Tabs in this area.
 - Formulas placed in this area are evaluated once per page, at the beginning of each new page.

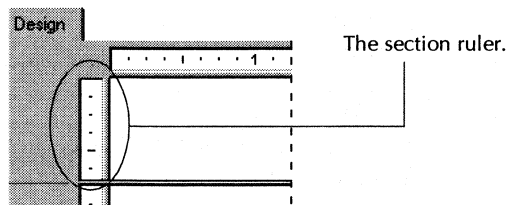
Identifying and working with areas and sections

By default, each area contains only a single section. The name for that section appears directly to the left of the section. If you have multiple sections in an area, the sections are designated as a, b, c, and so forth.

NOTE: Initials, such as RH, PH, D, PF, RF, and so on, are used to identify each section if you have activated the Short Section Names option in the File Options dialog box.

NOTE: If you right-click the shaded area containing a section name, a shortcut menu appears with section-specific options. If you right-click the shaded area to the left of the section names, a shortcut menu appears with area-specific options.

The program displays a section ruler immediately to the left of each section. You use the section ruler to add, remove, and move guidelines and to provide a visual reference when you are placing objects.



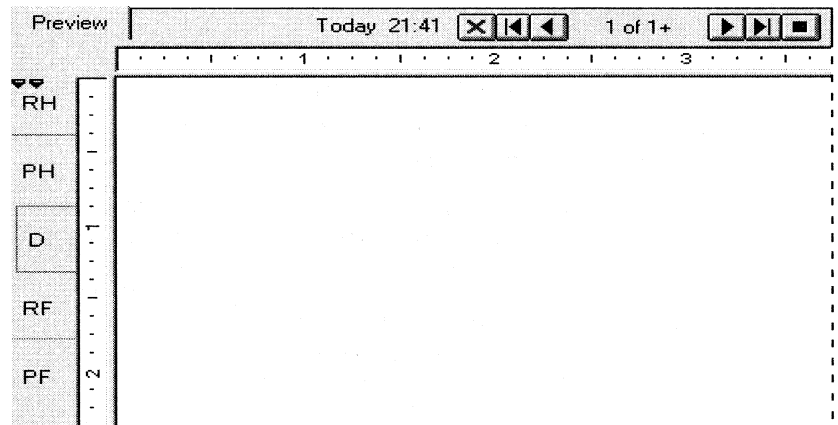
Whenever you add a new section, the program creates a ruler for that section. See *How to add, delete, move and merge sections*, Page 83.

Other Design Tab capabilities

There are several hidden capabilities built into the Design Tab.

- You can resize a section by dragging its bottom boundary. See *How to split and resize a section*, Page 86.
- You can split a section (create two sections from one) by clicking its left boundary. See *How to split and resize a section*, Page 86.
- You can add horizontal and vertical guidelines by clicking on the rulers. See *How to add and delete guidelines*, Page 75.
- You can zoom in and out on your report at any magnification from 25% to 400% of the original size. See *How to zoom your report in and out*, Page 79.

Crystal Reports Preview Tab



When you want to preview your report before printing, click the PRINT PREVIEW button. Crystal Reports gathers the data, makes the necessary calculations, and displays the report in the Preview Tab on electronic "paper." With the data in place, you can review the spacing and formatting of your report and see the actual results of all your summaries, formula calculations, and record and group selections.

In true WYSIWYG (What You See Is What You Get) fashion, you can work directly on this live data, fine tuning it until the report has the exact look you want.

NOTE: The program works with data in the following manner:

- ***The first time you use the Preview Tab it retrieves data from your underlying data source(s) and saves it with the report (unless you have set up the program not to save data).***
- ***From that point on, the program uses the saved data whenever you preview the report unless you specifically refresh it or add a field that requires the program to retrieve new data.***

Similarities/ Differences from Design Tab

You have the same formatting capabilities in the Preview Tab as you did in the Design Tab. Crystal Reports menus (both menu bar and shortcut), the toolbar, and the format bar remain active, giving you essentially the same functionality you have when working with a report in the Design Tab. However, when you are making many changes, it is quicker to make the changes in the Design Tab.

The Design Tab and Preview Tab are tied together internally. Any changes you make in one are reflected in the other.

The Preview Tab has many similarities to the Design Tab and some differences as well.

- The Preview Tab has a horizontal ruler at the top of the tab as does the Design Tab. Both rulers work in a similar fashion.
- The Preview Tab has a single vertical ruler at the left of the tab instead of the individual section rulers like the Design Tab. The functionality of the ruler is the same.
- The Preview Tab identifies report sections in the shaded area to the left of the data. With a quick look you can tell which report section the data is printing from. While section names appeared only once in the Design Tab, they print each time a section prints in the Preview Tab.
- The Record counter, the Data Age indicator, and the Page Forward/Page Back controls are all active in the Preview Tab. See *The Data Age indicator*, Page 67.

- The Preview Tab highlights every value when you select a field whereas only the field frame is highlighted in the Design Tab.

A DIFFERENT FEEL

Working the Preview Tab has a different feel than working in the Design Tab.

Each field in a database contains dozens, hundreds, or even thousands of values, depending on the number of records in the database. When you place a field in the Design Tab, a single field frame represents all those values. When you select the field, sizing handles appear on the frame and the frame changes color.

On the Preview Tab, however, you are working with the actual data. Instead of a field frame representing many field values, the values themselves appear.

- When you select a field or formula field value, you are actually selecting every value in the field.
 - The program places a sizing frame around the specific value you select.
 - It highlights every other value from the field.
- Likewise, when you select a summary value, you are actually selecting all of the related summary values.
 - The program places a sizing frame around the specific value you select.
 - It highlights all the related summary values.

Aside from the obvious appearance differences, the process of building and modifying a report is the same in both the Design Tab and the Preview Tab. You should find it easy to work with your reports in both places.

The Data Age indicator

Crystal Reports saves the underlying data with the report the first time you preview the report. From that point on, it uses the saved data every time you preview the report unless you refresh the data. The Data Age indicator indicates the date the data was last refreshed or initially retrieved, whichever is the most recent. If the data was initially retrieved or refreshed today, it indicates the time it happened. See *Save Data with Report command*, Page 535.

Other fundamentals

The Crystal Reports reporting environment is extremely flexible.

- You can turn on grid snap, set the grid to a maximum of up to 1", and make the grid visible or invisible on the Design Tab, the Preview Tab, or both.
- You can also work without a grid, placing your objects wherever you want them on your report.
- Finally, you can use guidelines if you wish to align and resize objects with precision.

You set up your environment so it works the way you work best.

Grid

The grid is a series of row and column coordinates. When the grid is active, Crystal Reports enables you to place objects only at these coordinates, not between them. In this way it makes it very easy for you to place and space data on your report and to align objects as needed. If you attempt to place an object between grid coordinates, the program "snaps" the object to the grid, that is, it moves the object automatically to the nearest set of row / column coordinates.

You activate the grid and specify its size and visibility properties using the `OPTIONS` command on the File menu. By default, the grid is not active. See *How to turn the grid on/off*, Page 78.

Freeform

Unlike earlier versions of Crystal Reports, in this version you can work without a grid, in a freeform environment similar to that of a drawing program. Freeform means simply that you can place objects anywhere you want them to appear on your report. Your only restriction is that the program will not allow you to place graph and Cross-Tab objects in the Page Header, Page Footer, or Details sections.

To work in a freeform environment, you toggle the *Snap To Grid* check box off on the Layout Tab of the File Options dialog box. Search for *File Options dialog box* in Crystal Reports online Help.

Freeform with guidelines

You may want to work in a freeform environment yet still have the ability to align objects, or to move or resize them as a group. You can do this using guidelines.

Guidelines are lines that extend vertically or horizontally from the Design and Preview Tab rulers. Guidelines have a snap property; when you move an object within a guideline's magnetic range, the object snaps to or attaches itself to the guideline.

- Once an object is snapped to a guideline, when you move the guideline, the object moves too.
- If you have several objects snapped to a guideline, they all move when you move the guideline.
- If you have several objects snapped to a guideline on two sides (right and left, or top and bottom) and you move one of the guidelines, you resize all of the objects similarly.

Using guidelines in a freeform environment gives you flexibility with control. See *How to move and position objects using guidelines*, Page 75.

Sections and objects

Crystal Reports enables you to insert a variety of objects in your report:

- **Field objects**
Fields from database tables and from the result sets returned by formulas, parameter, group name, queries, and stored procedures.
- **Text objects**
Characters, words, even entire documents. See *How to work with text objects*, Page 168.
- **Picture objects**
Bitmaps - *.bmp, *.gif, *.pcx, *.tif, *.tga. See *How to insert a picture*, Page 97.
- **Graph/chart objects**
Graphs that display summarized data. See Chapter 21, *Graphing*, Page 521.
- **Subreport objects**
Reports within reports, freestanding or bound to the data in the primary report. See *How to insert a subreport*, Page 340.

- **Cross-Tab objects**

Spreadsheet-like reports that help identify trends. See Chapter 16, *Cross-Tab Objects*, Page 351.

- **OLE objects**

Pictures, spreadsheets, text, and other objects created in OLE server applications. See Chapter 22, *OLE Objects*, Page 528.

Objects are containers. They can hold data, and in some cases, other objects (for example, a text object can contain field objects as well as text, and labels in a Cross-Tab object are actually text objects). Each object has properties that define the way the object acts in your report.

You can set fixed properties for objects, conditional properties, or a combination of the two. You set fixed properties using dialog box options. You set conditional properties using special formulas. See *Conditional Formatting*, Page 188.

You can insert most objects in most report sections. But the program restricts you from placing some objects in some sections because it never makes sense to place them there. For example, since a Details section prints with each record, a Cross-Tab object placed in a Details section would produce a Cross-Tab report for each record, not something that would be very useful. The program thus excludes Cross-Tab objects from the Details section. See *Area Printing Characteristics*, Page 62, for a summary of section/ object restrictions.

NOTE: See *How to make an object underlay a following section(s)*, Page 100, for information on printing objects in sections where they can not be physically placed.

Because it never makes sense to put some objects in some sections, and because the program prevents those kinds of insertions, you never have to worry about putting an object where it never belongs. But just because you can put an object in a section does not necessarily mean that it makes sense to put it there. That is a different situation that requires some judgement on your part. For example, if you put a picture object:

- in a Report Header section, it prints once at the beginning of the report.
- in a Group Header section, it prints once with every group.

- in a Details section, it prints once with every record.

Based on what you are trying to accomplish in the report, it clearly makes sense to put the object in one of the sections and not in the others. It is up to you to decide what is best for your report.

You can also set fixed and conditional properties for sections just as you can for objects.

Underlaying Objects

By default, when you place an object into a section:

- the section expands to accommodate the object, if necessary, and
- the object prints in the section where it is placed, whenever that section prints.

However, when you place an object in a section that you have set to underlay the following sections, however:

- the object still prints when the section it is placed in prints, but,
- it underlays the following section(s) as well.

NOTE: A section can underlay all sections up to (but not including) its "sister" section. For example, the Page Header section can underlay all sections up to (but not including) the Page Footer section. A group header section can underlay all sections up to its matching group footer section.

This enables you to produce a number of interesting report effects. For example you can:

- print an object so it appears one time in the Details section beside a number of details (for example, a graph that compares sales figures by region along side the details for the regions),
- print a company watermark that is centered on the page, flowing through multiple sections, and
- use a scanned bitmap of a form as a guide in setting up a report to print on preprinted forms.

Using the Underlay facility, you can produce the visual effects that you want without compromising your data in any way. See *How to make an object underlay a following section(s)*, Page 100.

HANDS-ON (Report Design Environment)

How to move the toolbar and format bar

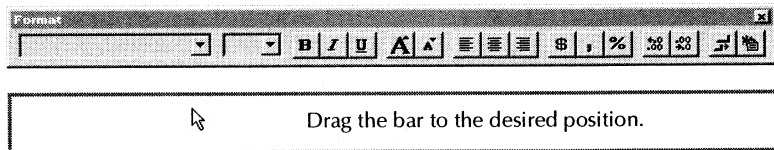
The 32-bit version of Crystal Reports includes a movable, resizable toolbar and format bar.

NOTE: You cannot move or resize the toolbar or format bar in the 16-bit version of Crystal Reports.

You can move these bars into fixed positions or you can turn them into floating palettes if you wish. You move and resize both bars in the following way:

Moving the bars

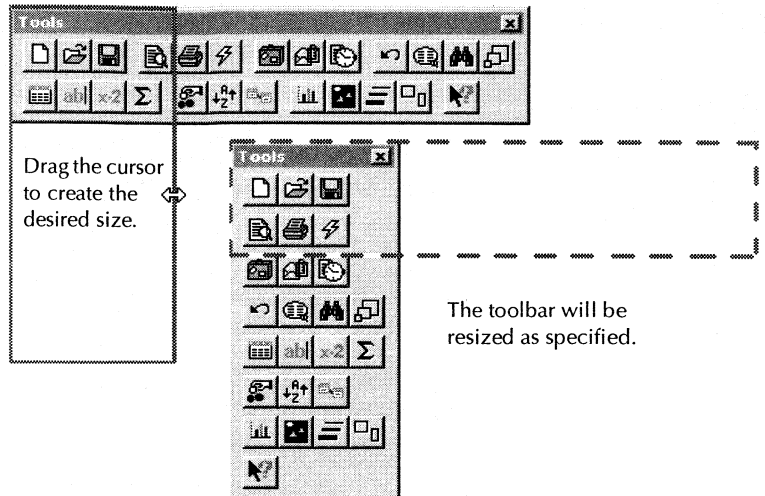
To move the bars, click on the bar where there is no button and drag the bar to a new position.



Resizing the bars

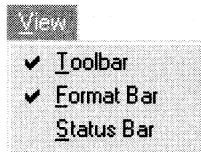
To resize the bars, position the pointer over an edge or corner of the bar. When the resizing pointer appears, drag it inward or outward to resize the bar.

To go from this...



to this . . .

You can toggle the toolbar, format bar, and status bar on and off using the respective commands on the View menu.



A check mark indicates that the bar is visible.

Related Topics

See *The Toolbar*, 51

See *The Format Bar*, 53

How to add, delete and move guidelines

Crystal Reports provides guidelines to help you accurately place objects on your report. Guidelines are non-printing lines that you can place anywhere in the Design and Preview Tabs to aid in alignment. You can toggle the visibility of the guidelines on and off in two ways:

- Using the two commands on the View menu, GUIDELINES IN DESIGN and GUIDELINES IN PREVIEW. See *Guidelines in Design command*, Page 545, and *Guidelines in Preview command*, Page 545.
- Using the two properties, *Show Guidelines in Design* and *Show Guidelines in Preview* on the Layout Tab in the File Options dialog box. Search for *File Options dialog box* in Crystal Reports online Help.

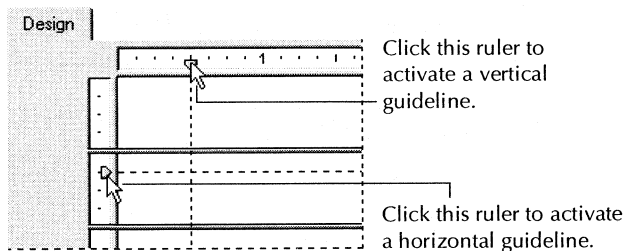
Crystal Reports inserts guidelines automatically in some situations:

- Whenever you insert a field or formula field in your report, the program automatically creates a guideline at the left edge of the field frame and snaps the field and field title to it.
- If you summarize a field, the program snaps the summary to the same guideline to assure proper alignment.
- When you right-click the shaded areas to the left of a section and choose the ARRANGE LINES command, the program automatically creates one or more horizontal guidelines in the section and snaps the fields to them.

To add guidelines

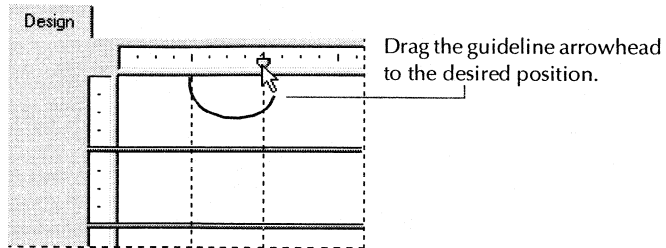
There will be times that you will want to insert guidelines manually.

- To add a vertical (up and down) guideline, click the ruler at the top of the Design Tab where you want the guideline to appear.
- To add a horizontal (side to side) guideline, click the ruler at the left of the Design Tab where you want the guideline to appear.



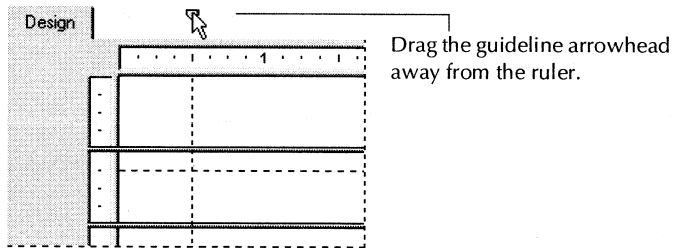
To move a guideline

Each guideline is attached to an arrowhead on its originating ruler. To move a guideline, simply drag its arrowhead to the place you want the guideline to appear.



To delete a guideline

To delete a guideline, drag its arrowhead away from the ruler.



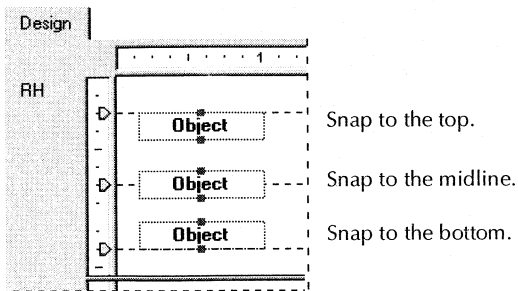
NOTE: With the *Snap to Grid* option selected you will only be able to insert or move guidelines in grid increments. To get the full power of guidelines, turn off the *Snap to Grid* property using the *Layout Tab* in the *File Options* dialog box. Search for *File Options* dialog box in *Crystal Reports online Help*.

How to move and position objects using guidelines

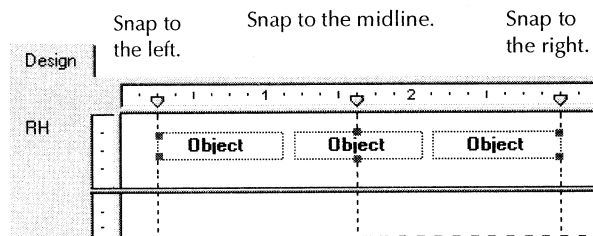
To position objects with guidelines

Guidelines have a snap property that automatically snaps objects to them. That is, objects will connect to the guidelines as positioning devices.

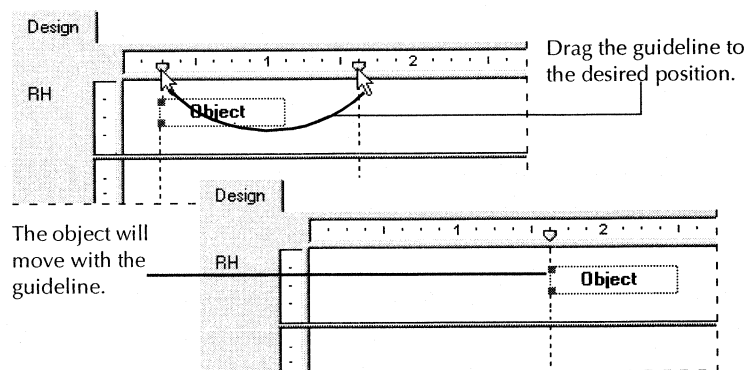
An object can snap to a horizontal guideline on its top, bottom, or horizontal midline (the invisible line that bisects the object horizontally).



An object can snap to a vertical guideline on its right or left side or vertical midline (the invisible line that bisects the object vertically).



When a guideline has its snap property toggled on and you move the guideline, you move all objects that are snapped to the guideline as well.



NOTE: When you move a guideline, you move any object that is snapped to it. But if you move an object that is snapped to a guideline, the program does not move the guideline.

To resize an object using guidelines

When you snap one or more objects to guidelines on two sides (top and bottom, or, right and left), you can resize the object(s) by dragging either or both of the guidelines. This can be especially helpful if you need to resize a number of objects similarly at the same time.

This is a two step process. First you have to snap the object(s) to two guidelines; then you have to drag a guideline to resize the object(s).

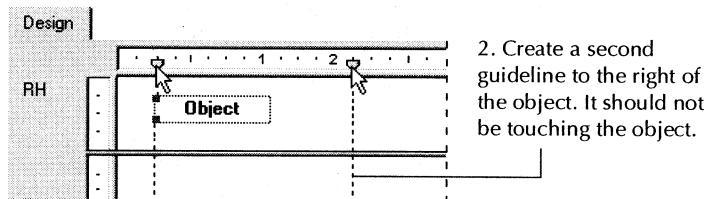
Snapping the Object(S) with Two Guidelines

There are two typical cases in which you might want to snap objects to two different guidelines.

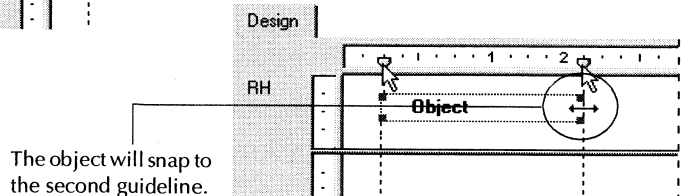
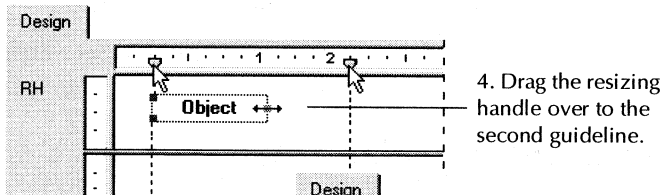
CASE 1

In the first case, you have two or more objects that you want to make the same size (height or width) and they are not the same size now.

1. Begin by creating a guideline and snapping one side of the object to that guideline.



3. Now click the object to activate the sizing handles.

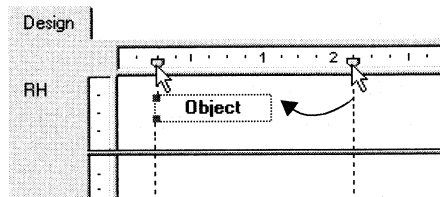


- Repeat Steps 2 and 4 for every additional object you want to snap to both guidelines.
- If the objects are not yet the size you want them, drag either or both of the guidelines until the objects are the correct size.

CASE 2

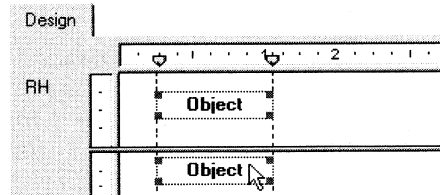
In this case, you have two or more objects that you want to resize and they are all the same size now.

- Create a guideline on one side of the first object and snap the object to it.



- Create a second guideline to the right of the object and drag the guideline so it touches the object.

- Click inside the object; the frame changes color when the snapping is complete. The guidelines are now the correct distance apart to hold all the remaining objects.



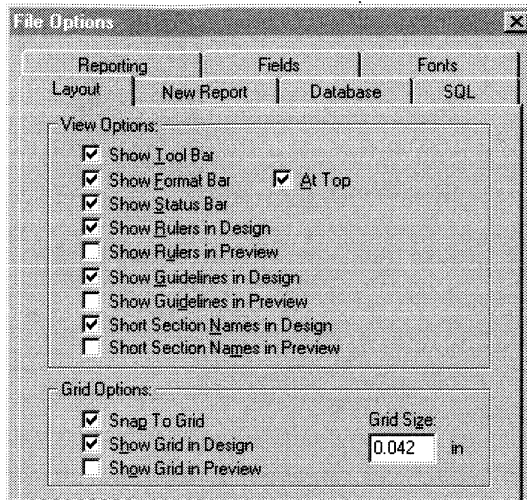
- Drag each of the other objects into position between the guidelines to snap them.

- Drag either of the guidelines to resize all of the objects proportionately.

How to turn the grid on/off

The Design and Preview Tabs have an underlying grid structure that you can activate and resize in the File Options dialog box.

- Choose the **OPTIONS** command from the File menu. The File Options dialog box appears with the Layout Tab active.



2. Set the Grid Size, Snap to, and Show options in this dialog box.

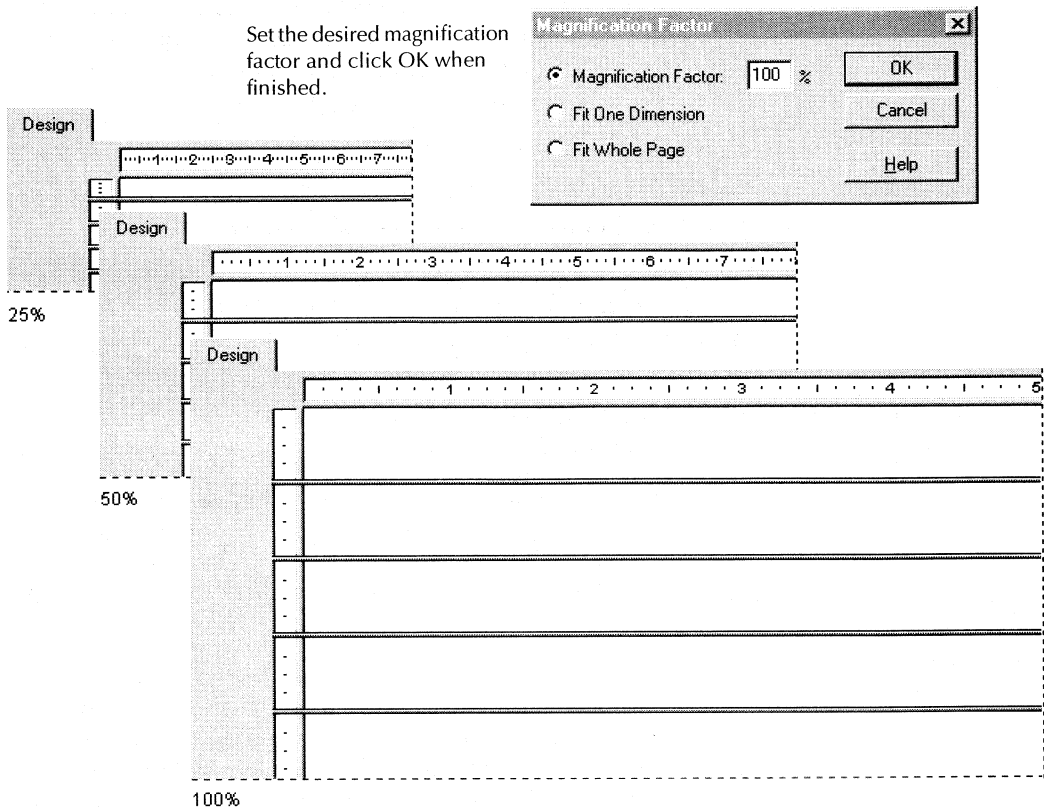
NOTE: By default, all of the grid options are toggled off.

How to zoom your report in and out

Crystal Reports enables you to zoom in on your report when you want to see details more clearly and to zoom out when you are more interested in overall layout. You can choose any magnification from 25% to 400%. This feature is active in both the Design and Preview Tabs.



To zoom your report in or out, click the ZOOM button. Set the zoom level you want in the Magnification Factor dialog box when it appears.



It is helpful to view reports at low magnifications in order to get an overall picture of the layout of your report. Views at higher magnifications work well for attention to the details of the report.

How to undo/redo activities

Crystal Reports includes multiple levels of undo. With multiple levels of undo, you can undo several changes, a step at a time, until you have your report in the condition you want it.

The program also has a redo feature that reverses an undo. If you move an object, for example, and do not like its new position, you can click UNDO to move it back to its original position. If you then change your mind, you can click REDO to restore it to the place where you moved it.



- To undo an action, click the UNDO button. The first time you click the button, it reverses the most recent change you made to your report. Each additional time you click the button it reverses the next most recent change.



- To redo a change after you have undone it, click the REDO button.

NOTE: If you undo an action (Action A) and then perform some new action, you will no longer be able to Redo action A.

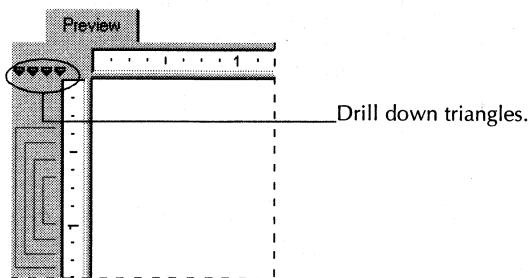
NOTE: The program disables the UNDO button and the UNDO/REDO commands on the Edit menu whenever there is nothing to undo/redo or you have made a change that can not be reversed.

How to drill down on summarized data

You can drill down on the Preview Tab using the drill down triangles (to show summary data for all the groups in a section) and/or the drill down cursor (to show the data behind individual groups). See Chapter 9, *Sorting, Grouping, and Totaling*, Page 201.

Drill down triangles

When you have group summary sections in your report (subtotals, counts, and so forth), the program displays drill down triangles that you can use to drill down through the summary data, one group area at a time. Each triangle is a toggle that you click to show (triangle pointed down) or hide (triangle pointed to the right) the section it represents.



The program creates one drill down triangle for each summary section and one for the Details section. For example, if your report has data that is subtotaled by Country, then by Region, and finally by City, the program displays four drill down triangles, three for the group summary sections and one for the Details section.



Group 1, Group 2, Group 3, Details (left to right)

Using these triangles, you can display

- only the country subtotals:



- only the country and region subtotals:



- the country, region, and city subtotals:



- all the subtotals and the details behind them as well:



You are not drilling down on individual groups here but on group sections. You either see the summaries for all of the groups in a section or none at all. To see the data behind individual groups, use the drill down cursor.

NOTE: When you click a triangle to point it down, all the triangles to its left point down as well. Thus, while you can view the Group #1 sections by themselves, you cannot view the Group #2 sections by themselves. If you make the Group # 2 sections visible, the program will make the Group #1 sections visible too.

Drill down cursor

When you position the cursor over any summary value that you can drill down on, Crystal Reports displays a drill down cursor. If you double-click, the program reveals the details behind that specific summary value. For example, if the drill down cursor becomes active over the city summary, you can double-click to see the details behind that summary. See *Crystal Reports cursors*, Page 58.

- If you have only a single summary, you can look at the summary or at the data from the individual records that are summarized.

- If you have multiple summaries, you can look at the summaries behind summaries (the city summaries that make up the region summaries, for example), or at the data from the individual records that are summarized.

HANDS-ON (Sections and Areas)

How to add, delete, move, and merge sections





The Crystal Reports Design Tab opens originally with a standard set of report sections, each in a standard size. You can not delete any of these original sections but you can add to them. Once you have added sections, you can delete them, move them in relation to other similar sections, or merge related sections together.

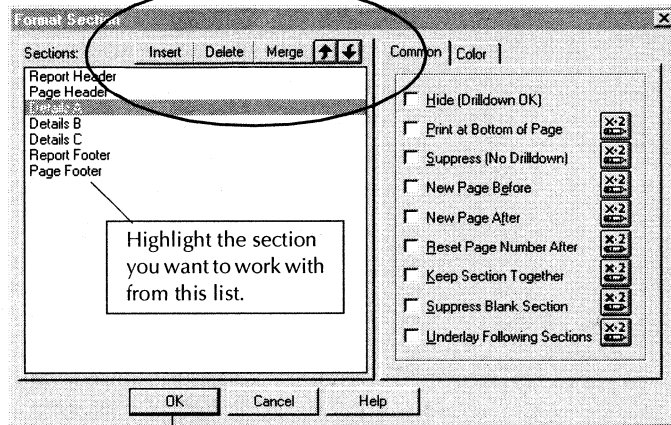
While there are different ways to do this, an easy way is using the SECTION EXPERT button on the toolbar.



Click the SECTION EXPERT button on the toolbar. The Format Section dialog box appears with a list of all the sections in the report. When there are more than one of any kind of section, the sections are lettered a, b, c, and so on.

NOTE: Crystal Reports enables only those options that apply to the highlighted section.

	Click Insert to insert a new section.		Click Merge to merge two sections.
	Click Delete to delete an existing section.		Click the arrow buttons to move a section up or down in an area.



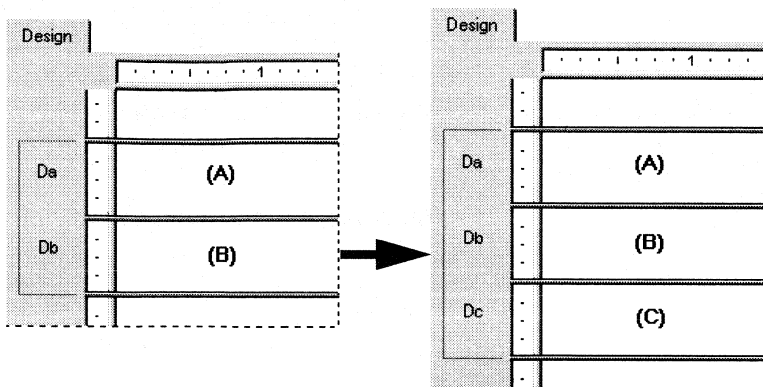
Click OK when finished.



TO INSERT A SECTION

Click the *Insert* button. The program adds a new section immediately below the highlighted section.

NOTE: You can also insert a section by right-clicking the shaded area to the left of any section and choosing the INSERT SECTION BELOW command from the shortcut menu that appears. For example, if you right-click to the left of a Details section and choose the INSERT SECTION BELOW command, the program inserts another Details section below that section.



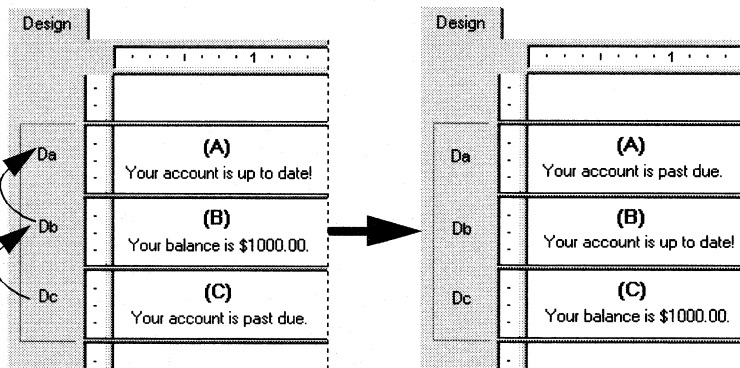
TO DELETE A SECTION

Click the *Delete* button. The program removes the highlighted section from your report.



TO MOVE A SECTION

1. Highlight the section you want to move.
2. Click the *Up* or *Down* arrow to move the section.



With Section (C) highlighted, click the up arrow twice.

The data originally in Section (C) is moved to Section (A). The data in the other sections is moved down.

NOTE: You can only move a section up or down within an area.

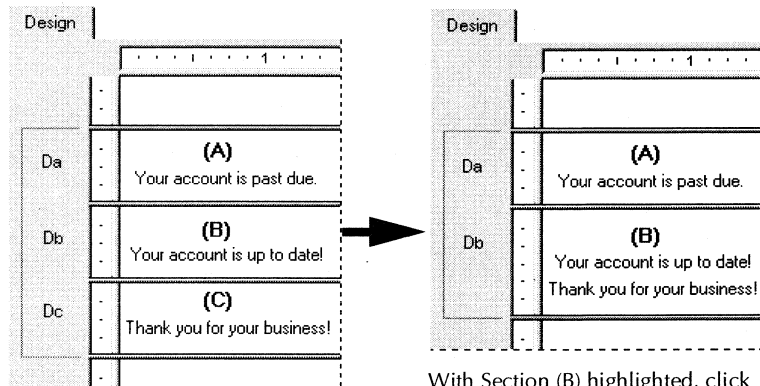
NOTE: The letters that identify the sections describe their relative (as opposed to original) position. Thus, if you move a "C" section up, it becomes a "B" section. It loses its original "C" designation.

Merge

TO MERGE TWO RELATED SECTIONS

There may be times when you have placed objects in two sections (where they print sequentially) and you want to put them all in a single section (where they print simultaneously). You can merge the two sections and then rearrange the objects as needed in the new section.

1. Move the sections so the two sections you want to merge follow each other in the list.
2. Highlight the top section.
3. Click the *Merge* button. The program combines both sections into a single section.
4. Rearrange the objects as needed.



With Section (B) highlighted, click the Merge button and Section (C) will be merged with Section (B) to form one section.

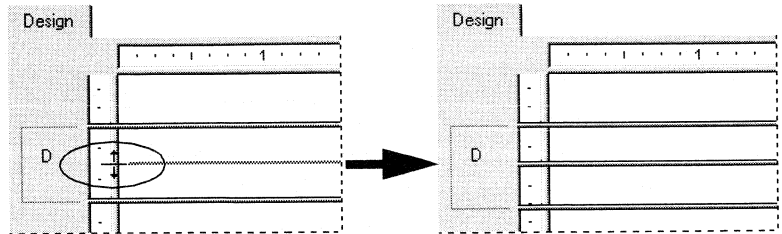
How to split and resize sections

You can split a section into two or more sections or resize sections using the mouse on the Design Tab.

TO SPLIT A SECTION

1. Move the pointer over the left boundary of the section you want to split.

2. When the pointer becomes the section splitting cursor, click the boundary. See *Crystal Reports cursors*, Page 58.
3. When a horizontal line appears, drag it up or down to split the section the way you want it.



When the section splitting cursor appears, drag the horizontal line to the desired position.

The section will split at the line.

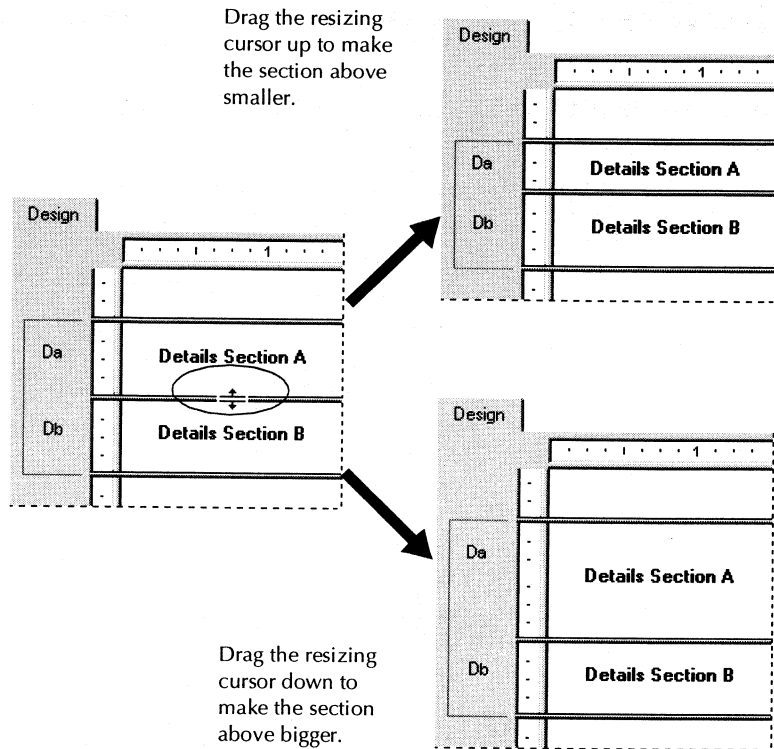
NOTE: When you split a section you are creating two sections out of one. Each is smaller than the original section. You can accomplish the same thing by adding a section and then resizing both sections.

TO RESIZE A SECTION

Move the pointer over the bottom boundary of the section you want to resize and, when the pointer becomes a resizing cursor, drag the boundary to make the section bigger or smaller as you wish.

NOTE: If you have one or more objects in a section and you want to resize the section to remove unnecessary white space, right-click in the shaded area to the left of the section and choose the **FIT SECTION** command from the shortcut menu that appears. The program automatically resizes the section, moving the bottom boundary of the section to the baseline of the bottom object in the section.

Drag the resizing cursor up or down to change the size of the section.



NOTE: A section will automatically expand vertically in two instances:

- When you place an object, the program puts it into the section where the top margin of the object is and expands vertically if necessary to ensure the entire object fits into the section.
- A section will also expand if you resize an object vertically to ensure the object fits.

NOTE: You can not resize a section smaller than an object in the section.

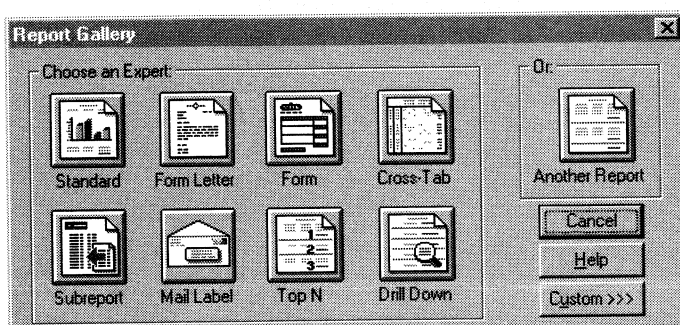
NOTE: When you have an object in a section, the program will not let you drag the bottom boundary of the section up above the baseline of the bottom object, regardless of where the object is positioned in the section.

HANDS-ON (Report Creation and Design)

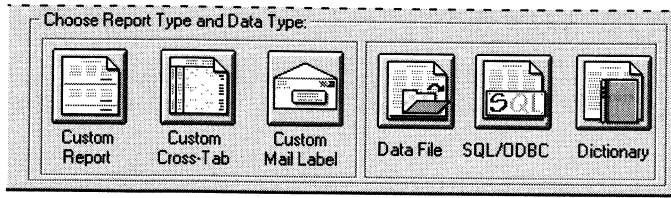
How to select data and begin creating a report



You can use Crystal Reports Experts to help you create reports as quickly as possible. When you click the **NEW REPORT** button, the Report Gallery appears. In the Report Gallery you will find a series of buttons representing the different types of experts that are at your disposal.



- Click the icon for the type of report you want to create and follow the steps outlined on the tabs in the Expert that appears.
- If you want to build a new report based upon one that already exists, click the *Another Report* button. You can select a report file to serve as a template for your new report. The program will make a duplicate of that original report, which you can modify however you please. This is a great way to create a report that is similar to an existing report; when you need to create a series of reports, each a little different than the last; when you need to reconstruct a report based on an earlier time period using the same report structure you use today, or when you need to create an entirely new report based on a set of databases that are linked in another report.
- If you want to build a report from scratch, click the *Custom* button in the Report Gallery. Several Report Type and Data Type icons will appear at the bottom of the Report Gallery. Click the buttons that are appropriate to your needs.

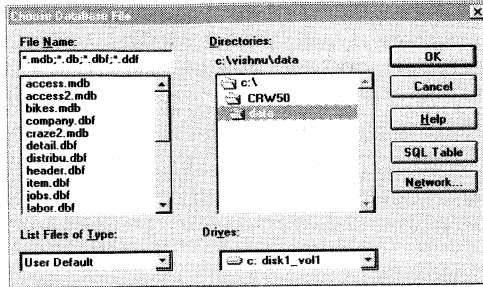


The program will take you to a dialog box that enables you to select the data you need. Since building reports from scratch is the easiest way to learn about all the powerful features of Crystal Reports, this is the method we will discuss in the manual. For information on creating reports from another reports, see Page 157. Search for *Sample Reports* in Crystal Reports online Help.

If you choose
Data File as
your data
source . . .

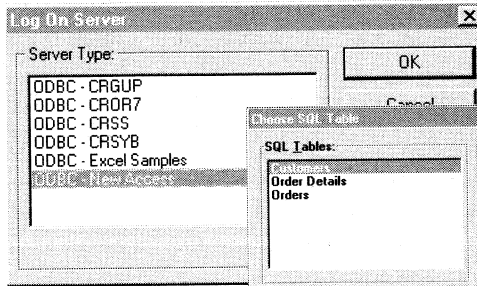
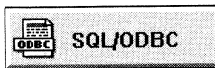


1. Using the File Name and Directories list boxes, select the desired file.

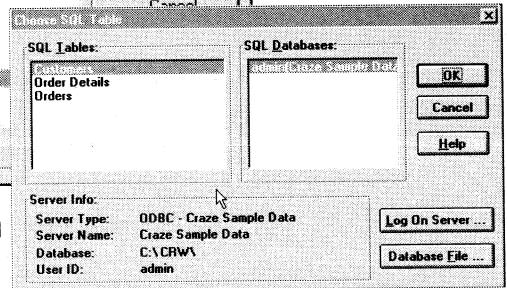


2. Click OK to open the file and add it to your report.

If you choose
SQL/ODBC
as your data
source . . .

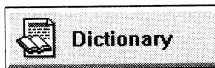


1. First, choose the Server Type.

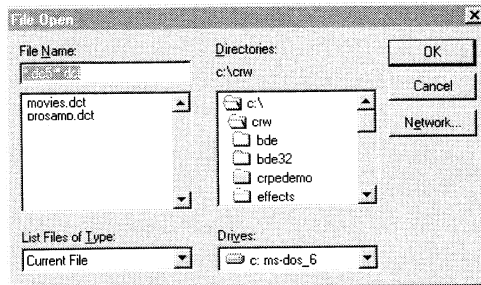


2. Then, using the SQL Tables and SQL Databases list boxes, select the desired table and click OK.

If you choose
Dictionary as
your data
source . . .



1. Using the File Name and Directories list boxes, select the desired file.



2. Click OK to open the file and add it to your report.

See Chapter 20, *Data Sources*, Page 475.

See *How to Select a Crystal Dictionary for a Report*, Page 401.

How to add and link multiple tables

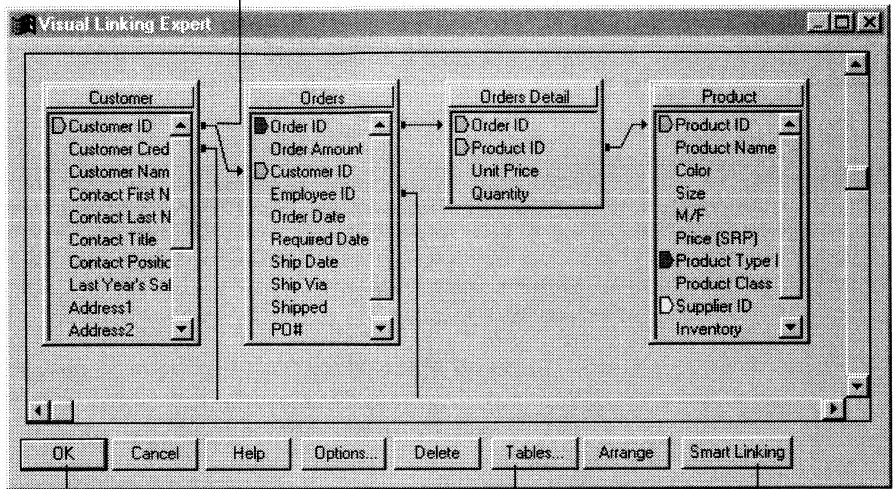
After you have selected a database table to begin working on your report, you may need to select a second table and then link the tables so the records in each table match up. For example, if you are using a customers table and an orders table for your report, you will need to link the tables so the records in the orders table are matched up with the records of the customers who placed the orders. Here is the process to follow.

Choose the ADD DATABASE TO REPORT command from the Database menu. The Choose Database File dialog box appears. Using the *Directories*, *Drives*, and *File Name* boxes, select the new database that you want to use in the report and click OK when finished. The Visual Linking Expert appears and displays the databases currently available for linking. Search for *Visual Linking Expert* in Crystal Reports online Help.

NOTE: *If the Auto-Smart Linking option is turned on in Database Tab of the File Options dialog box, you will not have to manually create links between the tables. See Linking Tables, Page 411.*

NOTE: *If your primary database is either Access or Btrieve, all tables contained in those databases will appear in the Visual Linking Expert. You do not have to manually add each table to your report. See Access, Page 504.*

To create links manually, drag a field from one table to a field in another table. If successful, a link line will be created. If unsuccessful, a message is issued.



Click OK when finished.

To specify the tables, click the Tables button.

To link databases automatically, click the Smart Linking button.

NOTE: When manually creating links, the "to" field must be an indexed field.

The Visual Linking Expert dialog box closes, and you are returned to your report. The linked databases are now available to use. If you are not satisfied with the link, you can modify it using the Visual Linking Expert. Search for *Visual Linking* in Crystal Reports online Help.

Related Topics

See *How to add and link multiple tables*, Page 92.

See *SQL join types*, Page 426.

How to insert database fields

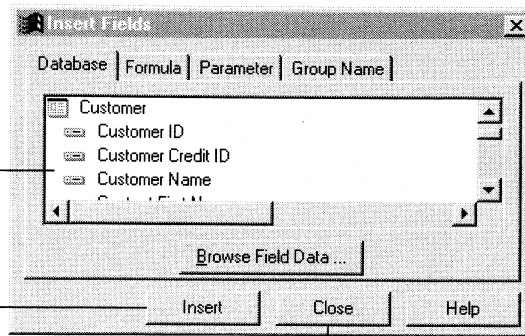


Click the INSERT FIELDS button on the toolbar. The Insert Fields dialog box appears with the Database Field Tab active listing all of the fields in the chosen database(s).

You can also double-click a field to place it or drag and drop the field directly onto your report.

Click the field you want to appear in your report.

Click the Insert button to place it in your report.



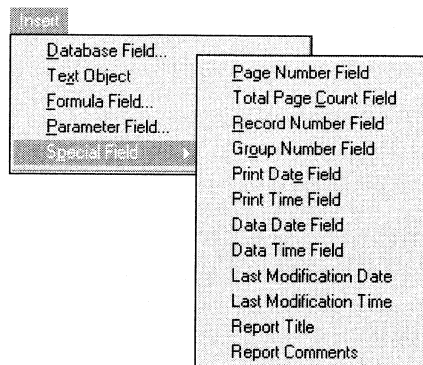
To speed the report building process, this dialog box is set to remain on screen until you click the Close button. You can move the dialog box wherever you wish.

NOTE: If you are working with Btrieve or Access, the Insert Fields dialog may originally list only the table names. If this is the case, just double-click on the table name to get a listing of the fields in that table.

How to insert special fields

Crystal Reports allows you to insert Page Number fields, Record Number fields, Group Number fields, Print Date fields, and Total Page Count fields into your report easily.

To insert special fields, choose the SPECIAL FIELD command from the Insert menu. A submenu appears with the special fields available for use in your report. See the *Special Field commands*, Page 547-549.



Choose the desired special field from this menu.

Each special field is inserted into your report as an object. Once you choose the field you want from the menu, a frame appears and you place it into the desired position on your report.

NOTE: *If you want to change the formatting of any of the inserted objects, right-click the object and choose the **CHANGE FORMAT** command from the shortcut menu. The **Format Editor** appears where you can make the desired changes. See *Types of Formatting Properties*, Page 186.*

How to insert a Page n of N field

You can use special fields and text object to create a Page *n* of *N* field where *n* is the current page number and *N* is the total number of pages in the report.

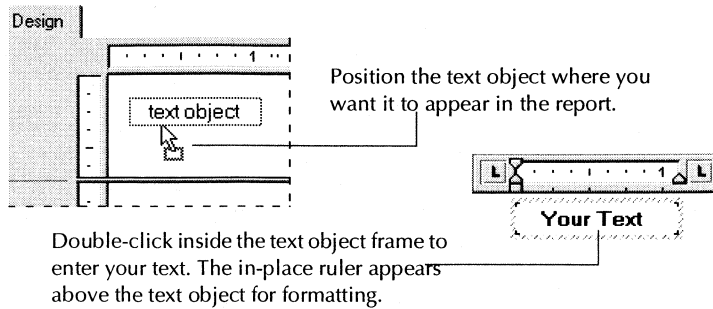
1. Insert a text object where you want the field to appear in the Page Header or Page Footer section of your report. See *How to insert text objects*, Page 95, and *How to work with text objects*, Page 168.
2. Type in the word "Page" followed by a space.
3. Insert a Page Number field in the text object using the PAGE NUMBER command on the Insert | Special Field menu.
4. Type in a space, the word "of" and another space.
5. Insert a Total Page Count field using the TOTAL PAGE COUNT command on the Insert | Special Field menu.

Now, when you print the report the program will print "Page *n* of *N*" for each page of your report.





How to insert text objects



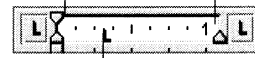
Click the INSERT TEXT OBJECT button on the toolbar. An empty object frame appears. Click once on the border of the text object to select it for resizing and moving it. Double-click inside a text object to select it for editing.



The in-place ruler allows you to add tabs, indent, and align text within the text object. By clicking the tab box on either end of the ruler, you can cycle through the four tab options available. Word wrap is active within the text object. As you type, the text will automatically wrap when it reaches the right edge of the text object. The object frame will expand vertically as you type to accommodate the text.

-  Left-aligned tab.
-  Right-aligned tab.
-  Center-aligned tab.
-  Decimal-aligned tab.

Drag to set the left margin. Drag to set the right margin.



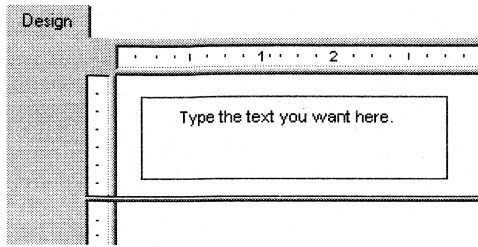
Once you have chosen the desired tab, click at the position on the ruler where you want to insert it.

NOTE: If you format text within a text object to a larger font size, the object frame will expand vertically to accommodate the text, but will not expand horizontally.

How to use a database field in a text object

The primary function of a text object is to hold text. But text objects can hold database fields as well, making them ideal for creating custom form letters. To insert a database field in a text object

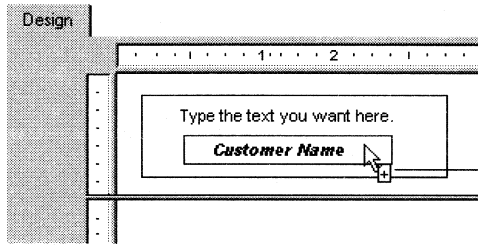
1. Insert a text object in your report.



2. Type in the text and spaces that you want to appear before the first database field (if any).



3. Click the INSERT FIELDS button on the toolbar. The Insert Fields dialog box appears with the Database Tab active.
4. Highlight the database field you want to insert and drag it into the text object. As you move the drop cursor over the text object, the program displays a movable insertion point.



5. Drag the object so the insertion point is located where you want the database field to appear and drop the field.

When you run the report, the program will print the database field values where you placed the field(s). See *How to work with text objects*, Page 168.

How to insert a picture

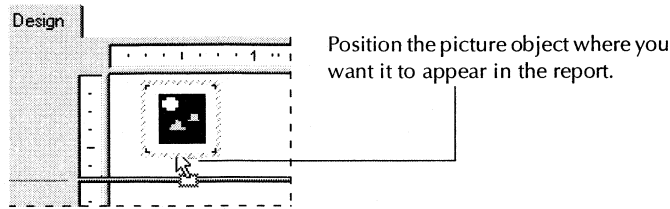
Often when you are developing reports you would like to include a picture. For example, you may wish to put a company logo into the heading of a report.

TO INSERT A PICTURE



Click the INSERT PICTURE button on the toolbar. The Open dialog box appears.

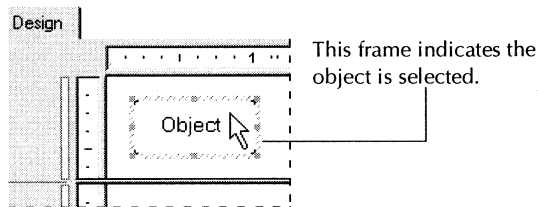
Select the desired picture file from the file list and click *Open* to return to your report. An object frame appears with your picture inside it ready to be positioned.



How to select, move and resize objects

How to select objects

To do change the font of an object, move an object, etc., you first have to select it by clicking it. When you select an object, sizing handles appears around it.



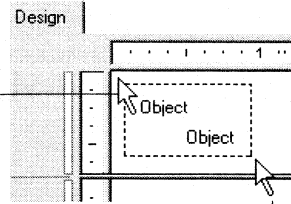
Once an object is selected, you then tell Crystal Reports what you want to do with it.

- For example, to change font size, you first select the element for which you want to change the font size.
- Then you choose the desired command from the Format menu (or the FORMAT FIELD command from the shortcut menu) to select the new font size.
- The new font size applies only to the field you selected.

How to select multiple objects

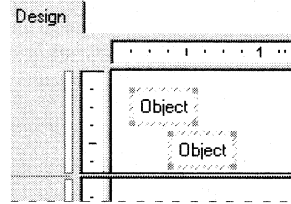
Crystal Reports allows you to select multiple objects using the marquee selection technique.

1. Position the mouse pointer above and to the left of the objects you want to select.



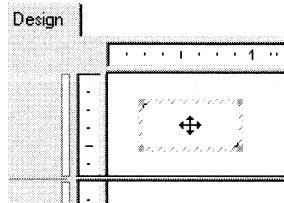
2. Drag the selection rectangle to the bottom right of the objects, surrounding them completely.

3. The rectangle will disappear and object frames appear around all of the selected objects.



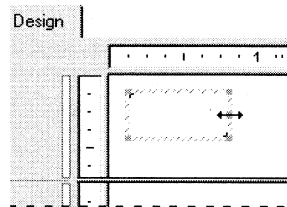
Once the objects are selected, you can move them as a group.

To move an object . . .



Press and hold the mouse button to activate the move cursor and drag the objects to the desired position.

To resize an object . . .



Drag the handles of the object frame to change the size of the objects.

NOTE: *Crystal Reports allows you to move fields across other fields without affecting the placement of the bottom fields.*

NOTE: *You can move fields between sections with the following exceptions:*

- *grand totals can only be moved within the Report Footer section or to the Report Header section, and*
- *a subtotal or summary can be moved only within its originating section or to the header portion of its originating section.*

Related Topics

See *Formatting Concepts*, Page 184.

How to make an object underlay a following section(s)

To make an object underlay a following section(s), you place it in the section above the section(s) you want it to underlay. Then you toggle the *Underlay Following Sections* property on for the section that you placed it in.

In this example you will make the CRAZE logo (CRAZEC.BMP) underlay multiple sections. This is a procedure similar to what you would follow if you wanted to insert a company watermark to serve as a background for your reports.

CREATING A SIMPLE REPORT

1. To get started, you need to create a simple report using the Customer table in CRAZE.MDB.
2. Place {customer.CUSTOMER NAME} and {customer.LAST YEAR'S SALES} side by side in the Details section of your report. To eliminate unnecessary objects in this example, delete the field titles that the program places in the Page Header section for each of these fields.
3. Break your data into region groups. To do this, choose the GROUP command from the Insert menu and choose {customer.REGION} as the sort and group by field.

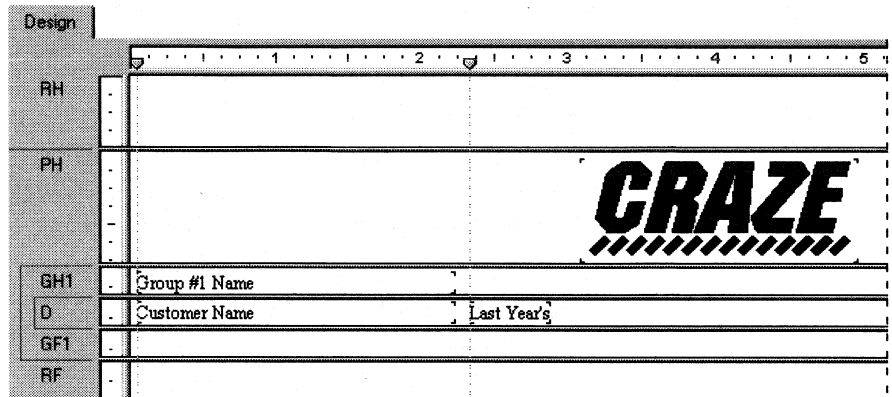
The image shows a report design grid with a 'Design' tab selected. The grid is divided into sections: RH, PH, GH1, D, GF1, and RF. The D section is expanded to show three fields: 'Group #1 Name', 'Customer Name', and 'Last Year's'. The grid has a header row with columns 1, 2, and 3. The D section is positioned in column 1, spanning rows 1 and 2.

Section	Field
RH	
PH	
GH1	Group #1 Name
D	Customer Name
D	Last Year's
GF1	
RF	

INSERTING A PICTURE



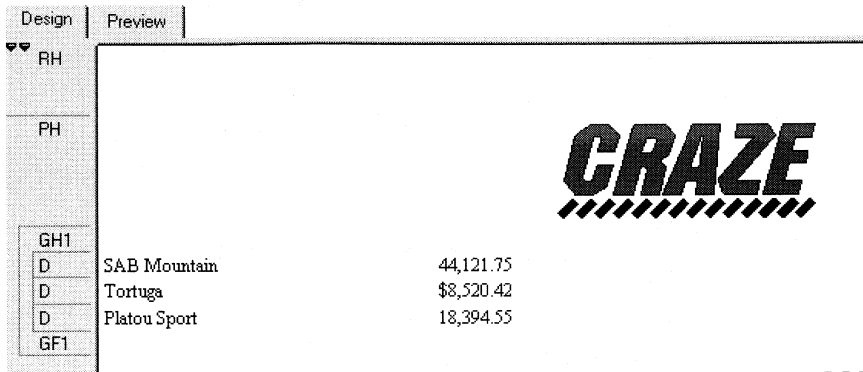
1. Click the PICTURE button on the toolbar.
2. Choose the picture file CRAZEC.BMP and place it in the Page Header section, to the right of the body of your report.



NOTE: You are placing it here because it is a bold bitmap and if it underlays the text in the report the text will be difficult to read. If you are working with a watermark which is a subdued bitmap designed to be nearly invisible, you want it to underlay the text so you place it directly above the text.

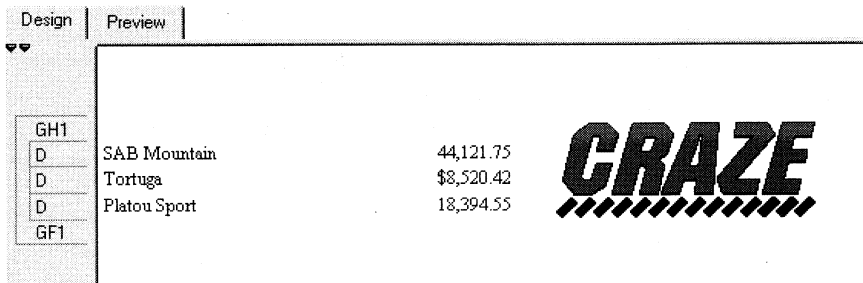


3. Click the PRINT PREVIEW button to see your report at this point. Note that the bitmap prints first (entirely in the Page Header section) and then the body of the report follows. When you are done previewing, return to the Design Tab.



UNDERLAYING THE FOLLOWING SECTION(S)

1. Click the SECTION EXPERT button and when the Section Expert appears, highlight the Page Header section and toggle the *Underlay Following Sections* property on. See *Types Of Formatting Properties*, Page 186, and *To set section properties*, Page 105.
2. Preview your work again. Note that now the bitmap prints in the Group Header and Details sections, beside the text in the body of the report.



NOTE: *Placing the bitmap to the right of the body of the report, as you did here is the same technique you would follow if you wanted a graph or an employee picture to print beside the details that pertain to the graph or to the employee.*

3. Return to the Design Tab and resize the object vertically so it is two or three times bigger and preview the report again. See how the bitmap now underlays more sections.

Design		Preview																																			
<table border="1"> <tr><td colspan="2">GH1</td></tr> <tr><td>D</td><td>SAB Mountain</td></tr> <tr><td>D</td><td>Tortuga</td></tr> <tr><td>D</td><td>Platou Sport</td></tr> <tr><td colspan="2">GF1</td></tr> <tr><td colspan="2"> </td></tr> <tr><td colspan="2">GH1</td></tr> <tr><td>D</td><td>Piccolo</td></tr> <tr><td>D</td><td>Paris Mountain Sports</td></tr> <tr><td>D</td><td>Océano Atlántico Ltda.</td></tr> <tr><td>D</td><td>Magazzini</td></tr> <tr><td>GF1</td><td>Furia</td></tr> </table>				GH1		D	SAB Mountain	D	Tortuga	D	Platou Sport	GF1				GH1		D	Piccolo	D	Paris Mountain Sports	D	Océano Atlántico Ltda.	D	Magazzini	GF1	Furia	<table border="1"> <tr><td>44,121.75</td></tr> <tr><td>\$8,520.42</td></tr> <tr><td>18,394.55</td></tr> <tr><td> </td></tr> <tr><td>38,942.34</td></tr> <tr><td>42,508.90</td></tr> <tr><td>33,382.12</td></tr> <tr><td>\$0.00</td></tr> </table>	44,121.75	\$8,520.42	18,394.55		38,942.34	42,508.90	33,382.12	\$0.00	
GH1																																					
D	SAB Mountain																																				
D	Tortuga																																				
D	Platou Sport																																				
GF1																																					
GH1																																					
D	Piccolo																																				
D	Paris Mountain Sports																																				
D	Océano Atlántico Ltda.																																				
D	Magazzini																																				
GF1	Furia																																				
44,121.75																																					
\$8,520.42																																					
18,394.55																																					
38,942.34																																					
42,508.90																																					
33,382.12																																					
\$0.00																																					

The area the bitmap underlays is determined by:

- its size,
- the section you place it in originally, and
- the position in that section in which you place it.

By modifying object size and placement, you can create many stunning effects using the underlay capabilities.

NOTE: If you print to preprinted forms, you may be able to:

- scan a form,
- place it in your report as a bitmap,
- line up the bitmap and the report using the underlay capability and the ability to move objects anywhere you want them, and then
- print your report and the form as a single unit, thus eliminating the need to print the forms separately.

How to hide parts of the report

There are a number of properties you can set to keep particular parts of your report from printing.

To hide sections

Crystal Reports has three properties you can set in the Format Section dialog box to hide report sections.

- **Hide (Drill down OK)**

The *Hide* property hides the section whenever you run the report. You can use the *Hide* property, for example, in a summary report where you want to display only the summaries but not the details behind the summaries. When you apply the *Hide* property to a section, the section can become visible if you drill down to see the section contents. You can only apply this property absolutely; you cannot apply it conditionally with a formula.

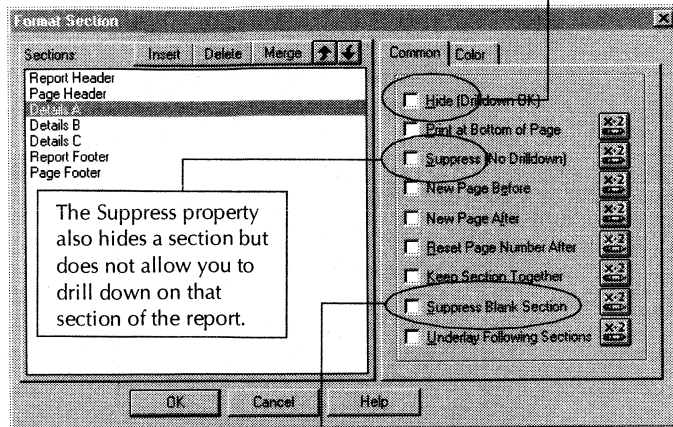
- **Suppress (No drill down)**

The *Suppress* property also hides a section when you run the report. Unlike the *Hide* property, however, you cannot apply the *Suppress* property and then drill down and reveal the section contents. You can apply this property absolutely, and you can apply it conditionally, as well, using a formula. A Form Letter is a great example of the *Suppress* option. You create two details sections - one to suppress if sales are over \$X and one to suppress if sales are less than \$X.

- **Suppress Blank Section**

The *Suppress Blank Section* property hides a section whenever there is nothing in it. If something is in the section, it remains visible.

The Hide property hides a section of your report when printing and allows you to drill down on that section of the report.



The Suppress Blank Section property eliminates non-essential blank sections from your report.

To set section properties



Click the SECTION EXPERT button on the toolbar. The Format Section dialog box appears.

- In the sections list, highlight the section you want to hide.
 - To hide the section while keeping it available for drill down, toggle the *Hide* property on.
 - To hide the section while making it unavailable for drill down, toggle the *Suppress* property on.
 - To hide the section only if there is nothing in it, toggle the *Suppress Blank Section* property on.

Now, when you run the report, the program will hide the section(s) based on your specifications.



NOTE: You can click the *Formula* button for the *Suppress* and *Suppress Blank Section* properties and create a formula that will make the setting conditional on some event. See *Conditional Formatting, Page 188*.

To hide objects

Crystal Reports has three formatting properties in the Format Editor that you can use to hide individual objects.

SUPPRESS IF DUPLICATED (Common Tab) prevents a field value from printing if it is identical to (a duplicate of) the value that comes immediately before it. The value does not print but the space in which it would have printed remains.

Cust ID	Orde
1	100.00
1	157.00
1	0.00
1	10.00
5	146.00
5	0.00
7	153.00
7	0.00
7	186.00

→

Cust ID	Orde
1	100.00
	157.00
	0.00
	10.00
5	146.00
	0.00
7	153.00
	0.00
	186.00

Duplicated values are suppressed and do not print.

SUPPRESS IF ZERO (Number Tab, Number/Currency Format dialog boxes only) prevents a value from printing if it is a zero value. The value does not print but the space in which it would have printed remains. To remove the blank space, toggle the *Suppress Blank Section* property on in the Section Expert.

NOTE: This will only work if there are no other objects in the section.

Cust ID	Orde
1	100.00
	157.00
	0.00
	10.00
5	146.00
	0.00
7	153.00
	0.00
	186.00

→


Cust ID	Orde
1	100.00
	157.00
	10.00
5	146.00
7	153.00
	186.00

The zero values are suppressed and do not print.

NOTE: To eliminate the blank lines in this situation, use the Section Expert and toggle the *Suppress Blank Section* property off for the section the field is in. This will eliminate the lines as long as there are no other objects in the section.

SUPPRESS (Common Tab) hides an object when you run the report. It is common, for example, to apply this property to formulas that are needed to do some report calculations but that you do not want to print when you run the report. When this property is checked, the object will not print.

Product ID	Unit Price	SRP
1101	4.00	6.67
1102	8.00	13.33
1103	13.00	21.67
1104	2.00	3.33
1105	11.00	18.33
1106	16.00	26.67
1107	7.00	11.67
1108	4.00	6.67
1109	12.00	20.00



Product ID	SRP
1101	6.67
1102	13.33
1103	21.67
1104	3.33
1105	18.33
1106	26.67
1107	11.67
1108	6.67
1109	20.00

The object is invisible and will not print.

To set these properties, right-click the object, choose the **FORMAT FIELD** command from the shortcut menu and set the property in the Format Editor when it appears.

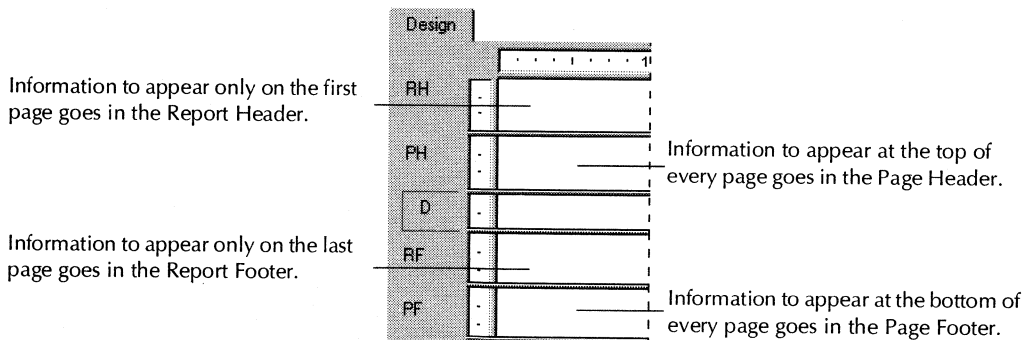


NOTE: You can click the **Formula** button for any of these properties and create a formula that will make the setting conditional on some event. See *Conditional Formatting, Page 188*.

HANDS-ON (Finishing Your Report)

How to insert page headers and footers

Insert page headers and footers by placing the information in the Page Header or Page Footer sections of the Design Tab.



- Any information you place in the Page Header and Page Footer sections prints on each page.

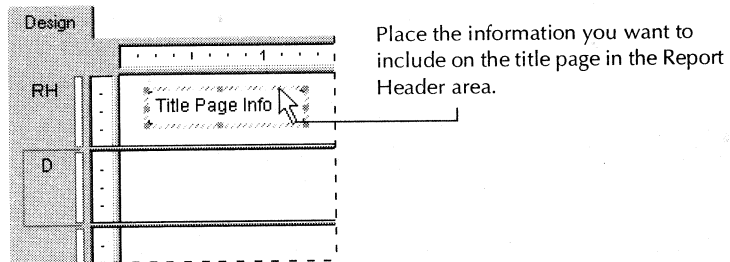
- You can use text, fields, or formulas in these sections just as you can in the Details section.

Related Topics

See *How to create a footer that appears on all pages but the first*, Page 198.

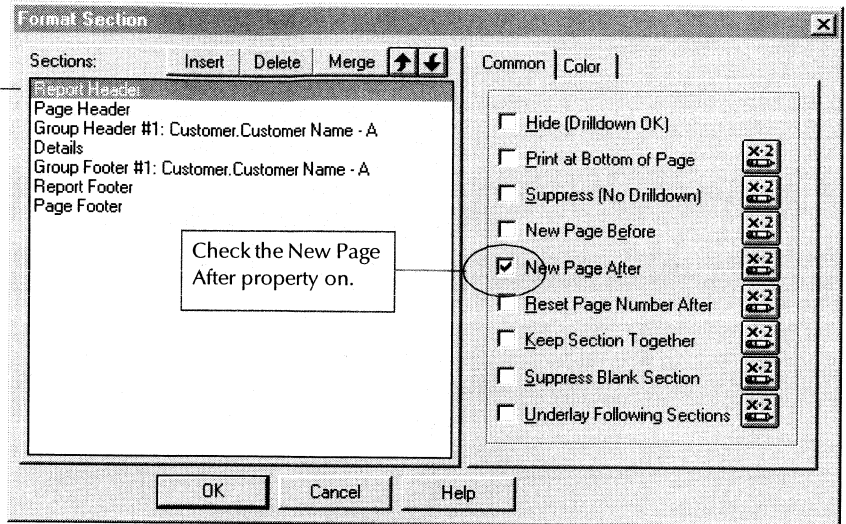
How to add a title page to your report

Adding a title page to your report is easy. Put all the information you want to include as the title page in the Report Header area of your report.



Click the SECTION EXPERT button. Using the Format Section dialog box, turn the *New Page After* property on to begin the report on the following page. See *How to set a report title using parameter fields*, Page 313.

With the Report Header section highlighted . . .



Related Topics

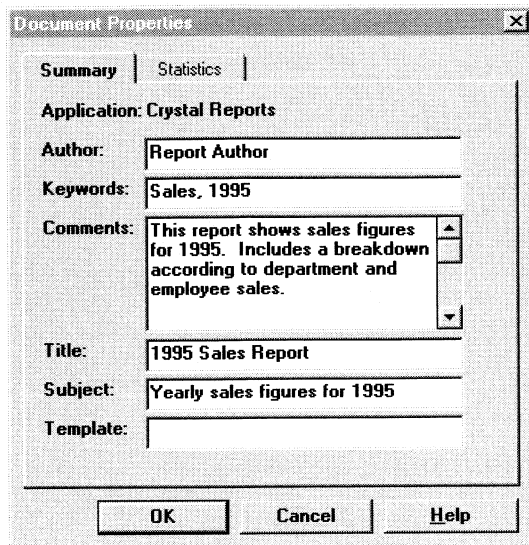
See *How to set a report title using a parameter field*, Page 313.

How to add summary information to your report

There may be times when you want to include non-printing comments with your report (a personal note to the report recipient, a note to explain more thoroughly the data on which the report is based, a report title, a comment about some particular data on the report, etc.).

The SUMMARY INFO command on the File menu provides a facility for including anything from a short note to hundreds of lines of text with your report. The comments do not print with the report; they remain in the Summary Tab of the Document Properties dialog box where they can be reviewed on demand.

When you choose the SUMMARY INFO command from the File menu, the Document Properties dialog box appears with the Summary Tab active.



Enter the desired information and click OK when finished to return to your report.

Enter whatever information you want to include with your report. If you want to include a comment, enter your comment in the *Comments* edit box. Word wrap is active in the edit box. Thus, on multi-line comments, Crystal Reports automatically breaks the lines so they fit within the margins of the edit box.

Related Topics

See *Report Title command*, Page 549, *Report Comments command*, Page 549 and *Special field commands*, Page 547-549.

HANDS-ON (Distributing Your Report)

How to export reports

You can export your finished reports to a number of popular spreadsheet, and word processor formats, into HTML format, ODBC format, and into a number of common data interchange formats as well. This makes the distribution of information easier. For example, you or your colleagues may want to use report data to project trends in a spreadsheet package or enhance the presentation of data in a desktop publishing package.

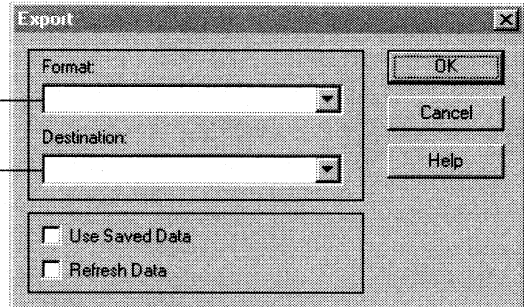
NOTE: When you export a report to a different file format other than Crystal Reports format, you may lose some or all of the formatting that appears in your report. Crystal Reports attempts to preserve as much formatting as the export format allows.



With the report active that you want to export, click the EXPORT button on the toolbar. The Export dialog box appears.

Select the format you want to export the report in from this drop down box.

Select a destination for your file from this drop down box.



The dialog box is broken into two sections, a *Format* section and a *Destination* section.

- In the *Format* drop down box, select the format in which you would like to export the report. For example, if you would like to convert the report to Microsoft Excel 4.0 format, choose Excel 4.0 (XLS) from the list.
- In the *Destination* drop down box select a destination for your file.
 - If you want to save the report to a disk file, select *Disk file*.
 - If you want to attach the report to an electronic mail document, select the appropriate Mail application. Crystal Reports supports Microsoft Mail (MAPI), Microsoft Exchange Mail (also MAPI), and Lotus cc:Mail (VIM).
 - If you want to store the report in a Microsoft Exchange folder, select *Exchange Folder* from the list.
 - If you want to send the report to a Lotus Notes database, select *Lotus Notes Database* from the list.

Click *OK*, and the export process begins.

When you export a report to a disk file, the Choose Export File dialog box appears. Use the controls in this dialog box to select a path and file name for your file.

NOTE: While the program assigns the native extension to all files you export in a specific word processor, database, or spreadsheet format, it automatically assigns the extension *.TXT for all files you export in one of the common data interchange formats. The program you want to use the data in, however, may look for specific extensions other than *.TXT. Consult the manual for that program to determine the correct file extension, and change the extension accordingly in the File Name edit box.

Click OK. The program exports the report to a disk file in the format you specify.

Exporting to e-mail

Since each e-mail system supported by the Export facility operates a little differently, the following instructions are generic in nature.

1. When you export to e-mail, the program asks you to log on to your system. Log on in the normal way.
2. Select the individual(s) to whom you are going to send the report.
3. When the message screen appears, the system displays an icon indicating that a report is attached. The icon will vary, depending on the format you used to export the report.
4. Type in any message that you want to include with the report.
5. Send the message. A Record Counter appears so you can monitor the progress of your export.
6. When the export is finished, the message will appear in the addressee's mailbox. The recipient opens the message in the normal way and double-clicks the icon to call up the report.

NOTE: Exporting to VIM is not supported by the standard 32-bit version of Crystal Reports.

NOTE: Both Microsoft Mail and Microsoft Exchange Mail use MAPI. If you intend to attach a report to an Exchange e-mail message, do not select Exchange Folder as the export destination.

NOTE: If you have questions regarding the operation of your electronic mail system, please refer to the documentation that came with the system.

Exporting to Lotus Notes

Lotus Notes is a powerful groupware application that promotes communication and information sharing between different departments in an organization. The 16-bit version of Crystal Reports allows you to export a report to a Lotus Notes database.

NOTE: You must have version 3.0 or later of the Lotus Notes Windows client. Crystal Reports will not export to a Lotus Notes OS/2 client.

To export a report to Lotus Notes:

1. In the Export dialog box, select an export file format, select Lotus Notes Database as the export destination, and click *OK*. The Select Database dialog box appears.
2. In the Select Database dialog box, select your Lotus Notes server from the *Servers* list, and select the database to which the report will be exported.
3. Verify that the file name in the *File* name text box is correct, and click *OK*. The Comments dialog box appears.
4. Type in any comments that you want to appear when another user selects your report from the Lotus Notes Desktop.
5. Click *OK*, and the report is exported.

The next time a user logs on to Lotus Notes with access to the Lotus Notes database that you selected, that user will see your report listed in the Lotus Notes Desktop. The user can double-click on the report file name to display the comments you wrote and double-click again on the report icon that appears to actually view the report.

Exporting to an Exchange Folder

Exchange is Microsoft's successor to Microsoft Mail. Exchange, however, includes not only electronic mail, but also management of group scheduling, electronic forms, groupware, and Internet connectivity. An Exchange folder can contain standard notes (mail), files, and instances of Exchange forms.

Crystal Reports lets you export a report file to an Exchange folder. You select the folder, and the report is stored there in the format that you specify. To export to an Exchange folder:

1. In the Export dialog box, select an export file format, select Exchange Folder as the export destination, and click *OK*. The Export to Exchange Folder dialog box appears.

2. In the Export to Exchange Folder dialog box, click the Report Instance option to export your report to an Exchange Folder. The other two options in this dialog box, Folder Report and Personal Report, are only available for users who have upgraded from Crystal Reports for Exchange. If you are one of those users, refer to your previous documentation for how to use this options.
3. Click *OK* in the Export to Exchange Folder dialog box. If you have more than one Exchange Profile, the Choose Profile dialog box appears. Type in the name for the Profile with the folder you want to export your report to, and click *OK*.
4. When the Select a Folder dialog box appears, select the folder in your profile where you want the report to appear, and click *OK*.
5. The report is exported to the Exchange folder you selected. The exported report can be accessed through the Microsoft Exchange client.

Exporting to HTML

Providing support for the Internet and corporate intranets becomes more important with every passing month. Crystal Reports recognizes this importance and provides World Wide Web support. Although incorporated as an export format, HTML represents a whole new export destination for your reports, as well.

By exporting your reports in HTML format, Crystal Reports provides you with a new option for rapid, convenient distribution of important company data. Once exported, your reports are accessible with any of the most popular Web Browsers including Netscape and Microsoft Internet Explorer.

To export your report to HTML format:



1. With your report open in Crystal Reports, click the EXPORT button on the toolbar. The Export dialog box appears.
2. From the *Format* drop down list box, select one of the HTML formats listed.
 - If you use Netscape as your Browser, select *HTML for Netscape 2.0* format.
 - If you use Microsoft Internet Explorer as your Browser, select *HTML for Internet Explorer 2.0* format.

- The third HTML option, *HTML Draft 3.0 Hypertext Markup Language*, is a new form of HTML that has not been officially released yet, but has been released for comments. If you are working with this new form of HTML, select this HTML format in the Export dialog box.
3. Select a destination from the *Destination* drop down box such as a disk file, an e-mail address, Microsoft Exchange, or Lotus Notes. The rest of this section assumes you select *Disk file* to store the HTML document in a directory on a Web server.

NOTE: Procedures for exporting to other destinations, such as Lotus Notes, follow closely the same procedure for exporting any other format to that destination.

4. Click OK, and the Export To Directory dialog box appears. When exported to HTML format, a report may take more than one HTML file. For this reason, Crystal Reports asks you for the name of a directory to export the report to, and uses default names for the HTML files. The initial HTML page will be saved as DEFAULT.HTM. This is the file you open if you want to view your report through your Browser application.
5. Select an existing directory, or create a new directory for the report.
6. Click OK, and Crystal Reports exports the report to HTML format.

Exporting to an ODBC data source

Crystal Reports allows you to export reports to any ODBC data source. If you have an ODBC data source set up for a database or data format, you can export your report to that data format through ODBC.

For instance, you may have an ODBC data source set up through ODBC Administrator that you normally use to access database tables designed in Microsoft SQL Server. Using the Crystal Reports Export dialog box, however, you can select your SQL Server data source and export your report as a new SQL Server database table.

NOTE: You must have an ODBC data source set up through ODBC Administrator for Crystal Reports to export to a particular ODBC database format. See *Exporting to an ODBC data source*, Page 115, and *How to set up an ODBC data source*, Page 453.

Exporting to an ODBC data source lets you:

- Change data from a centralized database into a format compatible with a local DBMS application,
- Change data from a local database format into a format compatible with a centralized database,
- Create a new database table that can be used as a separate data set in future reporting,
- Create a mini data-warehouse, and
- Massage database data by filtering records, adding formulas, and removing fields to create a new database table that provides the data you need most for your work.

To export a report to an ODBC data source:



1. With your report open in Crystal Reports, click the EXPORT button on the toolbar. The Export dialog box appears.
2. From the *Format* drop down box, select the ODBC data source for the format you want to export your report. For example, ODBC - CRSS allows you to export your report to a Microsoft SQL Server database.
3. The Destination selection is ignored when you are exporting a report to an ODBC data source. You do not need to make any changes to the Destination drop down list box. Simply click OK in the Export dialog box.
4. If your ODBC data source specifies a particular database, the report will be exported to that database. Otherwise, the Select Database dialog box appears. Select the database that this report will be added to as a new table, and click OK.
5. If the ODBC data source you selected requires a log on ID and password, the Login or SQL Server Login dialog box appears. Enter your ID and password, then click OK. The Enter ODBC Table Name dialog box appears.
6. Enter the name you want to give the new table in the database, and click OK. Crystal Reports exports the report as a new table in the database you specified.

Related Topics

See Exporting to an Exchange Folder, Page 113.

See MS Exchange, Page 490

Search for Export format and destination files in Runtime File Requirements Help.

How to fax a report

Many Fax applications, such as Microsoft Fax and Delrina WinFax, allow you to set up a printer driver that will fax documents over a modem. When using one of these applications, you can fax a report from Crystal Reports.

1. Use Print Manager to set up a printer driver for faxing documents over a modem. Consult the documentation for your fax application for instructions on how to do this.
2. From Crystal Reports, choose the **PRINTER SETUP** command from the File menu. The Print Setup dialog box appears.
3. In the dialog box, click **Specific Printer**, and select the fax driver from the drop down list box. Click **OK** when finished.
4. Choose the **PRINTER** command from the **File | Print** menu. The Print dialog box opens.
5. Click **OK** to fax the report. Your fax application will appear, allowing you to select a cover page and fill in the appropriate fax information. For instructions on how to use this software, refer to your fax application documentation.

5

Tutorial - Customer List

What you will find in this chapter...

Overview 120

Getting Started 121

Record Selection 135

Grouping and Sorting 140

Completing the Report 145

Overview

The following tutorial has been designed to give you confidence when creating your first report.

You begin by learning the basic concepts - calling up a database, placing some fields on your report, and selecting records for your report. You will also learn how to:

- start a report,
- insert and move database fields,
- add a title to your report,
- format the title in your report,
- display your report in the Preview Tab so you can fine-tune your work,
- use the Select Expert to ensure your report includes only the data you need for your report,
- move objects on your report,
- group and sort data,
- insert pictures in your report, and
- print your report.

Before you begin

- This tutorial assumes you are familiar with Microsoft Windows and uses conventional names and procedures common to the Windows environment. If you are not familiar with Windows, you may have trouble understanding basic procedures such as *scrolling* and *clicking*. Please refer to the documentation that came with Microsoft Windows for further explanation of these procedures. Also see the Preface, Page 1, for a description of the conventions used in this chapter.
- The default font for all report sections in the program is set to 10 point Times New Roman. If you have changed the default font, or if your printer does not support this font, the field size, field spacing, and screen displays will be different than displayed in this tutorial.

- This tutorial has been designed under Microsoft Windows 95. Screen displays may vary slightly if are using Windows 3.1, or NT.

If you are not familiar with the Crystal Reports environment, please review the following sections before beginning this tutorial:

- Chapter 4, *Getting to Know Crystal Reports*, Page 47, which explains the main window of Crystal Reports, the Design and Preview Tabs, the menu bar, toolbar, format bar, and status bar, as well as many other features of Crystal Reports.
- *How to add, delete, and move guidelines*, Page 73, and *How to move and position objects using guidelines*, Page 75.
- *Other fundamentals*, Page 68, which describes working with the grid, freeform placement of objects, freeform placement with guidelines, as well as a description of how to work with sections and objects.

Getting Started

In this tutorial you will get an introduction to Crystal Reports as you create a Customer List report. The Customer List is one of the most basic business reports and typically has information such as Customer Name, City, Region and Contact Name. Use customer lists for mailings, sales activities and a proprietary phone book of your customers.

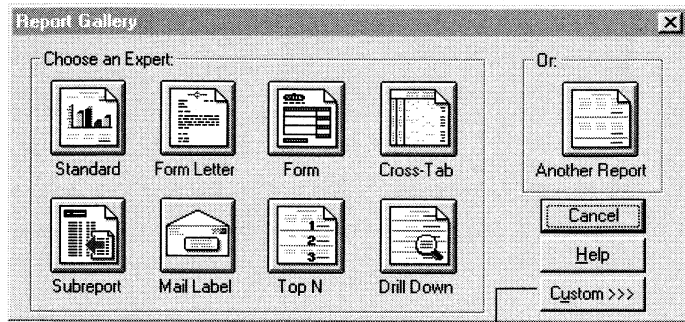
Creating Your Report



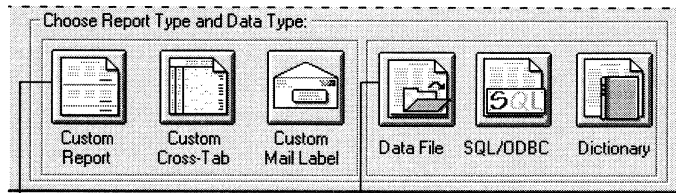
1. Click the NEW button on the toolbar. The Report Gallery appears.

The Report Gallery contains a number of buttons. These buttons can call up Experts that guide you through the creation of specific kinds of reports. Since you can be learning reporting concepts here, you can skip the Experts and build your report from scratch. After you have completed this tutorial, you may want to build some reports using the Experts to decide which method of report construction you are most comfortable with.

To build a report without an Expert:



2. Click the Custom button to expand the Report Gallery.



3. Click the Custom Report button.

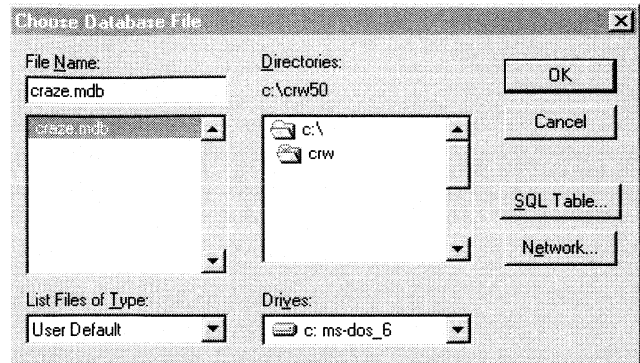
4. Click the Data File button.

The Report Gallery disappears, and the Choose Database File dialog box appears.

NOTE: You can also create reports based on SQL/ODBC data sources, Crystal Dictionary files, or Crystal Query Files.

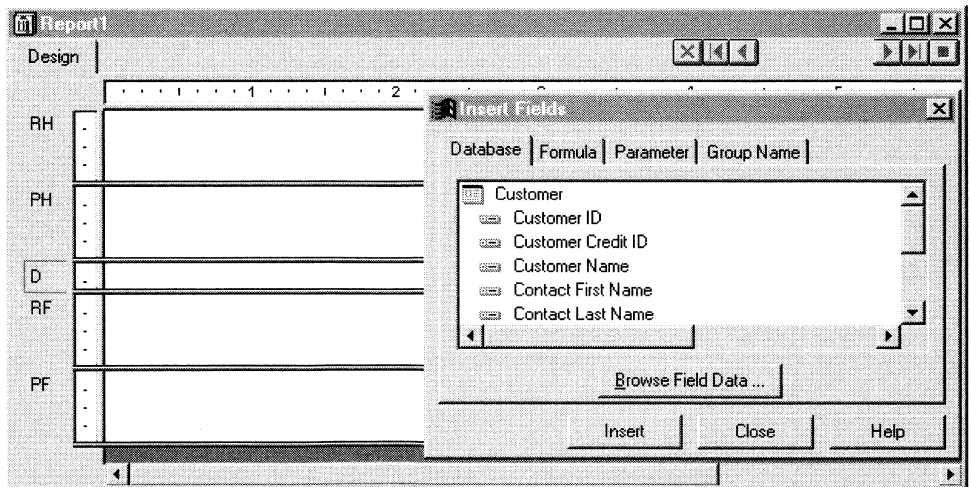
Selecting a database to use

The next step in creating a report is to select a database. Use the CRAZE.MDB sample database for this tutorial.



1. Select the file CRAZE.MDB from the *File Name* list box. This file was installed in the \CRW directory (or the directory in which Crystal Reports resides).
2. Click *OK* to open the file.

The Design Tab appears along with the Insert Fields dialog box.



NOTE: If you see a list of tables in the Database Tab of the Insert Fields dialog box, simply double-click over a table name to toggle fields in table list on and off.

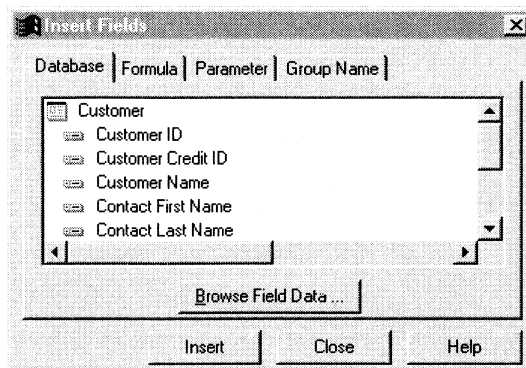
Report sections

The Design Tab is divided into five sections: *Report Header (RH)*, *Page Header (PH)*, *Details (D)*, *Report Footer (RF)*, and *Page Footer (PF)*. If, at any time, you are unsure of the report section you are working in, simply look to the shaded area to the left of the report which will always display the section names or the initials that designate the names. See *Design Tab*, Page 60.

NOTE: If you have already activated the Use Short Section Names check box in the File Options dialog box prior to reading this tutorial, the Report Header, Page Header, Details, Report Footer and Page Footer section names will appear as RH, PH, D, RF and PF respectively as shown in the diagram above. You will learn how to make changes in the File Options dialog box later. For now, realize that PH on your screen represents Page Header, and so on. Search for Setting up Crystal Reports in Crystal Reports online Help.

For the diagrams in this tutorial *Short Section Names in Design* has been selected for the sake of displaying more information.

This dialog box appears automatically with the Database Tab active because it is almost certain you will want to insert database fields when you create a new report.



NOTE: If you see a list of tables in the Database Tab of the Insert Fields dialog box, simply double-click over a table name to toggle fields in table list on and off.

This dialog box is set to remain on screen until you click the Close button. All of the tables available for use in your report are listed in this box.

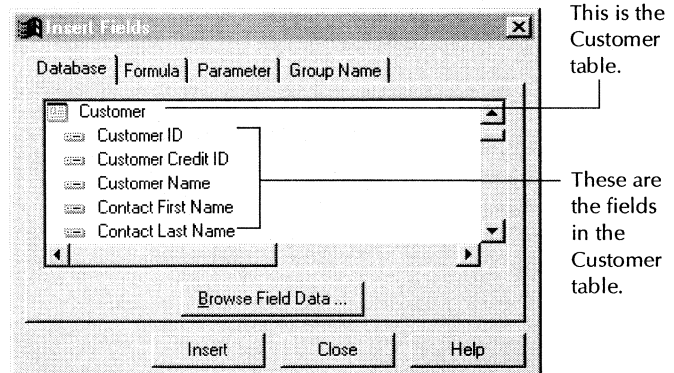
You can drag and drop the dialog box to another location on the screen by clicking on its title bar and dragging it to a new location. You can also resize the dialog box if you wish by dragging any of its edges, making it easy to see all fields in the database.

NOTE: It is at this point in creating your own reports that you may have to add and link additional tables. This Tutorial will not walk you through this step of report creation. Please refer to Chapter 19, Working with Databases, Page 403, and How to add and link multiple tables, Page 92.

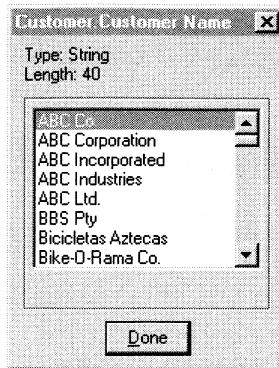
Inserting a field

You will now start placing objects onto your report by inserting the Customer Name field:

First, familiarize yourself with the Database Tab of the Insert Fields dialog box.

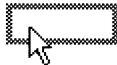


1. Highlight a field name in the dialog box by clicking the name once. When you highlight a field name, you can review the values for that field as well as the field type and size by clicking the *Browse Field Data* button. A scroll box appears listing the field name, type, length, and a subset of field values.



2. Select the Customer Name field and drag it into the details section of your report.

3. When you finish reviewing the data in the dialog box, click the *Done* button to return to the Insert Fields dialog box.
4. A placement box appears beside the arrow cursor as you drag the field over your report.

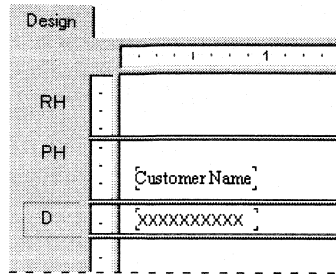


- The field box represents the object you have just selected for placement.
 - The size of the field box approximates the size of the data in the field selected.
5. Please move the field box as far to the left as you can in the Details section. If you move the field too far to the left, the arrow cursor will turn into a stop cursor indicating that you cannot drag the field that far. Keep in mind that you cannot place any objects outside the page margin.

A box containing X's appears and a text field box containing a column heading appears in the Page Header section directly above the field box.

NOTE: *The column heading, by default, is the name of the field selected.*

The Design Tab should look similar to this:



Understanding fields

Before we go any further, take a look at the field you just placed in the Details section.

- First of all, the field box indicates that when your report is printed, a field value will appear where the box is positioned.
- The X's in the field box (the field picture) indicate that the database field contains a text string. Other data types have different field pictures.
- The number of X's in the field box is the data width, the maximum number of characters in the field as defined by the database. The width of the field box is the field width (the amount of space allocated to the field for printing). Initially it is set to the width needed to display the maximum number of characters in the field (using the font selected for the field). You can change this width by resizing the field.
- The size of the X's indicates the point size selected for the characters in the field.
- The font and style (Bold, Underline, etc.) used in displaying the X's indicate the font and style selected for the characters in the field. You will learn how to make changes to these attributes later in the tutorial.
- The line spacing is adjusted to the point size selected for the characters in the field.

Handles, selecting fields

When a field selected, the field box appears with a handle (box) on its right, left, top, and bottom edge. These handles indicate that the field is selected, and therefore active. To do anything with a field (change the font, move it, etc.), you first have to select it.

- Position the cursor inside the field box and click once. The handles appear indicating the object is selected.
- Move the cursor away from the object and click in an empty part of the window. The handles disappear.

That is all it takes to select and deselect objects.

Adding additional fields

In this section, you will add two additional fields to your report. This time, however, you will add them at the same time by using the Ctrl-click combination.

1. Highlight the City field, press the Ctrl key on your keyboard, and highlight the Country field. Release the Ctrl key. If you scroll through the field list, you will notice that both fields remain selected. Using the Ctrl-click combination allows you to pick a non-continuous range of fields. You can use the Shift-click combination to pick several fields from the list that are adjacent to each other.
2. Click the *Insert* button to place the fields.
3. Move the cursor over your report. A placement box appears along with the arrow cursor.
4. Place the fields to the right of the Customer Name field. Click once to insert the fields. Both fields appear in the Details section of your report in the same order they are listed in the Insert Fields dialog box.

Reviewing your work



With three fields placed and positioned, let's see how your report looks.

1. Click the PRINT PREVIEW button on the toolbar to open the Preview Tab.

Your screen should look similar to this:

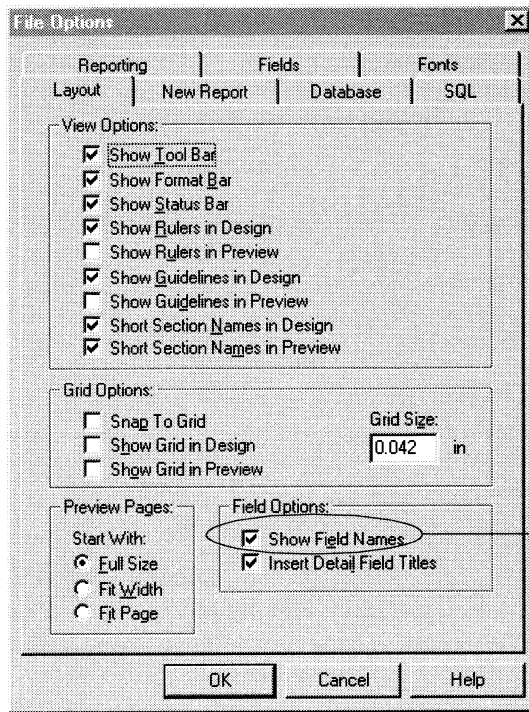
	Customer Name	City	Country
D	Bike-O-Rama Corporation	Sterling Heights	USA
D	The Pedallers Inc.	DeKalb	USA
D	Bikes 'R Us Enterprises	Blacklick	USA
D	CycleSporin Corporation	Huntsville	USA
D	Sporting Wheels Inc.	San Diego	USA
D	The Cyclists Company	Austin	USA
D	Ride 'Em Cowboy Corp.	Eden Prairie	USA
D	XYZ Enterprises	Des Moines	USA
D	Trail Blazer's Place	Madison	USA
D	The Cyclists Incorporated	Newbury Park	USA
D	Bob's Bikes Incorporated	Carskadonka	USA

You have the beginnings of a customer list report, but you still have several fields to add. When you are finished looking at your report, return to the Design Tab by clicking once on the Design Tab.

Displaying Field Names

Field pictures have been discussed, but there will be times when you want to see the field names in the field box in the Design Tab:

1. In the Design Tab, select the **OPTIONS** command the File menu. The File Open dialog box appears with the Layout Tab active. Search for *Setting up Crystal Reports* in Crystal Reports online Help.



2. Check the Show Field Names option and click OK when finished.

Now in the Design Tab you will see the actual field names instead.

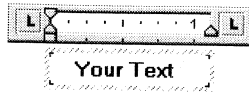
Combining database fields in text object

Instead of adding the Contact First Name and Contact Last Name fields as separate objects, you can insert both fields in a text object where you can control how you want them displayed. When you insert fields in a text object, the fields are automatically trimmed (i.e., they do not have any extra white space on either side). This is important because a field is a fixed size but the data in the fields can vary in size leaving various amount of unwanted white space.

1. Close the Insert Fields dialog box using the *Close* button so you can see your work clearly.
2. Click the INSERT TEXT OBJECT button on the toolbar. A field placement box appears next to the arrow cursor when you move the cursor over your report.



3. Insert the field to the right of the fields in the Details section. The Design Tab automatically scrolls to the right, if necessary, as you drag the field. When you click to place the object, a text object appears along with an in-place ruler. See *Inserting text objects*, Page 95, and *How to work with text objects*, Page 168.



4. Click once on the border of the text object to select it for resizing. Handles appear on all sides of the object.
5. Move the cursor over the right sizing handle of the text object and increase the width of the by about 1 inch. You may need to scroll to the right and continue resizing.
6. Double-click in the text object to select it for editing. Notice the insertion point is now flashing within the text object.
7. Reopen the Insert Fields dialog box by clicking the INSERT FIELDS button on the toolbar.
8. Select the Contact Last Name field. Remember, you can move the Insert Fields dialog box by dragging and dropping it by its title bar.
9. Click the *Insert* button to place the field on your report.
10. Move the cursor over the text object until your cursor becomes a drag and drop cursor. See *Crystal Reports cursors*, Page 58.
11. Click once to place the field in the text object.
12. Your cursor now appears after the Contact Last Name field in the text object. Type a comma and a space after the Contact Last Name.
13. In the Insert Fields dialog box, select the Contact First Name field.
14. Click the *Insert* button to place the field on your report.
15. Move the cursor over the text object until your cursor becomes a drag and drop cursor. Move the cursor to the



right of the comma and space you just typed, and click once. The database object will be inserted to the right of the comma and space.

16. You are finished inserting fields, so click *Close* on the Insert Fields dialog box.



17. Click the PREVIEW button to look at the fields you just placed.

Your report should now look like this:

	Customer Name	City	Country	
D	Bike-O-Rama Corporation	Sterling Heights	USA	Campbell, Matt
D	The Pedallers Inc.	DeKalb	USA	Manley, Stacy
D	Bikes 'R Us Enterprises	Blacklick	USA	Jackson, Daryl
D	CycleSporin Corporation	Huntsville	USA	Mast, Alexander
D	Sporting Wheels Inc.	San Diego	USA	Reynolds, Matt
D	The Cyclists Company	Austin	USA	Davidson, Heather
D	Ride 'Em Cowboy Corp.	Eden Prairie	USA	Smith, Alexander
D	XYZ Enterprises	Des Moines	USA	Holt, Christine
D	Trail Blazer's Place	Madison	USA	Belling, Alexandra
D	The Cyclists Incorporated	Newbury Park	USA	Hopkins, Anthony
D	Bike Bikes Incorporated	Coshocton	USA	Campbell, Matt

Adding titles using text objects

The report looks pretty barren without a report title. The next step is to insert a text object for the report title and then format the title. This time, however, you will do the work in the Preview Tab. This will make it easier to see your work while you add the report title. You are placing the title in the Page Header section because you want it to be displayed on every page of your report.



1. In the Preview Tab, click the INSERT TEXT OBJECT button on the toolbar. A field placement box appears next to the arrow cursor when you move the cursor over your report. See *How to work with text objects*, Page 168.

2. Place the object in the upper left hand corner of the Page Header (PH) section. A text object appears. Placing the text object in the Page Header section ensures that the report title is printed on every page of your report.
3. Double-click the text object to select it for editing.
4. Type Customer List in the text object.
5. To center the title, you will first need to expand the title field box so it's about the same width as the data in your report. To do this, click once on the border of the text object to select it for resizing.
6. Position your cursor on the right edge of the field box until the cursor turns into a resizing cursor. Drag the right edge of the field box until it is even with the right edge of the data in the Contact Name field object.

NOTE: You may need to release the field box, scroll the window to the right, then continue expanding the field box until it is even with the right edge of the Contact Name column.

You have created a large field that extends from the left edge to the right edge of your report.



7. Double-click inside the text object to select it for editing.
8. Center the title inside the field box by clicking the CENTER ALIGNMENT button on the format bar.
9. Highlight the text within the text object by dragging the I-beam cursor over it. See *Crystal Reports cursors*, Page 58.
10. Right-click in the text object and choose the CHANGE FONT command from the shortcut menu. The Text Format dialog box appears with the Font Tab active.
11. Set the report title to a larger, bolder version of the active font by choosing Bold from the *Style* drop-down box and 18 (or a point size suitable to the font you are using) from the *Size* drop-down box.
12. Change the color of the text by choosing Maroon from the *Color* drop-down box. Notice that the *Sample* box shows an example of how the text will look.

13. Click *OK* when finished. The title is now formatted to stand out on your report.

ADDING A HEADING FOR THE CONTACT NAMES

Now that you are familiar with text objects, you will use one to create a heading for the Contact Names.



1. In the Design Tab, click the **INSERT TEXT OBJECT** button on the toolbar. A field placement box appears next to the arrow cursor when you move the cursor over your report.
2. Place the object the Page Header (PH) section above the contact name object. A text object appears.
3. Double-click the text object to select it for editing.
4. Type Contact Name in the text object.

Saving your report



1. Click the **SAVE** button to save the work you have done on the report to date.

Since this is the first time you are saving the report, the File Save As dialog box appears with the directory already set to the directory where the database resides.

2. Type **CUSTLIST.RPT** in the *Filename* edit box and click *OK*. Crystal Reports saves your report to the **C:\CRW** directory or the directory in which Crystal Reports resides, and displays the new report name on the title bar.

Customer Name	City	Country	Contact Name
Bike-O-Rama Corporation	Sterling Heights	USA	Campbell, Matt
The Pedallers Inc.	DeKalb	USA	Manley, Stacy
Bikes 'R Us Enterprises	Blacklick	USA	Jackson, Daryl
CycleSporin Corporation	Huntsville	USA	Mast, Alex ander
Sporting Wheels Inc.	San Diego	USA	Reynolds, Matt
The Cyclists Company	Austin	USA	Davidson, Heather
Ride 'Em Cowboy Corp.	Eden Prairie	USA	Smith, Alex ander
XYZ Enterprises	Des Moines	USA	Holt, Christine
Trail Blazer's Place	Madison	USA	Belling, Alexandra
The Cyclists Incorporated	Newbury Park	USA	Hopkins, Anthony

Congratulations! You have just created a basic listing report. You will continue to refine this report during the rest of the tutorial

Record Selection

A first look at selecting records

The Select Expert allows you to limit or restrict the records that are to be included in a report. In this section you will learn how to:

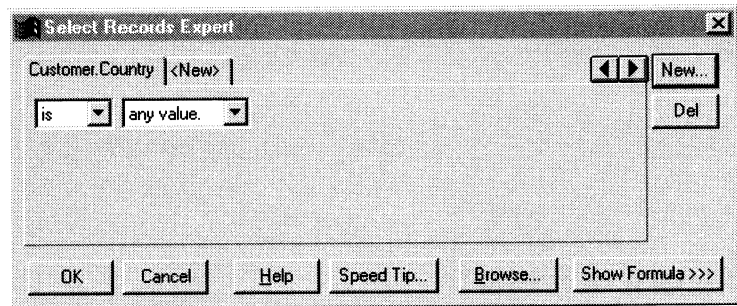
- select the records you want included in your report, and
- save a report including your selection criteria.

For example, it may be useful to have a customer list that only lists customers from the USA. The sample data contains records from the United States and International customers. Crystal Reports makes it easy to restrict lists like this using the Select Expert. See Chapter 14, *Record and Group Selection*, Page 315.

Entering your selection criteria



1. To begin, switch to the Design Tab.
2. Click on an empty area of your report to deselect any fields.
3. Click the SELECT EXPERT button on the toolbar. The Choose Field dialog box appears. This dialog box lists all the fields currently in the report and then lists all fields that are available from each table.
4. Since you are going to base record selection on the value that appears in the country field, choose the Country field from the Customer Table (Customer.COUNTRY) in the *Fields* list box and click OK. The Select Expert appears.



5. Your job in this dialog box is to imagine that you are completing the sentence:

```
Select all records where a customer's  
COUNTRY is
```

You complete the sentence with the condition you want the program to use when selecting records for your report. Right now the condition is any value, clearly not a restrictive condition. Click the arrow on the right drop-down list box to see what other options you have.
6. Since you want only those records where the Country is USA, select the *equal to* condition. A new list box appears on the right. The dialog box sentence would now read:

```
Select all records where Country is equal  
to
```

All that you need to complete the sentence is the value USA.

7. Click the arrow on the drop-down box. A list of all the country values appears. Select USA from the list.

Your sentence would now read:

Select all records where Country is equal to USA

8. Click OK to return to the Design Tab.
9. Click the PREVIEW button so you can review the results of your work. Scroll down through the report. Now, only customers from the USA appear in your report.



	Customer Name	City	Country	Contact Name
D	Bike-O-Rama Corporation	Sterling Heights	USA	Campbell, Matt
D	The Pedallers Inc.	DeKalb	USA	Manley, Stacy
D	Bikes 'R Us Enterprises	Blacklick	USA	Jackson, Daryl
D	CycleSporin Corporation	Huntsville	USA	Mast, Alex ander
D	Sporting Wheels Inc.	San Diego	USA	Reynolds, Matt
D	The Cyclists Company	Austin	USA	Davidson, Heather
D	Ride 'Em Cowboy Corp.	Eden Prairie	USA	Smith, Alex ander
D	XYZ Enterprises	Des Moines	USA	Holt, Christine
D	Trail Blazer's Place	Madison	USA	Belling, Alexandra
D	The Cyclists Incorporated	Newbury Park	USA	Hopkins, Anthony

10. Save this version of the report without overwriting the original report in the process by using SAVE AS command from the File menu and giving the new report the name USA.RPT.

Congratulations! You have started formatting your report and added selection criteria to it. More than that, you have learned how to manipulate your data. By now, you have a good idea of the powerful kinds of reports you can prepare with Crystal Reports. As you can see, it is an easy program to use.

Deleting the Country field

Now that your report only contains records from the USA, displaying the Country field in the body of the report is not necessary. We will quickly delete this before continuing.

1. In the Design Tab, select the Country field and the Country column heading by using the Ctrl-click combination.
2. Press the Delete button on your keyboard. That is all it takes to delete fields from your report.
3. Click the PREVIEW button so you can view your report without the Country field.



Your report should look like this:

Customer Name	City	Contact Name
Bike-O-Rama Corporation	Sterling Heights	Campbell, Matt
The Pedallers Inc.	DeKalb	Manley, Stacy
Bikes 'R Us Enterprises	Blacklick	Jackson, Daryl
CycleSporin Corporation	Huntsville	Mast, Alex ander
Sporting Wheels Inc.	San Diego	Reynolds, Matt
The Cyclists Company	Austin	Davidson, Heather
Ride 'Em Cowboy Corp.	Eden Prairie	Smith, Alex ander
XYZ Enterprises	Des Moines	Holt, Christine
Trail Blazer's Place	Madison	Belling, Alexandra
The Cyclists Incorporated	Newbury Park	Hopkins, Anthony
The Bike Experimented	Candlish	...

Balancing field spacing

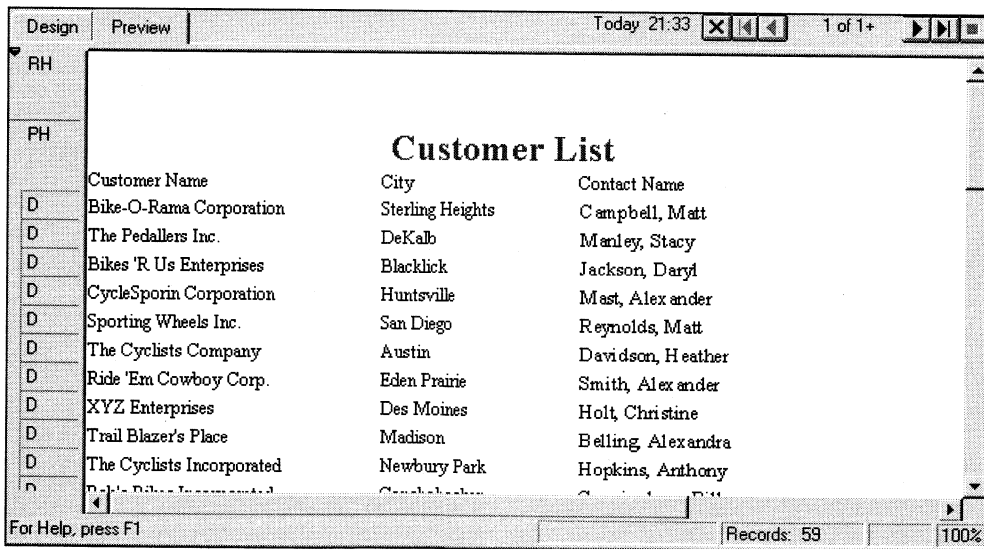
It looks as if there is more than enough white space on this page and you might be fine with the spacing as it stands, but it might be more readable if the columns were better balanced across the page.

You might wonder, if you placed Customer Name and City closer together on the report, where did the white space in the Preview Tab come from. The white space between Customer

Name and City occurs because the data in each of the fields is far shorter than the amount of space allotted for that data in the report. There is clearly some excess space that can be rearranged.

1. Return to the Design Tab by clicking once on the Design Tab. Select the Customer Name field and its field heading by using the Ctrl-click combination. Handles appear on both field boxes.
2. Place your mouse pointer over one of the two highlighted fields. Resize the fields to the left about one half inch.
3. To move the City field and its title closer to the Customer Name field by simply clicking on the guideline arrow head and dragging it to the left about one half inch.
4. Select the Contact Name field and its heading using the Ctrl-click combination.
5. Place your cursor over one of the two highlighted text objects and move them to the left, closer to the City field.
6. Click the Preview Tab and review your work again.

Your report should look like this:



Notice that all of your tools and commands are still available to you in the Preview Tab. You will find that sometimes it is more convenient to work on real data in the Preview Tab, while other times it is more convenient to work with field boxes in the Design Tab.

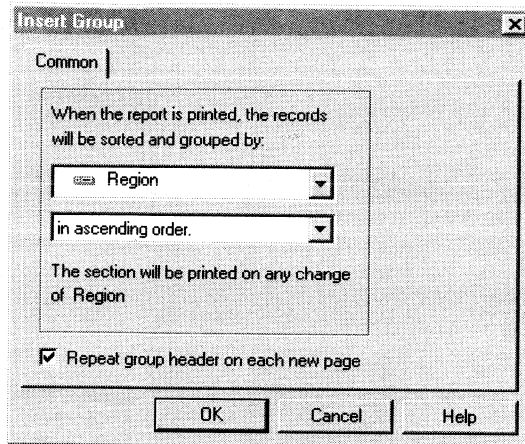
Grouping and Sorting

Reports can be grouped and sorted in a variety of ways. Sorting and grouping tools provide you with a great deal of flexibility for customizing your reports.

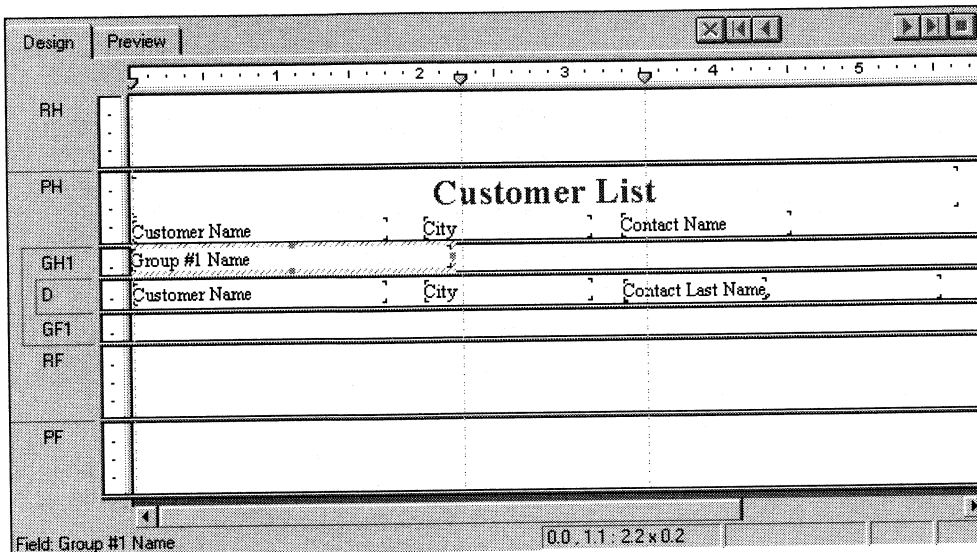
Grouping your report

In many reports you need to break your data into groups to make it easier to read and to understand. Crystal Reports lets you do this easily. But sometimes, you want to do more than group the data. You may want to summarize the values in each group (total them, count them, etc.). For this customer list, you will group the customers by region and sort the customers alphabetically within each group. See *How to group data*, Page 215.

1. In the Design Tab, choose the GROUP command from the Insert menu. The Insert Group dialog box appears.



2. Select the Region field from the Customer table in the first drop down box. The program will now take all records with the same value in the region field and place them together on your report in a group.
3. Choose *in ascending order* from the second drop down box. The region grouping will be displayed on your report in alphabetical ascending order
4. Click *OK*. Notice that two new sections now appear in the Design Tab: GH1 (Group Header) and GF1 (Group Footer). This is how Crystal Reports shows that the report has been grouped.



5. Click the Preview Tab to see what your report looks like.

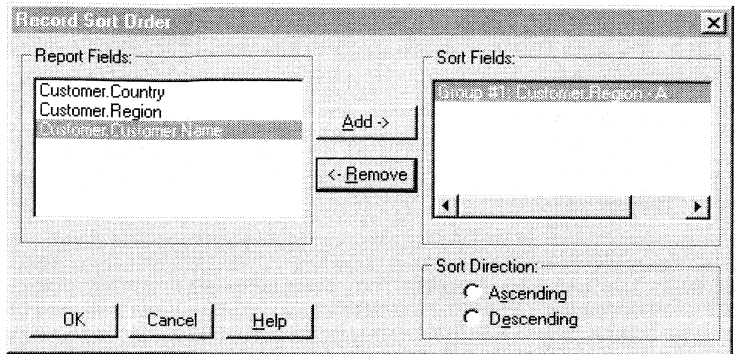
NOTE: For many of your reports you will want to insert summaries, subtotals and grand totals. For example, you were creating a sales report rather than a customer list, you would want to calculate the total sales amount for each region. See Chapter 9, Sorting, Grouping and Totaling, Page 201.

Sorting Records

On a typical Customer List report customer names are listed alphabetically. In our example, we want to see an alphabetical listing by Customer Name within each region. See *How to sort the records within groups*, Page 217.

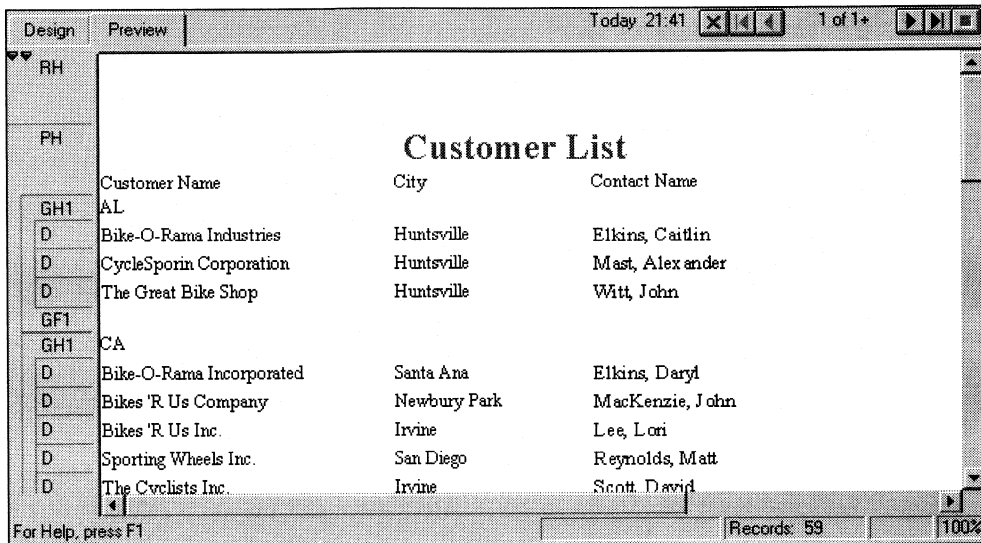


1. In the Preview Tab, click the SORT ORDER button on the toolbar. The Record Sort Order dialog box appears.



The *Sort Fields* box lists all fields currently on your report. You can choose to sort based on any of these fields. The *Sort Fields* box already displays the Region field that you just used to create a group. This indicates that the sorting you are about to do will be within each region, not for the entire report.

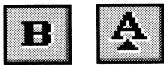
2. Select the Customer Name field and click the *Add* button. Notice that the Customer Name field now appears in the *Sort Fields* list box.
3. Select *Ascending* as the *Sort Order* and click *OK*.
4. Your report should now look similar to the following:



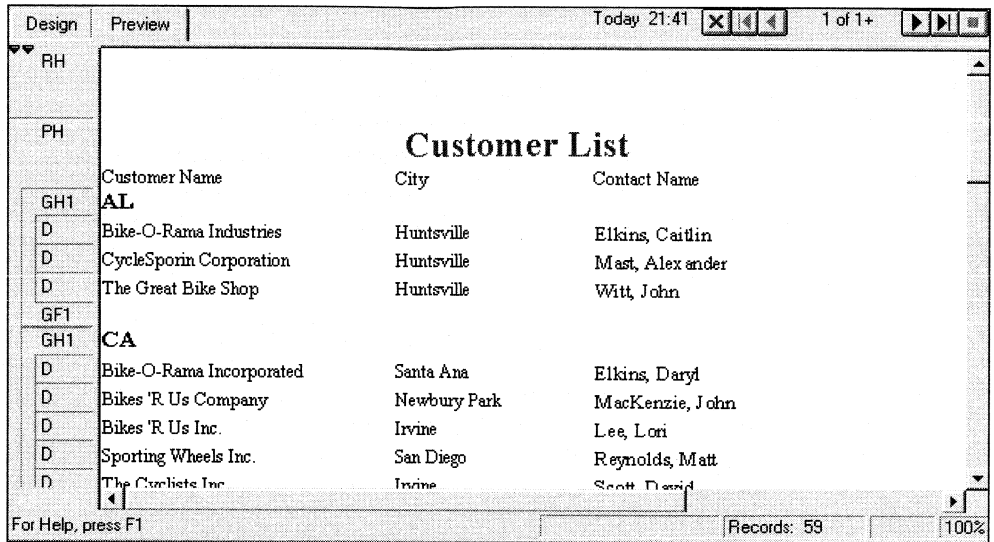
Understanding "Live" group headers

When you inserted a group, a group name field was automatically inserted in the Group Header section of your report. The group name field displays the current group's name. For example, if you group by region, when you preview your report, the group header for the CA (California) group will show "CA".

To finish up the customer list, you can format the group name field to highlight each group better.



1. In the Design Tab, select the Group #1 Name field.
2. Click the BOLD button on the format bar once.
3. Click the INCREASE FONT SIZE button on the format bar twice.
4. Click the Preview Tab and review your work again. Now the group name stands out.



Completing the Report

You have just two steps left to complete your report. You are going to add a company logo on the first page of the report and print out the completed report.

Inserting company logo

In this section, you will place a company logo on the top of the first page of your report using the INSERT PICTURE button.

1. In the Design Tab, click the INSERT PICTURE button on the toolbar. The Open dialog box appears.
2. Choose the CRAZEC.BMP file and click *Open*. The Open dialog box disappears, and a large box appears along with the cursor when you move the cursor over your report. The large box represents the graphic logo you will place.

NOTE: The file *CRAZEC.BMP* is a 16 color bitmap created in Microsoft Paintbrush. A second file, *CRAZEB.BMP*, was created as a grayscale image and will print to a printer more clearly than the color image. If you prefer, use *CRAZEB.BMP* in the above step instead of *CRAZEC.BMP*.

- Position the rectangle in the upper left hand corner of the Report Header section of your report. Placing the logo in the Report Header section ensures that the logo is printed only on the first page of your report.

NOTE: Although there does not appear to be enough room in the Report Header section when you place the graphic, the RH section will automatically expand to accommodate the picture.

Printing a hard copy



- Save your report by clicking on the SAVE button on the toolbar. If you want to print a hard copy of your report, click the PRINTER button on the toolbar. The Print dialog box appears. Select the pages you want to print, the number of copies you want, and toggle the *Collate Copies* check box on if you want your copies collated. When you click OK, the program prints your report.

NOTE: See Hands-On (Distributing Your Report), Page 110, for information on other means of distributing your report.

The screenshot shows a software window titled "Design Preview" with a status bar at the top indicating "Today 21:41" and "1 of 1". The report content is displayed in a table format. The top section is the Report Header (RH) containing a large, stylized logo for "CRAZE". Below this is the Page Header (PH) with the title "Customer List". The main body of the report (GH1) is a table with three columns: "Customer Name", "City", and "Contact Name". The table is divided into two sections by state abbreviations: "AL" and "CA".

Customer Name	City	Contact Name
AL		
Bike-O-Rama Industries	Huntsville	Elkins, Caitlin
CycleSporin Corporation	Huntsville	Mast, Alexander
The Great Bike Shop	Huntsville	Witt, John
CA		
Bike-O-Rama Incorporated	Santa Ana	Elkins, Daryl
Bikes 'R Us Company	Newbury Park	MacKenzie, John
Bikes 'R Us Inc.	Irvine	Leg, Lori

At the bottom of the window, a status bar shows "For Help, press F1", "Records: 59", and "100%".

Congratulations! You have just completed your first report with Crystal Reports. It is a pretty impressive report, and Crystal Reports made it easy to do.

6

Reporting 101

What you will find in this chapter

Basic Report Design 148

How to design a paper prototype with Crystal Reports 156

Concepts in Reporting 157

Formatting your report 163

Report Distribution 163

Beyond Basic Reports 163

Basic Report Design

The purpose of this section is to suggest a structured approach to preparing reports with Crystal Reports. The approach includes the following elements:

- deciding on the content of your report,
- developing a prototype on paper,
- setting up the prototype using Crystal Reports,
- specifying the records/groups to be included in the report,
- manipulating the data with formulas and functions,
- grouping, summarizing, and sorting your data,
- editing and formatting the data,
- adding graphic enhancements and OLE objects, and
- printing the finished report.

This section has been designed to provide you with a conceptual understanding of the reporting process.

Deciding on the content of your report

Before you do anything else, you should outline the information you want your report to provide. Use the following list of questions as a guide in making that outline:

- What is the overall purpose of the report?
- Who is going to read the report?
- What is the report title going to be?
- What information do you need besides the title to identify the report?
 - Where is that information to come from?
 - If the information exists in a database table, what types of fields is the data stored in: number, text, etc.?
- What identifying information do you want to appear at the bottom of each page?
 - Where will that information come from?

- If the information exists in a database table, what types of fields is the data stored in: number, text, etc.?
- What specific data do you want to appear in the body of the report?
 - Where will that data come from?
 - Does that data exist in data fields or does it need to be calculated from data field values?
 - What types of fields is the data stored in: number, text, etc.?
 - Do you want data to show only if certain conditions are met?
 - Do you want your data sorted?
 - How?
 - Do you want the data broken into groups?
 - What kind of groups?
 - Do you want your data summarized?
 - Subtotaled?
 - Some other kind of summary value?
 - What do you want to total?
 - Do you want to show summaries for all the groups or just the top or bottom groups?
 - Do you want the group summaries sorted in any special order?
 - Do you want to add text labels to the summaries?
- What information, if any, do you want flagged on the report?
 - How do you want it flagged?
 - By color?
 - By a text flag?
- What other information do you want highlighted in some way so that it really stands out?
 - How do you want it highlighted?

- Colored text?
- Special font or font size?
- Borders or background color?
- Do you want to label the highlighted information as well?
- Do you want the report to be based on all records in the database or only on specific records?
 - Which records?

Purpose

What is the overall purpose of the report?

Reports are management tools. Their purpose is to help individuals quickly grasp the essential elements and relationships found in raw data so they can make effective decisions. For a report to be effective, it has to present the right data in a logical way. If it presents the wrong data or if it presents the right data in a haphazard manner, the report may slow the decision making process or even encourage incorrect decisions.

A good starting place in the development of a report is to write out the purpose of the report in a sentence or two. The purpose statement helps you focus on your primary needs, and it gives your report both a starting point and a goal. Here are some examples of purpose statements:

- The purpose of this report is to show monthly and year-to-date sales by sales representative, compare this year's numbers to last year, and flag representatives whose sales figures do not meet company standards.
- The purpose of this report is to show sales activity for each item in inventory, and to suggest reorder quantities based on that activity.
- The purpose of this report is to calculate bowling averages and handicaps for each member of the bowling league.

Clarifying the purpose of the report before you start is a critical step in the overall process. A report without a clear purpose is like a meeting without a clear agenda; it rambles and accomplishes little.

Readers

Who is going to read the report?

A single report is often used by many individuals. A detailed, company-wide sales report, for example, may be used by sales representatives, the regional sales manager, the national sales manager, and the Chief Operating Officer (COO).

Each of these individuals will be interested in different aspects of the report.

- A sales representative will use the report to evaluate his/her individual sales performance and to compare that performance to that of other representatives in the region.
- The regional sales manager will use the report to evaluate the representatives in his/her region and to compare the region's performance to that of other regions.
- The national sales manager will use the report to evaluate the performance of his/her regional managers and to compare overall sales to the current sales forecasts.
- The COO will use the report to evaluate the performance of the Vice President of Marketing and the sales department as a whole, and to project such things as manufacturing needs, warehouse locations, etc.

Since each of the users of the report has different interests, it is important to plan the report so it includes the information each of the users is looking for.

Title

What is the report title going to be?

Write out a working title for your report. You may decide to change it later, but at least you will have a title to use when creating your prototype report. See *How to add a title page to your report*, Page 108.

Do you know the data you want to use in your report?

Do you know the type of database you are reporting from and whether to choose Data File, SQL/ODBC or Dictionary when you want to report off the data in Crystal?

If you do not know, ask an internal source for the database type and location of your data and to set you up with access to that database if necessary. See Chapter 20, *Data Sources*, Page 475.

Are you familiar enough with your data to find the information you want? When you are looking for the Customer Contact name, can you find the field in a database table?

Your MIS professional, database administrator or coworkers will have to help you become familiar with your data. Just playing with some basic listing reports is also a good opportunity to see what information is available and how it is stored.

Header information

What information do you need besides the title to identify the report?

Are you going to include identifying information in addition to your report title? You may wish to include the current date, information on who prepared the report, a block of text to describe the purpose of the report, the range of data covered, or something similar. If you are going to include such information, write it down so you can use it in preparing your prototype.

Header information sources

Where will that information come from?

The information can come from a variety of sources, depending on the kind of information you plan to use. For example, the current date can be inserted using the PRINT DATE FIELD command on the Insert | Special Field menu). Information on who prepared the report might be drawn from individual data fields in the database table(s) used. (If it is to be drawn from a database table, what table? Or, what combination of tables?) A block of text can be created as a text object and placed anywhere on the report. As you begin to think of where the information is to come from, you begin formally structuring the report.

Data types in the header

If the information exists in a database, what types of fields contain the data: number, text, etc.?

Crystal Reports uses different rules for working with different types of data. You will find it helpful later if you note the data type of each piece of data you plan to draw from a database.

Footer information

What identifying information do you want to appear at the bottom of each page (page number, page x of y , report name, author's name, the word "Confidential")? See *How to insert a Page of N field*, Page 95.

Footer data sources

Where will the information come from?

Data types in the footer

If the information exists in a database table, what types of fields is the data stored in: number, text, etc.?

Report body data

What specific data do you want to appear in the body of the report?

When you think of a report, it is probably the body of the report that you think of. The body should contain all the data that you need to fulfill the statement of purpose you wrote for the report. It should also contain all of the data needed by the various users that you have identified. You might find it helpful to list first the basic data that is required to fulfill the purpose statement, and then list the more specific kinds of data needed by the various users.

Body data sources

Where will that data come from?

This step requires you to look at the available database table(s). Crystal Reports allows you to combine data from different databases to create your reports, so you have a great deal of flexibility in your work.

- Much of the data in a typical report will be taken directly from data fields. Which data fields will you be using and where are they located?
- Other data will be calculated based on data fields. Which data fields will be used in the calculations?
- Still other data will be input directly into the report using text objects (headings, notes, labels).

Existing or calculated?

Does that data exist in data fields or does it need to be calculated from data field values?

Some report information can be drawn directly from data fields (sales information, for example); other information will have to be calculated based on data field values (sales commission, for example, based on the relationship of sales to quota). In your

planning, it can be helpful to segregate or flag data that needs to be calculated from that which can be used directly. See Chapter 10, *Formulas 101*, Page 249, and *How to flag values that meet certain conditions*, Page 199.

Data types in the body

What kinds of fields contain the data: number, text, etc.?

While it is important to understand data types for all data you will be using, it is of critical importance that you know the data type for data fields that will be used in calculations. Functions and operators work with specific kinds of data, so it's important to know the data type to know which functions and operators you can use in your calculations.

Record or group selection

Do you want the report to be based on all records or groups in the database or only on specified records or groups?

Crystal Reports gives you the opportunity to base a report on all records in a given database, or on a limited set of records from the database. Using Crystal Reports you can select records based on simple date ranges or comparisons, or you can create complex formulas to identify the records to be included. Take a few minutes to determine the records needed for your report and list the criteria to be used for selecting those records. See Chapter 14, *Record and Group Selection*, Page 315.

Groups

Do you want your data organized into groups? How do you want it grouped? By customer? By date? Or by other criteria? See *How to group data*, Page 215.

Group values

Do you want to show a subtotal at the end of each group? A count? An average? Crystal Reports allows you to specify several kinds of group values. See *How to summarize grouped data*, Page 219, and *How to subtotal grouped data*, Page 221.

Group value positions

Where do you want the group values to appear? With the group data? With the group data but on a page separate from other groups? Only at the bottom of the page?

Crystal Reports gives you all of these options.

Grand totals, subtotals, averages, etc.

Do you want to total (or average, count, or determine the maximum or minimum value included in) all the values in any column on your report?

Crystal Reports allows you to do this and place the grand total (or the grand total average, grand total count, etc.) at the bottom of the selected column.

Flags

What information, if any, do you want flagged on the report?

You may want to call attention to some data by flagging it on your report. For example, non-moving inventory items are often flagged on inventory reports so they can be given special attention. If you want any information flagged, identify the information and the conditions that will trigger the flagging. Using the inventory report example, you might want to flag each item that has shown no activity during the last month, during the last three months, or during some defined period. See *How to flag values that meet certain conditions*, Page 199.

Flag options

How do you want it flagged?

You may want to flag items with an asterisk or some other symbol, or you may want a word to appear as a flag. In any case, you should write out flagging instructions so they are handy.

Highlights

What information do you want highlighted in some way so that it really stands out?

Crystal Reports gives you the opportunity to underline report elements, or to change the fonts or font size or color used for specific report items. It allows you to put borders around items and to draw lines and boxes to break your report into sections, set off headings, etc. All of these formatting tools can be used to highlight key data on a report. If you have data that you want highlighted, you should write down highlighting instructions so they are handy too. See Chapter 8, *Formatting*, Page 183.

Sorting

Do you want your data sorted based on record or group values?

Crystal Reports gives you both alternatives. See *How to do a single field sort*, Page 201.

Developing a prototype on paper

Graphic designers generally begin their work on a magazine cover, brochure, or display advertisement with a rough pencil sketch. They often use boxes, circles, or other symbols to represent the graphic elements they intend to include in the final product, and they often use lines or scribbles to represent text. Doing the rough design on paper, in pencil, helps them create a look for each page. It helps them find a balanced way of positioning the various elements before they begin working with sophisticated graphics tools. We think you will find a similar exercise helpful in designing your reports.

While a paper prototype is useful regardless of your expertise with Crystal Reports, it is particularly valuable when you are first learning to use Crystal Reports. With the paper prototype in hand, you can put your full effort into learning and using the Crystal Reports commands instead of trying to design and learn at the same time.

How to design a paper prototype with Crystal Reports

- Get paper of the size you will be using for your finished report.
- Position your title and other descriptive header information, using boxes or lines to represent report elements.
- Position your footer information.
- Review the page for balance.
- Look at the information you intend to include in the body of your report.
 - Count the number of fields you will be using and estimate the appropriate spacing between fields.
 - Use rectangles to pencil in the fields using your estimated spacing.
 - Change the spacing if you need to.
 - Decide on a logical sequence for presenting the data in the body of the report.
 - Label the fields to indicate that sequence.
- Use small boxes to indicate group values and totals.

- Place some random flags in the column where you want the flags to appear.
- Darken any elements you want highlighted so they stand out from the rest of your prototype.
- Review your finished product for look and balance, and make changes as needed.

Concepts in Reporting

The purpose of this section is to give you a conceptual understanding of the tasks necessary to create a fairly standard report such as that created in the *Tutorial*, Chapter 5. Each topic is discussed in relation to the *Tutorial*; please refer to Chapter 5 throughout for an illustration of these concepts.

The concepts are presented in the order you may use to create such a report and sources of additional information will be provided.

Report Expert, Custom Report or Custom?

Each time you create a report in Crystal Reports, you can use the Report Expert, Another Report, or Custom Report option. You will probably use all options at one point in time.

EXPERTS

The Reports Experts help you create reports as quickly as possible and many new users and developers alike prefer to create the majority of their reports using Experts. All you have to do is choose the Expert that most closely matches your report type. The Expert walks you through the process of creating your reports step-by-step.

You can quickly create the report and see how it looks against your actual data. And best of all, if you then decide you want to make changes you can get back to the Report Expert to further modify your report. Search for Experts in Crystal Reports online Help.

ANOTHER REPORT

If you want to build a new report based upon one that already exists, click the *Another Report* button. The program will make a

duplicate of the original report, which you can modify however you please to create your new report. Use this option whenever you think they can save you time. Some of the times templates are useful are:

- when you need to create a new report with a different grouping or different record selection than an existing report,
- when you need to create a series of reports, each a little different than the last,
- when you need to reconstruct a report based on an earlier time period using the same report structure you use today, or
- when you need to create an entirely new report based on a set of databases that are linked in another report. You can create a report and delete the fields from the Design Tab without disturbing the underlying links. Then, without relinking, you can build all your new reports based on this report.

CUSTOM REPORT

The Custom option is used when you want to create your report from scratch. This is used often when you want the full flexibility and control of building your reports from the ground up or if your report type is different than the many report types available in the Experts.

For example, this is the option used in the *Tutorial* in Chapter 5, but it was chosen in that case to fully illustrate the basics of reporting.

Selecting Your Data

The next step when creating your reports, is to select the data you want to use in your report. This is a two step process. The first step is to determine what type of data you want to work with and the second step is to actually select the data.

The next four sections discuss the different data types you can use with Crystal Reports.

DATA FILE

Click the *Data File* button if you want to use any of the standard (not client-server) PC databases. These are typically databases whose data and all software used to access that data are on a single machine. Crystal Reports can access many of the most

common PC database formats directly. In other words, Crystal Reports has built-in capabilities to directly open database files and tables designed in dBASE, FoxPro, Clipper, Btrieve, Paradox, and Microsoft Access, among others. Once Crystal Reports is installed on your system, you can immediately begin creating reports based on these databases simply by selecting the appropriate file. See Chapter 20, *Data Sources*, Page 475.

The *Tutorial* in Chapter 5 uses this option because the sample data, CRAZE.MDB, is a Microsoft Access database.

SQL/ODBC

Click the *SQL/ODBC* button if you want to use SQL or ODBC data sources.

- Structured Query Language (SQL) databases are perhaps the most popular and most powerful database formats. They usually work over a client/server network architecture and they use:
 - an SQL server to create, store, and manipulate database files, tables, fields and records, and
 - an SQL Client interface allowing workstation users to retrieve data.
- Open Database Connectivity (ODBC) is a standard developed by the Microsoft Corporation through which many different types of data can be accessed by a single application. An application need only communicate with one set of files, ODBC, to instantly be able to work with any source of data that can be accessed by ODBC.

Crystal Reports provides direct drivers for many of the most popular SQL systems, and ODBC capabilities as well.

DICTIONARIES

Instead of directly selecting a database table for building a report, you may be able to select a Dictionary if your company uses Dictionaries. A Dictionary is a one-stop, ready-to-use source of data.

Dictionaries let you concentrate on the things that are important to you when designing reports. They let you get out the reports you need in a hurry without your having to wade through

extraneous information or deal with the complexities of locating and linking tables.

There is nothing technical about using a Data Dictionary. With a Dictionary, instead of using multiple databases, struggling with links, building formulas from scratch, locating graphics, and decoding cryptic field names, you simply select the pieces that you want, and build your report. It couldn't be easier. See *How to Select a Crystal Dictionary for a Report*, Page 401.

CRYSTAL QUERIES

A query is simply a request for specific information from a database. If you are reporting on large SQL databases (or databases accessed via ODBC), a Crystal Query will pass much of the processing off to the server. The data the query generates can then be used to create reports. See Chapter 17, *Queries*, Page 359.

Linking

If your report contains data from two or more database tables you will need to link at this point when creating your reports. You link database tables so records from one database will match related records from another. For example, if you activate a Suppliers table and a Product table, you link the databases so that each product (from the Product table) can be matched up with the supplier that made the product (from the Supplier table).

Linking is not necessary in the *Tutorial* in Chapter 5 because all the data comes from the Customer table. The majority of your reports will probably require data from two or more tables so linking will be necessary. The process of linking is made easy in Crystal Reports with the Visual Linking Expert. See *How to add and link multiple tables*, Page 92, and search for *Visual Linking* in Crystal Reports online Help.

NOTE: *You will never find it necessary to link tables in reports from a Crystal Dictionary or Crystal Query because any links required by the data have already been set up (dictionaries) or processed (queries).*

Placing data on your report

Placing data on your report is a very important task when creating your reports. You have to know what type of data you want to place on your report and also where on your report you want to place them.

DATABASE FIELDS

Much of the data you place on your report will be database fields, displaying data as it is stored in the database. For example, in the *Tutorial* in Chapter 5, the Customer Name, City and Country fields are placed on the report. Normally, you will place database fields into the Detail section, but in certain circumstances you will place them in other sections of your report. See *How to insert database fields*, Page 93.

TEXT OBJECTS

Text objects will be used in your reports for a multitude of purposes. They are a powerful way of inserting titles; labelling summaries and other data on your report; and for easily combining database fields. For example, in the *Tutorial*, you use text objects to easily display the two contact name database fields as one object; to insert a column heading for this contact name; and to insert a title in your report. See *How to insert text objects*, Page 95, and *How to work with text objects*, Page 168.

SPECIAL FIELDS

To display information such as Page Numbers, Print Date and Report Comments use the commands on the Insert | Special Field menu. See *How To Insert Special Fields*, Page 94, and Chapter 23, Special Field commands, Page 547-549.

FORMULA FIELDS

If you want to display data that is a calculated value, you will need to create a formula field and place that formula field on your report. For example, if your database only stores the order and ship dates for orders and you need to display the number of days it takes to ship the order, you will need to create a formula field that will calculate the number of days between ordering and shipping. This is just one simple example of the use of formula fields. Refer to Chapter 10, *Formulas 101*, for an introduction to formulas.

Formatting Data

At this point in creating a report you may want to do some basic formatting. Perhaps you would like to change the font size and style of a text object used as a title. Or if you have a number field such as a sales figure, you might want to place a dollar sign before the number or change the number of decimal places displayed.

For example, in the *Tutorial*, it is at this point that you format the title and add a text object to name the Contact Name information.

Record Selection

Record Selection, the task of paring down the data in your reports to include only the data required for your report, is a crucial step in report creation. You will rarely want a listing of all the information in a database. Most often you will be interested in only the sales in a given time period or for a certain product, etc. For example, a sales report may be designed to only include sales from one product line for the last calendar month.

The sample data used for the *Tutorial* has information from the United States and International customers. Record selection is used to create a report that lists only those customers in the USA. See Chapter 14, *Record and Group Selection*, Page 315.

Grouping, sorting and summarizing your data

Once you have created a basic report you will want to organize information by grouping related information, sorting individual records, and summarizing, subtotaling and grand totaling.

GROUPING RECORDS

To organize your data, you may want to group related data together. For example, after grouping the Customer List report in the *Tutorial* by region, the customer list is divided into region groups. That way, a sales manager for the California region could quickly locate the California group and see the customers within their region only. See *How to group data*, Page 215.

SORTING RECORDS

The SORTING RECORDS command allows you to specify the order in which you want the records on your report displayed. For example, after grouping in the *Tutorial*, you sorted the records within each region in alphabetical order by Customer Name. Many of your reports will use some type of sorting. Depending on the report, you will sort the records in a list or sort in conjunction with grouping. See *How to do a single field sort*, Page 210.

SUMMARIES, SUBTOTALS AND GRAND TOTALS

Many of your reports will use some sort of totaling. For example, in a North American sales report grouped by state, you might want to calculate the total dollar amount sold in each state. You do

this by creating a subtotal on the sales field. Summaries are also used at the group level, but they allow you to calculate averages, counts and other group (aggregate) values. For example, in a sales report you may want to calculate an average of sales per state (average summary on the sales field) and calculate the number of products sold in the state (distinct count of the product name field). Use the GRAND TOTAL command to calculate totals for the entire report.

The Customer List in the *Tutorial* does not require summaries, subtotals or grand totals. See *How to summarize grouped data*, Page 219, and *How to subtotal grouped data*, Page 221.

Formatting your report

At this point your report is almost complete, but may need some additional formatting. For example, it was at this point in the *Tutorial* that we inserted a company logo. See Chapter 8, *Formatting*, Page 183.

Report Distribution

In the *Tutorial* you simply print your report, but there are many other ways to distribute reports. You can choose to export your reports in the Crystal Report format. If you want to e-mail your report to someone who does not have Crystal Reports, you can export the report in MSWord or Excel format. You can even publish the report on the Internet. See *How to export reports*, Page 110.

Beyond Basic Reports

Once you are comfortable with the basics of reporting in Crystal Reports, you will be ready to investigate powerful reporting features such as graphs, OLE Objects, form letters, subreports, Cross-Tabs, multi-section reports and much, much more.

7

Multiple Section Reports

What you will find in this chapter...

Using Multiple Sections in Reports 166

How to prevent variable length objects from overwriting each other 167

How to work with text objects 168

How to create a form letter using a text object 170

How to print conditional messages in form letters 177

How to format objects conditionally 178

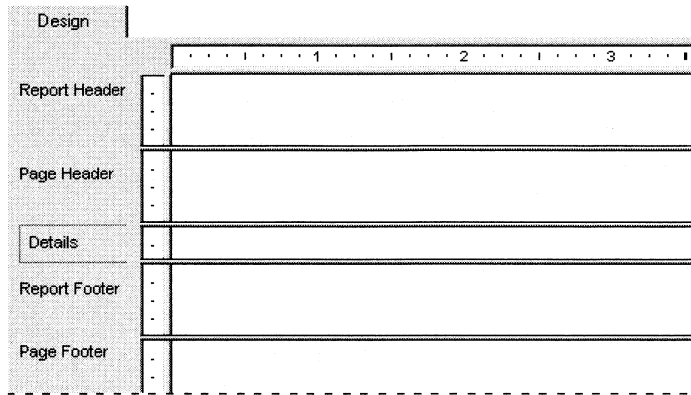
How to alternate background colors for rows 179

How to eliminate blank lines 180

How to add blank lines conditionally 181

Using Multiple Sections in Reports

By default, Crystal Reports gives you five design areas to use when building your report: Report Header, Page Header, Details, Report Footer, and Page Footer.



Each area contains only a single section when you first begin your report. There are certain reporting tasks that you can perform most efficiently by creating multiple sections within an area, such as:

- keeping variable length objects from overwriting each other,
- putting conditional messages in form letters,
- formatting objects/sections differently based on field values,
- alternating background colors on a row-by-row basis,
- eliminating blank lines when fields are empty, or
- adding blank lines under specific conditions.

This chapter will give you hands-on instructions for performing each of these tasks using multiple section reports.

When you understand the power of multiple sections, you will discover even more ways that you can use them to produce the report effects you want.

For more information see *How to add, delete, move, and merge sections*, Page 83.

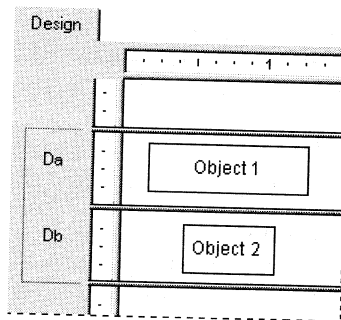
HANDS-ON

How to prevent variable length objects from overwriting each other

When you place subreports or other variable length objects above other objects in one section of your report and you have the *Can Grow* property for the variable length object toggled on in the Format Editor, that object may overprint the bottom object unless you have:

- expanded the section to fit the maximum size of the object, and
- spaced the objects, allowing enough space for the first object to complete printing before the second one begins. See Chapter 8, *Types of Formatting Properties*, Page 186.

You can eliminate this overprinting problem by creating multiple sections in an area and placing the object below the variable length object in its own section.



Now, when the report runs, the section with the variable length object will finish printing before the section below it prints and you will get the results you want. See *Two Unrelated Reports*, Page 345.

NOTE: *Subreports are not the only kinds of objects that can cause this problem. Memo and BLOB fields can cause the same kind of problem as well.*

How to work with text objects

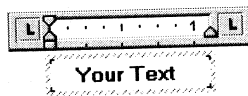
You will use many of the capabilities of text objects as you create form letters. A brief discussion of text objects should make it easier for you to create the form letter in the next section.

- A text object can contain both text and fields. You are going to use both in this example.
- You can resize text objects, and you will be resizing the text object so it prints as a letter.
- Text objects operate in two modes: the move/resize mode and the edit mode.
 - When the object is in the Move/Resize mode, it appears as a broken line frame with resizing handles.



In this mode, you can resize it by dragging any of the resizing handles or move it by placing the cursor inside the object and dragging it to a new location. You can also insert fields in this mode but you cannot insert text. You put a text object into the move/resize mode in one of two ways:

- by clicking the text object when it is inactive, or
- by clicking the frame when the object is in Edit mode.
- When the object is in the Edit mode, it appears as a broken line frame without sizing handles and with an in-place ruler above or below the frame, depending on its position in the report.

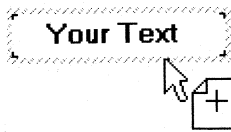


- When you first place a text object, the program sets it in the Edit mode. You can also put a text object into the Edit mode by double clicking it if it is inactive or in the Move/Resize mode.

- Each text object contains mini-word processor capabilities including the ability to change the fonts for individual characters and fields, and automatic word wrap. In the Edit mode you can insert text and such non-text objects as database fields and formulas. Whenever the object is in Edit mode, it contains an insertion point, a flashing vertical line that indicates the position that typed text or inserted fields will begin.
 - The insertion point moves as you type, automatically staying to the right of the last character.
 - It also moves when you insert a field, automatically staying to the right of the field
 - It moves one character position at a time when you press the space bar.
 - It moves down one line, to the inside left edge of the text object when you press Enter.
 - It moves to your cursor position when you click anywhere within the existing text.

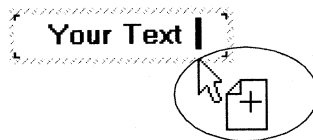
When you work through this tutorial, it is always expected that you will type or place fields at the existing insertion point unless you are told otherwise.

- To select text in a text object (to delete it, to change a font, and so on), you position the cursor over the text and, when the I-beam cursor appears, you drag the cursor to highlight the text you want to select.
- To select a field in a text object, you position the cursor over the field and, when the I-beam cursor appears, you right-click.
- To insert text, you type in the text you want and it appears at the insertion point.
- To insert a field, you must insert it using a menu command, a button, or a dialog box. You cannot simply drag a field into a text object from elsewhere in the report. When the program displays the placement frame, move the frame to the text object. The placement frame changes to a drop cursor when it is in a position where the field can be inserted in the text object.

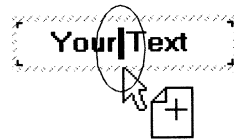


NOTE: It is critical that you see the drop cursor before you place the field. If you do not see it, you can place the field so it overlays the text object instead of being inserted in it. It may appear to be inside the text object, but if you move the object the field will not move with it.

- The insertion point is tied into the drop cursor. If you have existing text or fields in the text object, the insertion point moves as you move the drop cursor, enabling you to pick the exact position where you want to place the field. The program always places the field at the insertion point.



As you move the drop cursor . . .



the insertion point moves.

See *How to insert text objects*, Page 95.

How to create a form letter using a text object

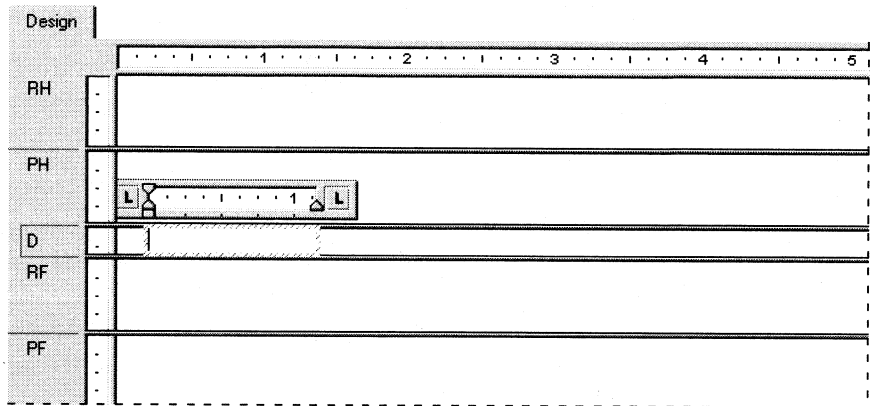
While form letters themselves are not necessarily multi-section reports, they are often used in multi-section reports to generate custom mailings. The next section, *How to print conditional messages in form letters*, explains how to use multiple form letters or multiple versions of the same form letter for custom mailings. This section shows you how to create a form letter in the first place.

You are going to use a text object to create a form letter. The form letter you create will be tied to a database table so that each letter is customized with company information from a different record.

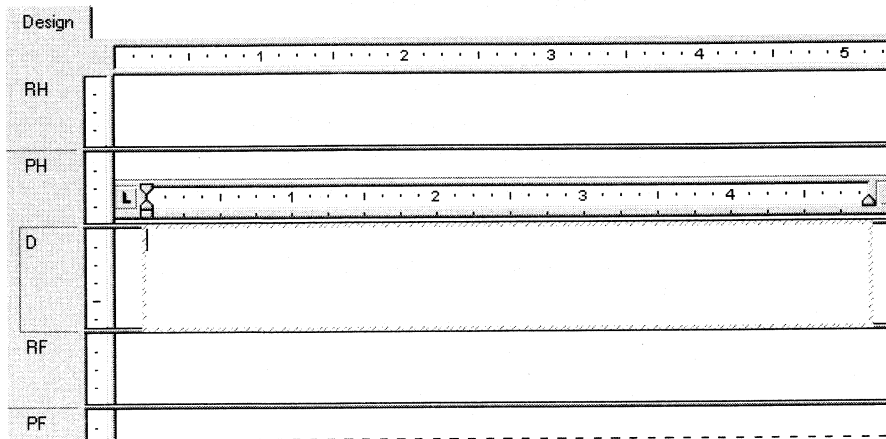
If you have difficulty performing any of the steps in this tutorial, please refer to the previous section, *How to work with text objects*.

CREATING THE FORM LETTER

1. Create a report using CRAZE.MDB. The Design Tab appears with the Database Tab active in the Insert Fields dialog box.
2. Since you do not want field titles to appear above the fields that you insert into the letter, choose OPTIONS from the File menu and toggle the *Insert Detail Field Titles* option off in the Layout tab.
3. Insert a text object in the Details section of your report.



4. Click the text object frame to put the object in move/resize mode.
5. Drag the resizing handle on the right side of the object to the right edge of the Design Tab. This will make the object about eight inches wide, the approximate width of a page. You may have to stop resizing, scroll the window, and resize some more to accomplish this.



6. Drag the bottom of the object down about an inch to give yourself some working room. The Details section will expand to fit the object.
7. Double-click inside the text object to place it in edit mode, ready for you to begin your work. When you do this, the insertion point will be positioned at the extreme left, inside the object.

The letter will consist of a date, an inside address, a salutation, a one paragraph body, and a closing section.

NOTE: When you place a database field or special field in a text object, you cannot just drag it in from elsewhere in your report. You need to insert it using a menu command or button and then drag it into the object until you see the drop cursor. If you place it without seeing the drop cursor, the field will overlay the text object instead of being inserted in it. In such a case, if you move the text object, the field will not move with the object.

Date

1. To put a date into the letter, choose the PRINT DATE command from the Insert | Special Field menu, drag the placement frame into the text object, and place it at the insertion point.

NOTE: If you want to change the way the date is formatted in the letter, right-click the Print Date field, choose FORMAT FIELD from the shortcut menu, and make your modifications on the Date Tab of the Format Editor when it appears.

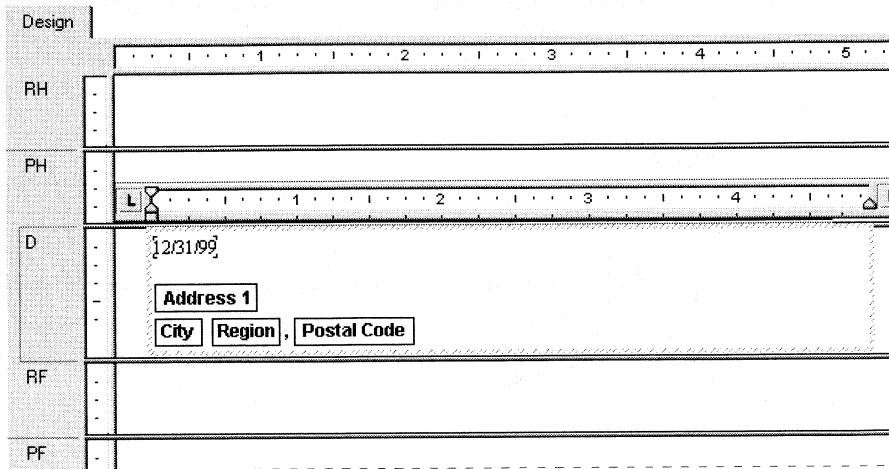
2. Press Enter twice to put some white space between the date and the inside address and move the insertion point down within the text object.

The image shows a report design grid with several sections: Design, RH, PH, D, RF, and PF. The 'D' section contains the date '12/31/99'. Below the date, there is an insertion point (a small vertical line) and a ruler indicating the position. The grid is bounded by a dashed line on the right and bottom.

Inside Address

To create the inside address, you will drag database fields into the text object from the Customer table in the Insert Fields dialog box.

1. Drag in the Address 1 field, place it at the insertion point, and press Enter to place the field and move the insertion point.
2. Drag in the City field and place it at the insertion point.
3. Type in a space to move the insertion point.
4. Drag in the Region field and place it at the insertion point.
5. Type in a comma and a space.
6. Finally, drag in the Postal Code field, place it at the insertion point, and press Enter twice to move the insertion point to the position where you want to start the salutation. This completes the inside address.



Salutation

1. Type in the word "Dear" and a space. (Do not include the quotation marks.)
2. From the Insert Fields dialog box, select the Contact Title field from the Customer table and drag it into the text object, placing it immediately after the space. The insertion point repositions itself immediately after the Contact Title field.
3. Type in a space. The program positions the insertion point immediately after the space.
4. Again from the Insert Fields dialog box, drag the Contact Last Name field into the text object and place it at the insertion point. The insertion point moves to the right of the field.
5. Type a colon (:) at the insertion point (do not include the parentheses) and press Enter to put in a carriage return and move the insertion point to the next line.

Letter Body

1. Now type in "Your company" (without the quotes) and type in a comma and a space after it.
2. Drag the Company Name field into the text object and place it at the insertion point, just after the space.
3. Type in a comma and a space.
4. Type in the following text (without the quotes): "helped make 1995 an outstanding year for CRAZE Mountain Bikes. I want to thank you and your staff for your support. I hope 1996 will be a banner year for you."
5. Press Enter twice.
6. Type in "Sincerely yours" (without the quotes) and a comma, and then press Enter four times.
7. Finally, to complete the form letter, type in your name.

The Design Tab should look similar to this.

Design | Preview

1 2 3 4 5

1 2 3 4

D 12/31/99

Address 1

City, **Region**, **Postal Code**

Dear **Contact Title** **Contact Last Name** :

Your company, **Customer Name**, helped make 1995 an outstanding year for Craze Mountain Bikes. I want to thank you and your staff for your support. I hope 1996 will be a banner year for you.

Sincerely yours,

John Manager

and when you preview the form letter it should look similar to this:

Design | Preview

D 6/1/96

7464 St. Georges Way
Sterling Heights MI, 48358

Dear Mr. Campbell:

Your company, Bike-O-Rama Corporation, helped make 1995 an outstanding year for Craze Mountain Bikes. I want to thank you and your staff for your support. I hope 1996 will be a banner year for you.

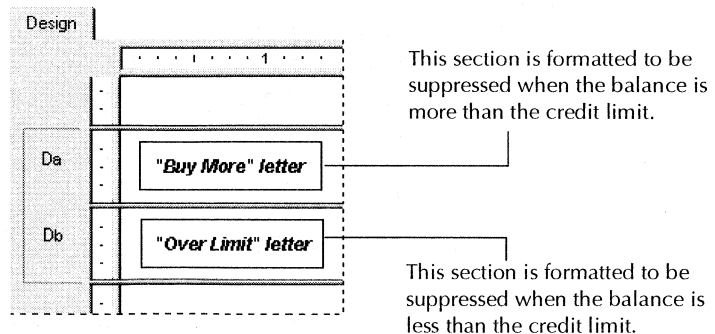
Sincerely yours,

John Manager

How to print conditional messages in form letters

Many times you may want to print conditional messages in form letters. For example, you may want to encourage customers with available credit to buy more and those who are over their credit limit to bring their accounts down below the limit once again. You can do this in the following way:

1. Insert a second Details section in your report using the Section Expert.
2. Create two different form letter text objects using the TEXT OBJECT command on the Insert menu. One letter should have the *buy more* message and the other should have the *over limit* message.
3. Place the *buy more* text object in Details A section and the *over limit* text object in Details B section.
4. Using the Section Expert, format each of the sections so they are suppressed only when certain conditions are met. In this example you could format the *buy more* section so it is suppressed when the balance is more than the credit limit and the *over limit* section so it is suppressed when the balance is less than the credit limit.



Now, when a record indicates available credit, the *buy more* letter will print. When the account is over limit the *over limit* letter will print. And when the customer is right at his credit limit, nothing will print at all.

How to format objects conditionally

You may want to create a report that uses different formats based on field values. For example, you may want to print an international report that prints date and currency values for each country in the format that is common in that country. You can do that using multiple sections. To create this kind of report you:

1. Create your report. See Chapter 5, *Tutorial*, Page 119, and Chapter 6, *Reporting 101*, Page 147.
2. Create one Details section for each country that requires a unique format. See *How to add, delete, move and merge sections*, Page 83.
3. Make certain that each of the Details sections contains the same data.
4. In the first section, right-click the currency field, choose **FORMAT FIELD** from the shortcut menu, and, when the Format Editor appears, set the currency values you want to use for the first country. Do the same thing with the date field. See Chapter 8, *Formatting*, Page 183.
5. Using the Section Expert, click the *Formula* button to the right of the *Suppress property* and create a formula that specifies the conditions under which the section should be suppressed. For example, if you set the date and currency values for the UK, you would create a formula that specified that the country value is not equal to UK. In other words, suppress the section for every record where the country value is not UK. This would make the section print only when the record contained a U.K. value.
6. Repeat steps three and four for each additional section.

Now when you print your report, the date and currency data for each country will appear in the format that is expected for that country. See *Conditional Formatting*, Page 188.

How to alternate background colors for rows

Another typical use of this would be to vary the background to see alternating lines in the Details section of your report to improve readability (a greenbar-paper effect). To do this, create a greenbar paper report:

1. Using the Section Expert, insert a second Details section. You should now have Details A and Details B sections. See *How to add, delete, move and merge sections*, Page 83.
2. While in the Expert, highlight Details A in the *Sections* list, click the Color Tab, and set the background color to White. See Chapter 8, *Formatting*, Page 183.
3. Click the Common Tab, click the *Formula* button to the right of the *Suppress (No Drilldown)* option, and type this formula in the Formula Editor when it appears.

```
Remainder (RecordNumber, 2) <> 0
```

This formula divides the Record number by 2 and if the remainder is something other than zero (which will happen for every odd numbered record), it tells the program to suppress the section.

4. Now highlight Details B and set the background color to Green.
5. Using the technique from Step 3, set the *Suppress* property conditionally for this section using the following formula:

```
RecordNumber = 0 or  
Remainder (RecordNumber, 2) = 0
```

6. Create your report and make certain that the information and layout of each of the Details sections is identical. In other words, whatever you put in Details A, put it in Details B as well.

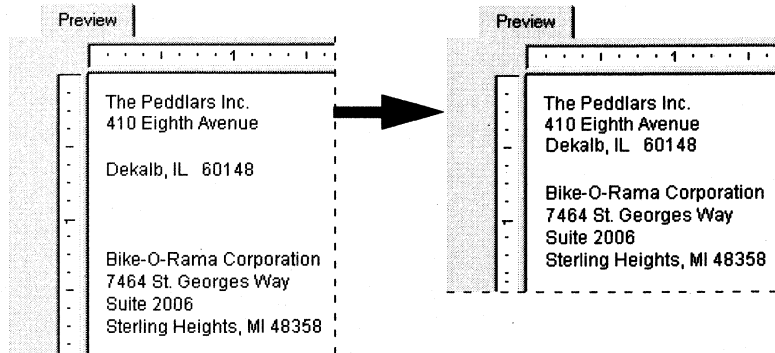
Now when you run the report, the program will print the first line and every even numbered line with a white background and every odd numbered line with a green background.

How to eliminate blank lines

It is very common to have two address lines in a customer table, one (Address 1) for street address and one (Address 2) that can be used for suite number or mail stop. Address 1 usually contains a value but Address 2 is often blank. If you create a customer list using this data and stack the fields on top of one another in mailing label format, those customer records with an empty Address 2 field will print with a blank line. You can eliminate this blank line using multiple sections. To do this:

1. Create two new Details sections so you have a total of three. See *How to add, delete, move and merge sections*, Page 83.
2. Place the Address 2 field in the middle section and the other data above and below it as you want it to appear in your report.
3. Format the middle section to *Suppress Blank Section*. See *How to hide parts of the report*, Page 103.

Now, when the report prints, if the Address 2 section is blank, the program will not print it and you will not get unwanted blank lines in your report.



Before suppressing blank lines, the Address2 field prints a blank line if it is empty. . . .

After suppressing blank lines, the Address2 field does not print if it is empty.

How to add blank lines conditionally

If you want to print a blank line on your report under specific conditions, you can do that using multiple sections. For example, if you want to insert a blank line after every fifth record, you can create two Details sections. Put the report detail data in the top section and leave the second section empty. Now, format the second section to be suppressed unless the following condition is met:

```
Remainder (RecordNumber,5) <> 0
```

This formula divides each record number by 5. If the division produces a remainder, it suppresses the blank section. But if the division produces no remainder, a zero (which it will whenever another five records are printed), the program prints the second section, thus inserting a blank line.

If you want to insert a blank line under different conditions, you can modify your formula appropriately. See *Conditional Formatting*, Page 188.

8

Formatting

What you will find in this chapter...

Formatting Concepts 184

Absolute Formatting 185

Types Of Formatting Properties 186

Conditional Formatting 188

How to add color, shading, and borders 190

How to add/edit lines and boxes 191

How to change margins 194

How to add/delete white space between rows 195

How to set page orientation and paper size 197

How to create a footer that appears on all pages but the first 198

How to flag values that meet certain conditions 199

Formatting Concepts

In this chapter, you will learn about formatting your report. Formatting refers to those things that you can do to change the layout and design of your report, and the appearance of text, of objects, or of entire report sections.

You use formatting to do many things:

- separate sections of your report,
- call attention to certain data,
- change the presentation of dates, numbers, Boolean, currency values, and text (strings),
- hide unwanted sections, or
- give your report a professional appearance.

Crystal Reports gives you a wide range of formatting commands and properties that you can apply to various elements in your report.

- You use formatting commands by choosing or setting options in dialog boxes that are specific to the kind of formatting you are doing. For example, if you want to change page margins, you use a dialog box that lets you specify the top, bottom, right and left margins.
- You set formatting properties in either the Format Editor (for objects) or the Section Expert (for sections) by toggling check boxes on and off or by supplying attribute values.
 - In most cases, you can set the properties in one of two ways:
 - absolute (always apply the property), or
 - conditional (apply the property only if certain criteria are met).

You can use both kinds of formatting properties wherever you need them in your report. See *Absolute Formatting*, Page 185, and *Conditional Formatting*, Page 188.

In the pages that follow, you will learn about the kinds of formatting you can do with Crystal Reports, and you will get step-by-step instructions for performing a variety of formatting tasks.

Properties

Formatting using menu commands works in several different ways. No general rule applies to all commands so you will learn about formatting with commands in the Hands-On section later in this chapter. But formatting by setting object or section properties always follows the same set of principles, and we will discuss those principles here before going on to the Hands-On section.

There are two kinds of object or section formatting you can do with Crystal Reports: absolute and conditional.

Absolute Formatting

Absolute formatting is formatting that applies under any condition.

SELECT, THEN APPLY

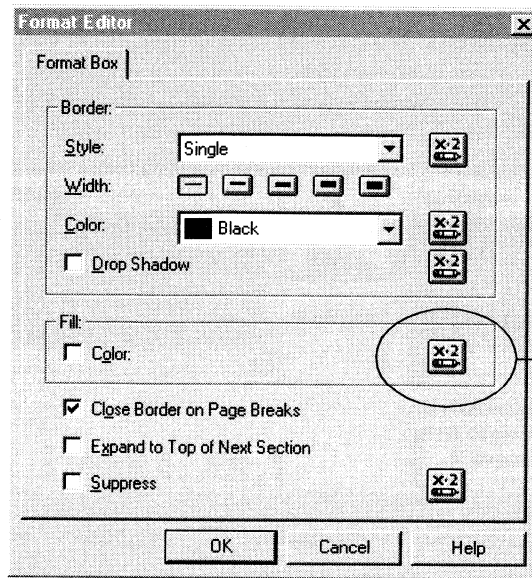
Absolute formatting always follows a *select then apply* procedure:

- you *select* what it is that you want to format (an object or a section), and then,
- you *apply* the formatting to your selection using property settings.

You use the following dialog boxes for formatting your reports:

- Format Editor for formatting field values,
- Format Section dialog box for formatting entire sections,
- Borders and Colors Tab of the Format Editor for formatting objects, numbers, and text,
- Format Box Tab of the Format Editor for formatting boxes, and
- Format Line Tab of the Format Editor for formatting lines.

Each of these dialog boxes contains a number of different formatting properties and the tools for turning the properties on or off, or for specifying attributes.



Make the desired selections and click Ok when finished.

Formula button.

Most properties also come with a formula button that enables you to set the condition(s) under which the property is to be applied. A discussion of conditional formatting follows later in this chapter.

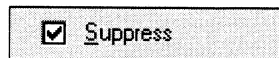
Types Of Formatting Properties

Formatting properties fall into two general categories:

- On or Off Properties
- Attribute Properties

ON OR OFF PROPERTIES

An on or off property is a property that is set using a check box.



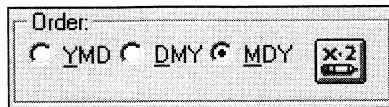
When you toggle the check box on, the property is always applied. When you toggle the check box off, the property is never applied. The *Suppress* property is one such on or off property; an object or section is either suppressed (on) or it is not (off).

ATTRIBUTE PROPERTIES

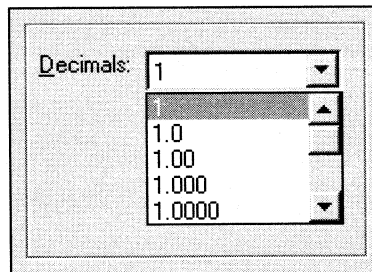
An attribute property is a property that specifies one of many alternative attributes. The *Color* property, for example, gives you the opportunity to specify one of the 16 basic Windows colors or to specify a custom color and then create that color using the tools provided. Clearly such a property cannot be handled with a simple on or off switch.

You set an attribute property in one of several ways, depending on the property. You can set it:

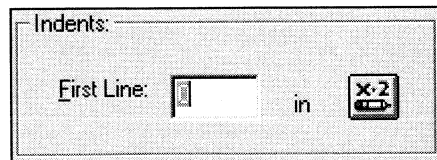
- by choosing an option box,



- by choosing an attribute from a drop down list, or



- by typing an attribute in a text box.



Conditional Formatting

Conditional formatting is formatting that applies only under certain conditions. For example, you may only want:

- customer balances printed in red if they are past due,
- the dates to appear in Day, Month, Year format if the customer is Canadian, or
- background color to appear on every other line.

Crystal Reports makes it easy for you to apply formatting conditionally in these and hundreds of other situations. See *How to format objects conditionally*, Page 178.

With standard formatting you followed the *select* then *apply* procedure. For conditional formatting you follow the same general procedure, but you go a step further and set up conditions that determine whether or not the formatting will be applied. You specify those conditions using simple formulas.

NOTE: When you set up a conditional formatting formula, the formula overrides any fixed setting you have made in the dialog box. For example, if you set the Suppress property toggled on and then you set up a conditional formula for the Suppress property, the property will still apply only if the condition in the formula is met.

The program enables you to set both on and off properties and attribute properties conditionally. Each of these requires a different kind of formula.

CONDITIONAL ON OR OFF PROPERTIES

A conditional *on or off* property tests to see if a condition has been met. It is *on* if the condition is met, *off* if the condition is not met. There is no middle ground.

Use Boolean formulas for this kind of formatting.

Value = condition

The program tests each value to see if it meets the condition and it returns a yes or no answer. It then applies the property to every value that returns a yes answer.

CONDITIONAL ATTRIBUTE PROPERTIES

A conditional *attribute* property tests to see *which* of two or more conditions was met. The program then applies the formatting appropriate to the condition. For example, assume that you want values under quota printed in red and all other values printed in black. The program tests to see whether the value is under quota or not. If it is under quota, it applies the *red* attribute, if it is not, it applies the *black* attribute.

Use an If-Then-Else formula for this kind of conditional formatting.

```
If Condition A Then
    Red
Else
    Black
```

NOTE: When you set up many kinds of conditional attribute properties, the program loads a selection of attributes into the Functions list in the Formula Editor. You can double-click any of those attributes to add them to your formula.

You can go further with this kind of property. You can specify a list of conditions and a property for each; you are not limited to two or three conditions. For example, if you have a number field on your report that contains sales figures from countries around the world, you can specify the number attribute(s) that you want to apply to each country. In this case, your conditions would specify that if it is from Country A, apply the Country A attribute; Country B, apply the Country B attribute; Country C, apply the Country C attribute, and so on.

With more than two alternatives, you use this kind of formula:

```
If Condition A Then
    Red Or
If Condition B Then
    Black or
If Condition C Then
    Green
Else
    Blue
```

See *How to create If-Then-Else formulas*, Page 270.

VALUES THAT DO NOT FIT ANY CONDITION

With conditional on or off properties, the value will either meet the condition or it will not. But conditional attribute properties are different. There may be some cases where the value does not fit any of the conditions. For example, if you want all sales figures that are 10% or more over quota formatted green and all sales figures that are 10% or less under quota formatted red, what is going to happen to values in between? The program takes care of that automatically. When a value does not fit any of the conditions in the formula, the program formats that value using the attribute setting in the dialog box. In this case, the dialog box setting is not overriding the formula setting; it is just providing an attribute when the formula does not provide it.

Several Hands-On examples follow showing how to format sections and objects conditionally. While the topics detail specific reporting needs, they illustrate basic procedures as well.

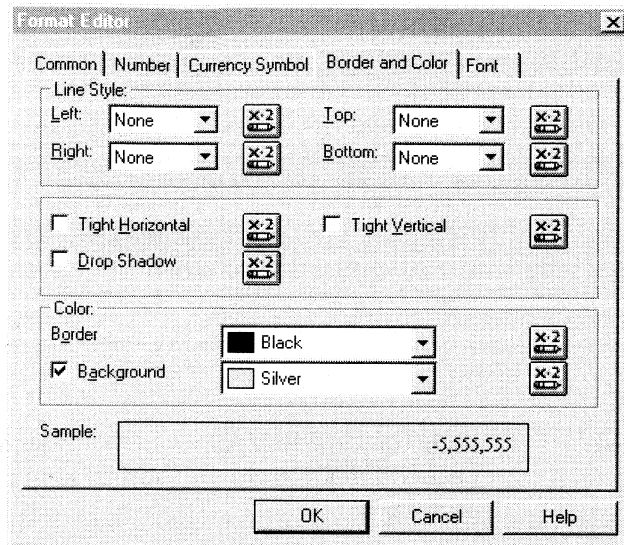
- the first example shows the procedure for setting an *on* or *off* formatting property conditionally, and
- the second shows how to set an *attribute* formatting property conditionally.

HANDS-ON (Absolute Formatting)

How to add color, shading, and borders

Crystal Reports allows you to add color, borders, and shading to fields on your report to emphasize important data and create professional looking reports.

Right-click the field you want to format and choose the CHANGE BORDER AND COLORS command from the shortcut menu that appears. The Format Editor appears. Click the Border and Color Tab to make it active.



Select the border type (line style), color, placement, and background color you want and click *OK* when finished. Crystal Reports formats the selected field to your specifications.

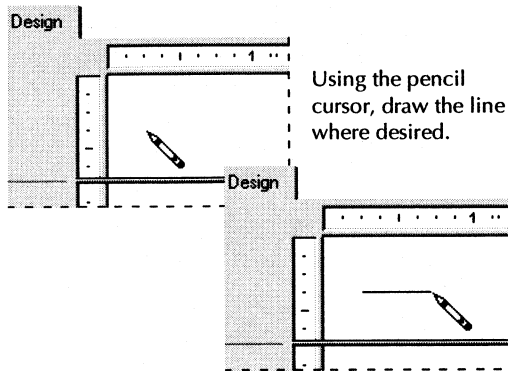
How to add/edit lines and boxes

Crystal Reports allows you to add lines and boxes to your report to emphasize important data and create professional looking reports.

How to add lines

Click the **INSERT LINE** button on the toolbar. A pencil-shaped cursor appears.

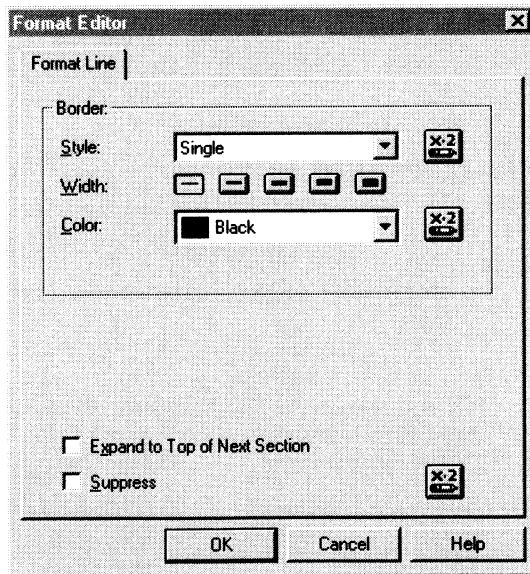




Using the pencil cursor, draw the line where desired.

How to edit lines

Right-click the line you want to format and choose the **FORMAT LINE** command from the shortcut menu. The **Format Editor** appears.

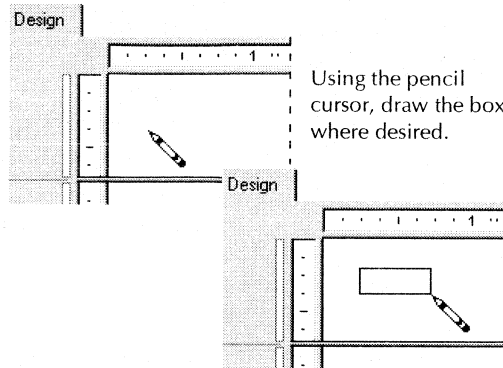


Make the desired changes and click **OK** when finished to return your report.

How to add boxes

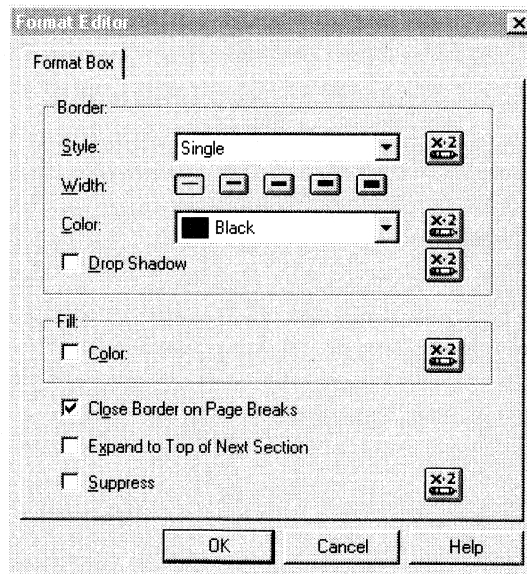


Click the INSERT BOX button on the toolbar. A pencil-shaped cursor appears.



How to edit boxes

Right-click the box you want to format and choose the FORMAT BOX command from the shortcut menu. The Format Editor appears.



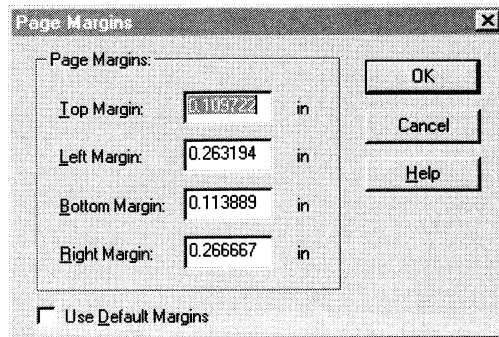
Select the color, width, style, and/or fill color you want and click OK when finished.

Crystal Reports formats the selected box to your specifications.

How to change margins

Using the Page Margins dialog box

1. Choose the PAGE MARGINS command from the File menu. The Page Margins dialog box appears.

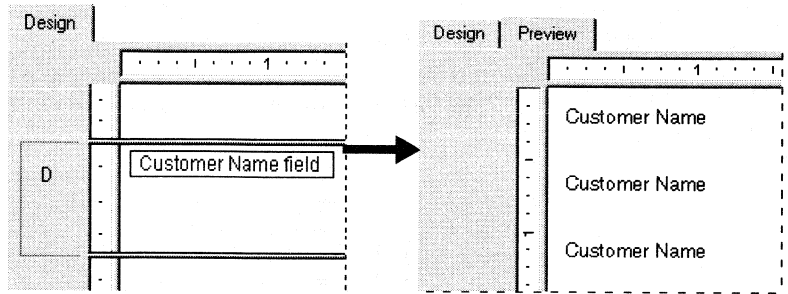


Change the default page margins to fit your needs and click Ok when finished.

- Crystal Reports uses a dashed line to display your margins in the Design Tab and Label Design Tab.
- The numeric margin settings appear in both the Mailing Labels dialog box and in the Printer Margins dialog box.
- All margins are calculated from the paper edge. Thus, a left margin of .25 inches causes the printing to start exactly one quarter inch in from the edge of the paper.

How to add/delete white space between rows

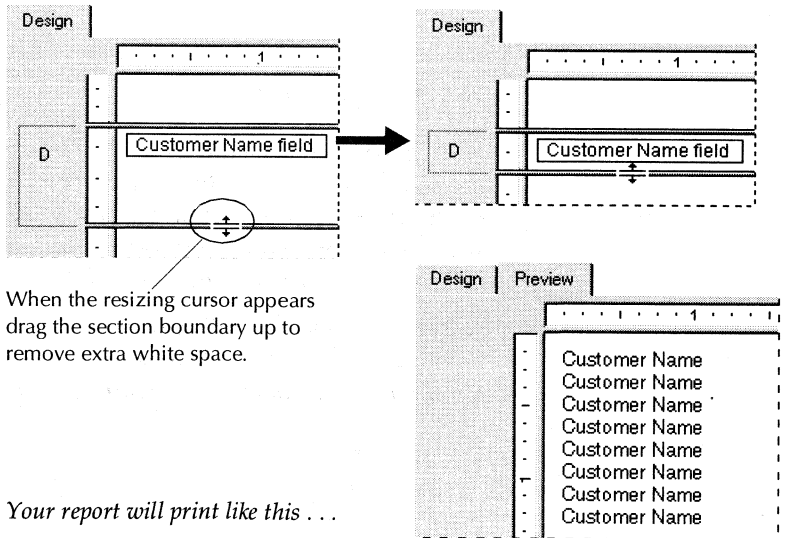
The height of a section in relation to the objects in it affects the amount of white space that appears between rows on your report



Crystal Reports' new freeform Design Tab allows you to add and delete white space two ways: by sizing the area in the Design Tab using the resizing cursor or by changing options using the Format Section dialog box.

How to delete white space by resizing

To delete unnecessary white space in a section, simply move the mouse pointer over the bottom section boundary line. The pointer will change to a resizing cursor.



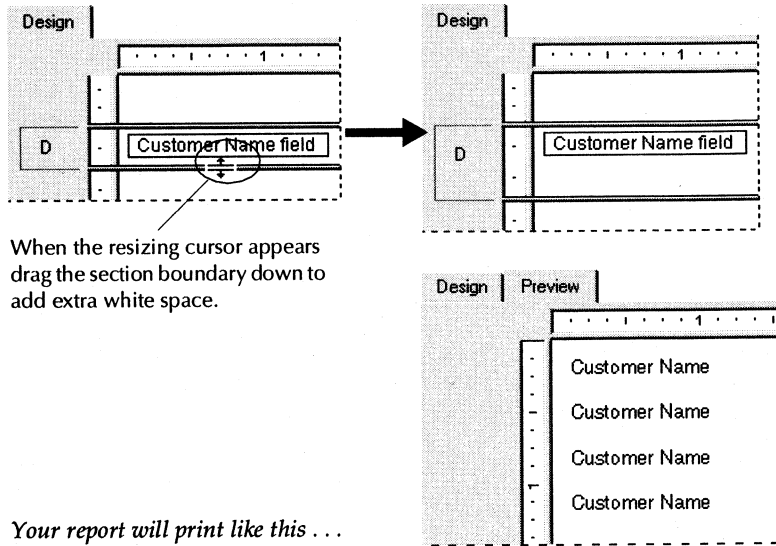
When the resizing cursor appears drag the section boundary up to remove extra white space.

Your report will print like this . . .

NOTE: You can also right-click the shaded area to the left of the section and choose *Fit Section* from the shortcut menu that appears. The program automatically resizes the section so the bottom boundary is even with the baseline of the bottom object in the section. See *How to eliminate blank lines*, Page 180.

How to add white space by resizing

To add extra white space between rows in your report, simply move the mouse pointer over the bottom section boundary line. The pointer will change to a resizing cursor.



NOTE: You can also add white space to a section by right-clicking the shaded area to the left of the section and choosing *Insert Line* from the shortcut menu that appears. The program resizes the section automatically, adding the amount of space necessary to hold a line of typical database fields.

NOTE: If an entire section is blank (i.e., if you are not putting anything into the Page Footer section of your report), you can eliminate the unnecessary white space by suppressing the section.

Related Topics

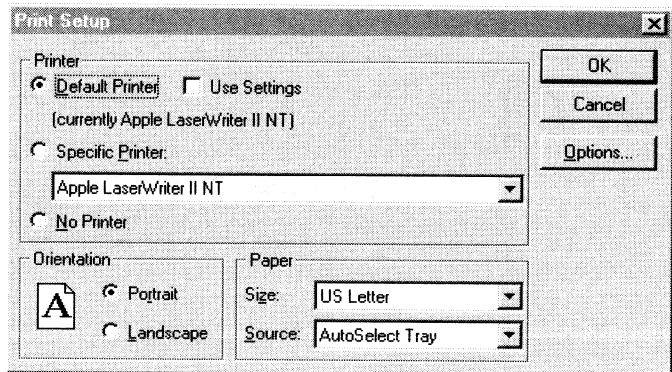
See *How to add, delete, move and merge sections*, Page 83.

See *How to add blank lines conditionally*, Page 181.

How to set page orientation and paper size

You can use Crystal Reports in portrait or landscape orientation and with a variety of paper sizes. You select these using the **PRINTER SETUP** command on the File menu. The **Printer Setup** dialog box will appear.

1. Choose the **PRINTER SETUP** command from the File menu. The **Printer Setup** dialog box will appear.



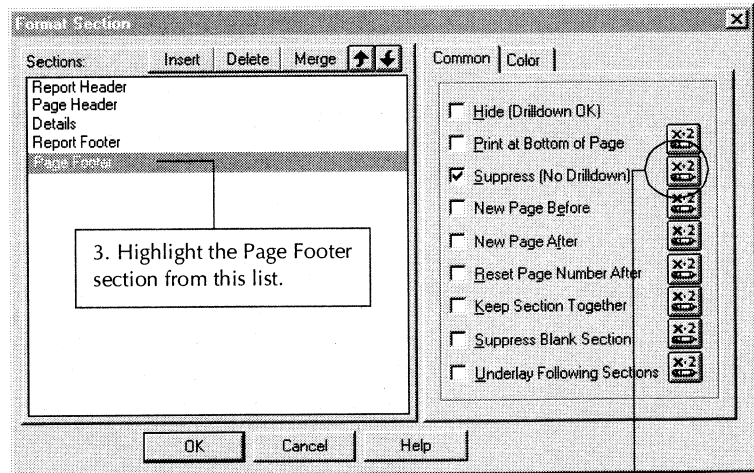
2. In the **Print Setup** dialog box, activate the printer you want to use if it is not already the active printer. Your paper size options are directly related to the printer you have selected. For example, the HP LaserJet driver (PCL) offers a choice of letter, legal, executive or A4 paper sizes, whereas the PostScript printer driver lets you choose from letter, legal, note, A4, B5, letter small, and A4 small paper sizes.
3. Select either **Portrait** or **Landscape** orientation by clicking the appropriate option button in the *Orientation* box.
4. Select the paper size desired and its source from the scroll lists in the *Paper* box.
5. Click **OK** when finished.

HANDS-ON (Conditional Formatting)

How to create a footer that appears on all pages but the first

You can have Crystal Reports print a page footer on all pages except the first page. You do this by formatting the page footer section conditionally using an *on* or *off* property.

1. Place the field you want to display as a page footer in the Page Footer section of your report.
2. Click the SECTION EXPERT button. The Section Expert dialog box appears.



4. Click the Formula button to the right of the Suppress button to open the Formula Editor.

5. Enter the following formula:

```
PageNumber = 1
```

This formula suppresses the section on the first page but not on any of the other pages.

6. Click the *Accept* button. If there is an error in your formula, Crystal Reports will display a Formula Compiler Error detailing your error. Search for *Formula Compiler Errors* in Crystal Reports online Help.

7. Preview your report to ensure that the page footer appears on all pages but the first.

NOTE: *If you have a multi-line page footer and you have put the lines into separate Page Footer sections, you will need to suppress each of those sections conditionally using the formula above.*

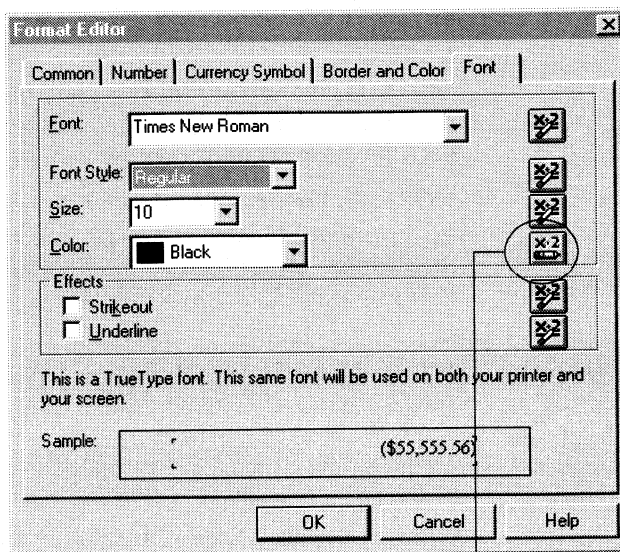
NOTE: *To create a page header that appears on all pages but the first, put your header information in the Page Header section and then suppress that section conditionally using the same formula that was used for suppressing the Page Footer section.*

How to flag values that meet certain conditions

You can flag field values that meet certain conditions by formatting the field conditionally using an attribute property.

For the purpose of this example, assume you have a Last Year's Sales field in your report and you want the value to print in Red if sales were less than \$10,000 and to print in Black in all other cases. In this way you're using color to flag values that fall below a certain threshold.

1. Right-click the Last Year's Sales field and choose the FORMAT FIELD command from the shortcut menu. The Format Editor appears. Click the Font Tab to activate it.



2. Click the Formula button to the right of the Color drop down box to open the Formula Editor.

3. Enter the following formula:

```
If {customer.LAST YEAR'S SALES} < 10000 Then
    Red
Else
    Black
```

See *How to create If-Then-Else formulas*, Page 270.

4. Click the *Accept* button to return to the Format Editor. Notice that the *Formula* button has changed. The text on the button is now red instead of blue. This indicates that a formula has been entered for that property. Click *OK* to return to your report.

Now, when you run your report, if the value in the Last Year's Sales field is less than \$10,000, the value will print in red.

9

Sorting, Grouping, and Totaling

What you will find in this chapter...

Sorting, grouping, and summarizing overview 202

Creating Custom Groups 207

How to do a single field sort 210

How to do a multiple field sort 213

How to group data 215

How to sort the records within groups 217

How to summarize grouped data 219

How to subtotal grouped data 221

How to sort based on summarized group values 223

How to extend prices and then subtotal the extensions 223

How to create multiple levels of subtotals 227

How to group based on first letter of company name 231

How to group data in intervals 235

How to calculate a percentage of the grand total 238

How to create Group Headers 241

Sorting, grouping, and summarizing overview

Sorting, grouping, and summarizing are the steps that turn disorganized data into useful information on a report. This chapter introduces you to the concepts behind sorting, grouping, and summarizing, and it gives you many hands-on tutorials for most of the things you want to be doing in these areas.

Sorting

Sorting means putting values in some kind of order to help you find and/or evaluate them. We sort the information in a phone book, for example, because it would have little usefulness if it was not sorted. Trying to find someone's phone number in an unsorted book would entail a random search through tens of thousands of names, a true *needle-in-a-haystack* experience. With sorting, however, we can find the number we need in a hurry. For example, if we are looking for the phone number of a John J. Smith, we use three different levels of sorting to find it.

- The data is sorted by last name so we know the name Smith is in the S section of the book and we turn there immediately.
- When we find the Smiths, (and there are a lot of them) we see that they are sorted by first name so that John Smith comes after Bob Smith. We turn to the John Smiths.
- Finally, when we find the John Smiths (and here again, there are several of them), we see that they are sorted by middle initial so the John J. Smiths come after the John D. Smiths. We turn to John J. Smith, find his phone number, and place the call.

Because of the sorting we can find anybody's listed phone number in seconds.

Sorting by record

When you sort in Crystal Reports, the program asks you to select two things:

- the field you want your sort to be based on (sort field), and
- the sort direction.

SORT FIELD

A sort field is a field that determines the order in which data appears on your report. Crystal Reports sorts field data using Windows' sort comparison algorithms, and it uses rules specific to the Country you select in the International section of the Windows Control Panel (Windows 3.x, Windows NT) or the Regional section (Windows 95).

You can use any field as a sort field. A field's data type determines the method in which the data from that field is sorted:

- String fields are sorted in the following manner:
 - Single character values are sorted so that blanks have the lowest value, then punctuation, then numbers, then uppercase letters, and finally lowercase letters.
 - Then two character values are sorted, then three, etc., using the same rules. As a result:
 - "BOB" comes before "bob",
 - "123" comes before "124",
 - " " (blank) comes before "a", and
 - "aa" comes before "aaa"
- Currency fields are sorted in numeric order.
- Number values (120, or 5555) are sorted in numeric order.
- Date fields are sorted in chronological order.
- `dateTime` fields are sorted in chronological order by date and same-date values are then sorted by time.
- Time fields are sorted in chronological order.
- Boolean comparison fields are sorted so that false values (0) come first, then true values (1).

When you select a sort field, Crystal Reports sorts the values from that field.

SORT DIRECTION

Direction refers to the order in which the values are printed, once sorted.

- Ascending order means smallest to largest, 1 to 9, A to Z, False to True.
- Descending order means largest to smallest, 9 to 1, Z to A, True to False.

Single field sorts

Single field sorts are sorts in which all the records used in the report are sorted based on the values in a single field. Sorting an inventory report by stock number and sorting a customer list by customer number are examples of single field sorts. See column 1 in the chart on Page 205.

Multiple field sorts

In multiple field sorts, Crystal Reports first sorts the records based on the values in the first field selected, putting them in ascending or descending order as specified. When two or more records have the same field value in the first sort field, it then sorts those records (and those alone) based on the value in the second sort field. For example, in a sort on last name and then first name (in ascending order), Smith, Bob would be returned before Smith, John no matter which way the fields are listed in the database.

It follows a similar process for three field sorts, four field sorts, and so on. See column 1 in the chart on Page 205.

Sorting and grouping together

Crystal Reports has the most powerful sorting and grouping capabilities of any Windows based report writer. When you select a grouping option, the program automatically sorts the data as part of the operation. You pick a field on which to base the sort and you pick a sort direction and the program does the rest. See column 1 in the chart on Page 205.

When you group data you can select one of four sort orders:

- **Ascending**
The program sorts the records in ascending order based on the values in the sort and group by field you select, and then it begins a new group whenever the value changes (from Adams to Brown, for example).
- **Descending**
The program sorts the records in descending order based on the values in the sort and group by field you select, and

then it begins a new group whenever the value changes (from Brown to Adams, for example).

- **Original**

The program leaves the records in the order in which they appear in their originating database table, and it begins a new group whenever the value changes in the sort and group by field you select (from CA to AZ, for example, and again from AZ to CA).

- **Specified order**

The program puts each record in the custom group you specify, and it leaves the records in each group in original order or it sorts them in ascending or descending order, depending on your instructions. For additional information on specified order grouping, see *Creating Custom Groups*, Page 207.

Results using different sorting and grouping operations

We have discussed the way the program handles several sorting and grouping operations. The following chart shows the way data would appear after being manipulated using these operations.

1	2	3	4	5	6	7	8	9
CO	AZ	WA	AZ	WA	CO	WA	CA	WA
WA	CA	WA		WA		CA	CA	WA
CA	CA	WA	CA	WA	WA	CA	CA	WA
CA	CA	CO	CA			CA	CA	CA
CA	CA	CO	CA	CO	CA	WA	CA	CA
AZ	CA	CA	CA	CO	CA	WA	WA	CA
WA	CO	CA	CA		CA	CA	WA	CA
WA	CO	CA		CA		CA	WA	CA
CA	WA	CA	CO	CA	AZ			
CA	WA	CA	CO	CA		CO	AZ	CO
CO	WA	AZ		CA	WA	AZ	CO	CO
			WA	CA	WA	CO	CO	AZ
			WA					
			WA	AZ	CA			
					CA			
					CO			

- **Column 1**
The data as it appears in the database table.
- **Column 2**
The data from Column 1 sorted in ascending order (A to Z, 1 to 9). No grouping.
- **Column 3**
The data from Column 1 sorted in descending order (Z to A, 9 to 1). No grouping.
- **Column 4**
The data grouped in ascending order. The program automatically sorts the data in ascending order and then inserts a group break whenever the value changes.
- **Column 5**
The data grouped in descending order. The program automatically sorts the data in descending order and then inserts a group break whenever the value changes.
- **Column 6**
The data grouped in original order. The data is not sorted before it is grouped. The program inserts a group break whenever the value changes. Note that similar values may appear in more than one group.
- **Column 7**
The data grouped in specified order. This is one of thousands of possible custom groupings. In this example, the first group consists of Pacific states and the second group consists of Mountain states. The records in each group are sorted in original order.
- **Column 8**
The same specified order grouping as Column 7, but the records in each group are sorted in ascending order.
- **Column 9**
The same specified order grouping as column 7, but the records in each group are sorted in descending order.

As you can see, your sorting and grouping choices can have a major impact on the way data appears on your report.

Creating Custom Groups

Most of the time you sort and group your data based on the values in some field in your report. For example, if you have a customer list and you want to sort and group it by state, the program first sorts the list by state and then breaks the list into state groups whenever the value in the State field changes.

Sometimes, however, you may not want to group based on the values found in one of the fields on your report.

- Your report may not contain the field you want to group on. For example, your report contains a City field and a State field but no County field, but you want to group by county.
- Your report may contain the field you want to group on, but you are not happy with the grouping based on the values in that field. For example, you have a Color field on your report that includes specific color names (Logan Green, Sky Blue, Emerald Green, Navy Blue, etc.) but you want all "flavors" of each color to appear as a single group (Greens, Blues, Reds, etc.). In this case you can build custom groups and manually assign the records you want to be in each group.
- Your report may contain the field you want to group on, but you want to select specific values or ranges of values for each group. For example, you might want one group to contain records where gross sales are less than a certain value, a second group where gross sales are greater than a certain value, and a final group where gross sales fall between two values. In this case, you can build your groups using the same range selection facilities that are available to you for building record selection queries.

Crystal Reports provides Specified Order grouping as a solution to these custom sorting and grouping challenges. Specified order grouping enables you to create the groups you want to appear on your report and the records that each group contains. Your only real limitation is that a record can be assigned to only one group.

To use Specified Order grouping, you select *in specified order* as your sort option whenever the program provides you with that option. The program gives you the *in specified order* option whenever you create groups using the GROUP, SUBTOTAL, or SUMMARY commands on the Insert menu, whenever you create groups while building a report using one of the Report Creation experts, or whenever you choose the CHANGE GROUP EXPERT command on the Report menu.

SUMMARIZING GROUP VALUES

One of the primary reasons you might break your data into groups is so you can run some calculations on each group of records instead of on all the records in the report. When you do this, the program evaluates all of the values in each group and then summarizes them.

- For a customer list, you might want to determine the number of customers in each state. For such a report, your summary would consist of counting the distinct customers in each state group.
- For an order report, you might want to determine the average order placed each month. For this, your summary would calculate the size of the average order for each month group.
- For a sales report, you might want to determine the total sales per sales representative. For this, your summary would sum or subtotal the order amounts for each sales representative group.

As you can see, you can summarize your data in a variety of ways to make useful reports.

When Crystal Reports summarizes data it sorts the data, breaks it into groups, and then summarizes the values in each group. It does this all automatically; all you have to do is specify:

- the field you want summarized,
- the type of summary operation to be performed on the field,
- the field that's going to trigger a new group whenever its value changes, and
- the sort order.

The program handles all the details.

The program includes a number of summarizing options. Depending on the data type of the field you are planning to summarize, you can:

- sum the values in each group,
- count all the values or only those values that are distinct from one another,
- determine the maximum, minimum, or average value, and
- calculate two kinds of standard deviations and variances.

You can set up all of these summaries using the `SUMMARY` command on the Insert menu. If you want a shortcut for summing the values, you can use the `SUBTOTAL` command on the Insert menu as well. See *Summary functions*, Page 598-599.

SORTING SUMMARIZED GROUP VALUES

You can sort summarized group values in either ascending or descending order. In an orders report, for example, if you subtotal orders by state, you could have

- the group with the lowest subtotal first, then the next lowest, and so forth (ascending), or
- the group with the highest subtotal first, then the next highest, and so forth (descending).

You can sort your report based on group values using the TopN/Sort Group Expert on the Report menu.

HANDS-ON

NOTE: To sort groups that are not summarized, chose the Change Group Expert command on the Report menu.

When you insert a database field into your report, the data within the fields appears in the order it was originally entered into the database.

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

Customer Name	Region	Postal Code	Country
Bike-O-Rama	MI	48358	USA
The Peddlers	IL	60148	USA
Bikes R Us	OH	43001	USA
CycleSporin	AL	35818	USA
Sporting Wheels	CA	92150	USA
Pedal Pusher Bikes	BC	V3C 1G2	Canada
SAB Mountain		3012	Switzerland
ABC Ltd.	BC	V6E 3T2	Canada
XYZ Industries	WI	53730	USA
The Bike Shop Co.	ID	83650	USA
The Cyclists Co.	BC	V6G 4V3	Canada
La Bomba de Bicicleta		28001	Spain

Often it is easier to review or find information when you can see it sorted in a logical format. For instance, you may want to have a customer list sorted alphabetically by name or by country. That is known as a single field sort.

NOTE: To see an example of the above report, open the *SGT01.RPT* file in the *\CRW* directory.

How to do a single field sort

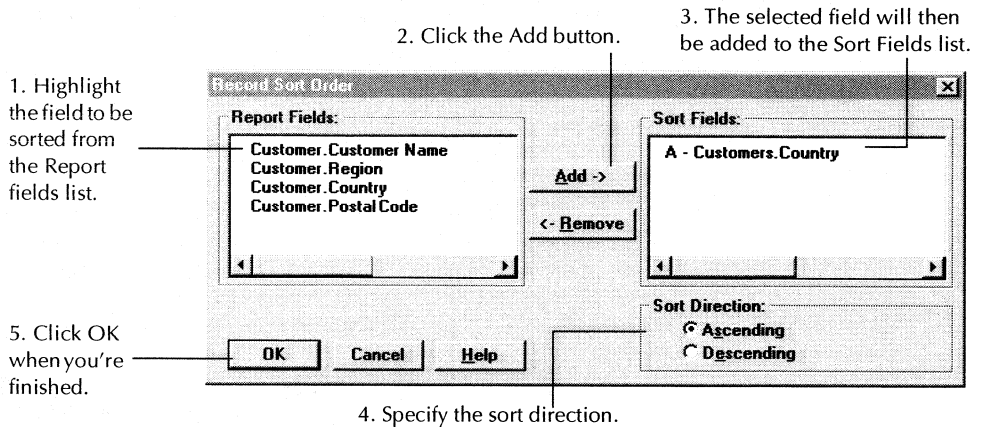
To get started, create a report using the customer table in *CRAZE.MDB*, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}
{customer.REGION}
{customer.POSTAL CODE}
{customer.COUNTRY}
```



Click the **Sort Order** button on the toolbar. The Record Sort Order dialog box appears.

NOTE: The following screenshot illustrates both a before and after state of the dialog box. Typically, any fields you move to the Sort Fields list box will no longer appear in the Report Fields list box.



Records are sorted based on the values in the sort field.

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

The Regions within each Country remain unsorted.

Customer Name	Region	Postal Code	Country
BG Mountain Inc.	BC	V3F 2K1	Canada
Montreal Mountain	Quebec	H2Z 1S4	Canada
Hansen MTB Inc.	Quebec	H1J 1C3	Canada
Deely MTB Inc.	BC	V6Z 2H2	Canada
Bicicleta Conexion		41101	Spain
Amadablam		28001	Spain
Sierra Bicycle Group	CA	92549	USA
Mountain Toad	CA	75369	USA
Desert Mountain	NV	89117	USA
Active Outdoors	IL	56478	USA
Sierra Mountain	NV	86521	USA
SFB Inc.	CA	94117	USA
Mountain Tops Inc.	IL	54321	USA
		92007	USA

Countries are now sorted in alphabetical order.

Notice that the records are displayed in alphabetical order by country: all of Canada, then Spain, then the United States. There is no further sorting of these records, however. If you want your customers sorted by Country and then, for example, by Region, you need to do a two field sort.

NOTE: To see an example of the above report, open the *SGT02.RPT* file in the *\CRW* directory.

Related Topics

See Chapter 6, *Reporting 101*

How to do a multiple field sort, Page 213

How to sort the records within groups, Page 217

How to do a multiple field sort

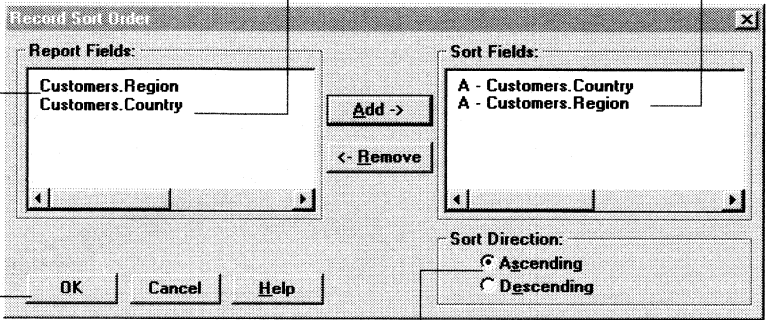


Click the SORT ORDER button on the toolbar. The Record Sort Order dialog box appears.

1. Highlight the field you want the data sorted by first and click the Add button to add it to the Sort Fields list.

2. Highlight the field you want the data to be sorted second and add it to the Sort Fields list.

3. The selected fields are displayed in the order they are sorted.



The dialog box titled "Record Sort Order" has two main sections: "Report Fields:" and "Sort Fields:". The "Report Fields:" list contains "Customers.Region" and "Customers.Country". The "Sort Fields:" list contains "A - Customers.Country" and "A - Customers.Region". Between these lists are "Add >" and "< Remove" buttons. Below the lists are "OK", "Cancel", and "Help" buttons. A "Sort Direction:" section at the bottom has radio buttons for "Ascending" (selected) and "Descending".

4. As you add each field to the Sort Fields list, specify the sort direction.

5. Click OK when you are finished.

Records are sorted by country first and then by region.

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

	Customer Name	Region	Postal Code	Country
The Regions within each Country are sorted in alphabetical order.	BG Mountain Inc.	BC	V3F 2K1	Canada
	Deely MTB Inc.	BC	V6Z 2H2	Canada
	Montreal Mountain	Quebec	H2Z 1S4	Canada
	Hansen MTB Inc.	Quebec	H1J 1C3	Canada
Postal Codes within each Region are unsorted.	Bicicleta Conexion		41101	Spain
	Amadablam		28001	Spain
	Sierra Bicycle Group	CA	92549	USA
	Mountain Toad	CA	75369	USA
	SFB Inc.	CA	94117	USA
	Active Outdoors	IL	56478	USA
	Mountain Tops Inc.	IL	54321	USA
	Sierra Mountain	NV	86521	USA
	Desert Mountain	NV	89117	USA
			39227	USA

Countries are sorted in alphabetical order.

Notice that the data remains in order by country, but when a country has customers in different regions, the regions are now in alphabetical order as well.

NOTE: To see an example of the above report, open the *SGT03.RPT* file in the *\CRW* directory.

Related Topics

See Chapter 6, *Reporting 101*

See *How to do a single field sort*, Page 210

See *How to sort the records within groups*, Page 217

How to group data

Sometimes sorting is not enough. You may want to break data into meaningful groups. Crystal Reports allows you to group data in one easy step.

NOTE: It is unnecessary to sort the data manually before you group it. Crystal Reports does the necessary sorting automatically, as part of the grouping procedure. If you just want your data sorted, follow the sorting procedures outlined above. If you want your data grouped, follow the grouping procedures outlined below.

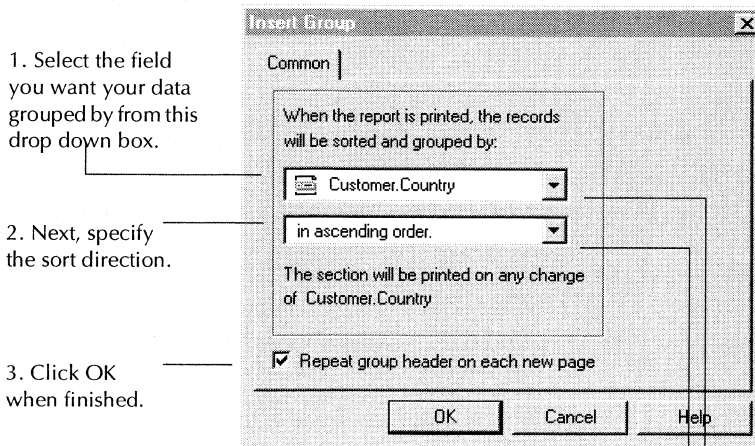
To get started, create a report using the sample data, CRAZE.MDB, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}  
{customer.REGION}  
{customer.COUNTRY}  
{customer.POSTAL CODE}
```

You will use this same data in the next example as well.

To do basic grouping, choose the GROUP command from the Insert menu. The Insert Group dialog box appears.

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.



Customer Name	Region	Postal Code	Country
BG Mountain Inc.	BC	V3F 2K1	Canada
Montreal Mountain	Quebec	H2Z 1S4	Canada
Hansen MTB Inc.	Quebec	H1J 1C3	Canada
Deely MTB Inc.	BC	V6Z 2H2	Canada
Bicicleta Conexion		41101	Spain
Amadablam		28001	Spain
Sierra Bicycle Group	CA	92549	USA
Mountain Toad	CA	75369	USA
Desert Mountain	NV	89117	USA
Active Outdoors	IL	56478	USA
Sierra Mountain	NV	86521	USA
		94117	USA
		54321	USA

The Customers are grouped by Country.

The groups are also sorted in ascending (A-Z) order.

But, the records within each group are unsorted.

Notice that the values are grouped by Country, and the country groups appear in ascending (A to Z) order. Notice too, however, that the records within each group remain unsorted. To remedy this, you need to sort the records within each group as described previously. See *How to sort records within a group*, Page 217.

NOTE: To see an example of the above report, open the *SGT04.RPT* file in the \CRW directory.

Related Topics

See Chapter 6, *Reporting* 101

How to sort the records within groups, Page 217.

How to group based on the first letter of company name, Page 231.

How to group data in intervals, Page 235.

How to create Group Headers, Page 241.

How to sort the records within groups



Once you have your data grouped, click the SORT ORDER button on the toolbar. The Record Sort Order dialog box appears.

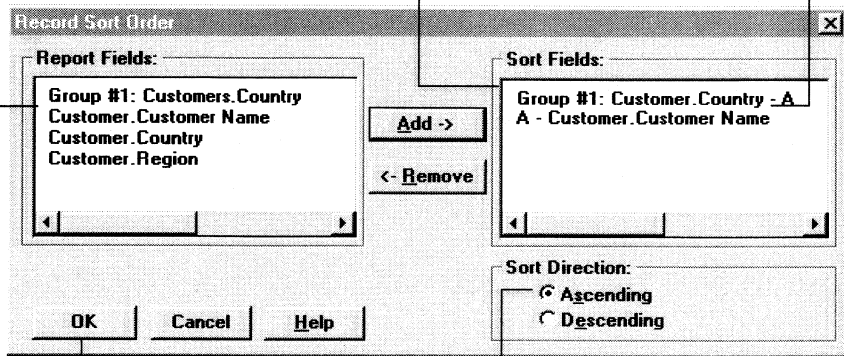
NOTE: The following screenshot illustrates both a before and after state of the dialog box. Typically, any fields you move to the Sort Fields list box will no longer appear in the Report Fields list box.

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

1. Highlight the field you want the records in the groups sorted by and click the Add button to add it to the Sort Fields list.

NOTE: This field specifies the sort that was done automatically when you grouped the data. It cannot be moved in this dialog box.

2. The selected fields are displayed in the order they are sorted.



4. Click OK when finished.

3. When you add the field to the Sort Fields list, specify the sort direction.

The customers within each country are also sorted.

Customer Name	Region	Postal Code	Country
BG Mountain Inc.	PQ	H1J 1C3	Canada
Deely MTB Inc.	BC	V6Z 2H2	Canada
Hansen MTB Inc.	BC	V3F 2K1	Canada
Montreal Mountain	PQ	H2Z 1S4	Canada
Amadablam		28001	Spain
Bicicleta Conexion		41101	Spain
Active Outdoors	IL	56478	USA
Desert Mountain	CA	89117	USA
Mountain Toad	NV	75369	USA
SFB Inc.	CA	94117	USA
Sierra Bicycle Group	CA	92549	USA
		86521	USA

The data is grouped by Country.

NOTE: To see an example of the above report, open the SGT05.RPT file in the \CRW directory.

Related Topics

See Chapter 6, *Reporting* 101

How to do a single field sort, Page 210.

How to do a multiple field sort, Page 213.

How to summarize grouped data

Often you will want to summarize the data in each group and print the summaries in your report. You can use summaries to:

- count the number of values in a group,
- calculate the sum, average, standard deviation, or variance of values in a group, or
- identify the minimum or maximum value in a group.

See *Summary functions*, Page 598-599.

Let's assume that you want to count the number of customers in each country. To do that, the data must be broken into country groups, and then the records in each group must be counted. Instead of grouping the data manually and then summarizing it, you can let Crystal Reports group and summarize in a single step.

To get started, create a report using the sample data, CRAZE.MDB, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}  
{customer.REGION}  
{customer.COUNTRY}  
{customer.POSTAL CODE}  
{customer.LAST YEAR'S SALES}
```



Now click the Customer Name field (the field you want to summarize) and click the SUMMARY button on the toolbar. The Insert Summary dialog box appears.

NOTE: *This example report has been designed to illustrate concepts only, not the actual look of your finished report.*

1. Select the desired summary operation.

2. Highlight the field that you want the data to be grouped by.

3. Specify the sort direction.

4. Click OK when finished.

Customer Name	Region	Postal Code	Country
BG Mountain Inc.	BC	V3F 2K1	Canada
Montreal Mountain	PQ	H2Z 1S4	Canada
Hansen MTB Inc.	PQ	H1J 1C3	Canada
Deely MTB Inc.	BC	V6Z 2H2	Canada
Number of customers in country			4
Sierra Bicycle Group	CA	92549	USA
Mountain Toad	CA	75369	USA
Desert Mountain	NV	89117	USA
Active Outdoors	IL	56478	USA
Sierra Mountain	NV	86521	USA
SFB Inc.	CA	94117	USA
Number of customers in country			6

The data is broken into Country groups.

This will count the records in each group.

NOTE: For more information see Summary functions, Page 598-599.

NOTE: To see an example of the above report, open the SGT06.RPT file in the \CRW directory.

Related Topics

Chapter 6, *Reporting 101*

How to subtotal grouped data, Page 221.

How to create multiple levels of subtotals, Page 227.

How to extend prices and then subtotal the extensions, Page 223.

How to calculate a percentage of the grand total, Page 238.

How to create Group Headers, Page 241.

How to subtotal grouped data

A subtotal is a special kind of summary. It totals or sums numeric values in a group. Even though it is a summary, the SUBTOTAL command is provided as a shortcut because it is used so frequently. Like any summary, Crystal Reports groups and subtotals in a single step. See *Summary functions*, Page 598-599.

To get started, create a report using the sample data, CRAZE.MDB, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}  
{customer.REGION}  
{customer.COUNTRY}  
{customer.POSTAL CODE}  
{customer.LAST YEAR'S SALES}
```

In this example you will group the data by Country and subtotal last year's sales by country. To do this, right-click the Last Year's Sales field and choose the INSERT SUBTOTAL command from the shortcut menu that appears. The Insert Subtotal dialog box appears with the chosen field listed at the top of the dialog box.

NOTE: *This example report has been designed to illustrate concepts only, not the actual look of your finished report.*

Choose the field you want the data grouped by, specify a sort direction and then click OK when finished.

The field that you selected to be subtotaled.

Data is broken into Country groups.

Customer Name	Region	Postal Code	Country	Sales
BG Mountain Inc.	BC	V3F 2K1	Canada	\$29,485.95
Montreal Mountain	PQ	H2Z 1S4	Canada	3,818.25
Hansen MTB Inc.	PQ	H1J 1C3	Canada	33,180.30
Deely MTB Inc.	BC	V6Z 2H2	Canada	
Subtotal				\$66,484.50
Sierra Bicycle Group	CA	92549	USA	
Mountain Toad	CA	75369	USA	
Desert Mountain	NV	89117	USA	\$18,778.80
Active Outdoors	IL	56478	USA	624.30
Sierra Mountain	NV	86521	USA	
SFB Inc.	CA	94117	USA	
Mountain Tops Inc.	IL	54321	USA	
Subtotal				\$19,403.10

The values in each group are subtotaled.

There are many different ways to group data. These next two examples show some popular methods.

NOTE: To see an example of the above report, open the **SGT07.RPT** file in the **\CRW** directory.

Related Topics

How to summarize grouped data, Page 219

How to create multiple levels of subtotals, Page 227.

How to extend prices and then subtotal the extensions, Page 223.

How to calculate a percentage of the grand total, Page 238.

How to create Group Headers, Page 241.

How to sort based on summarized group values

1. Choose the TOP N/SORT GROUP EXPERT command from the Report menu. The Expert appears with a tab for each of the groups in the report.
2. Select the tab for the group you want to sort.
3. Set the left hand drop down box to Sort All if it isn't set to that already.
4. In the top drop down box on the right, choose the summary you want to sort (if you have more than one summary in the group section. For example, you might have both a sum and an average summary in a section).
5. Click the option button for the sort direction you want.
6. If you want to select a second sort group, repeat Steps 2-5.

When you run the report, the program will sort your data based on the group order(s) specified. See *Summary functions*, Page 598-599.

How to extend prices and then subtotal the extensions

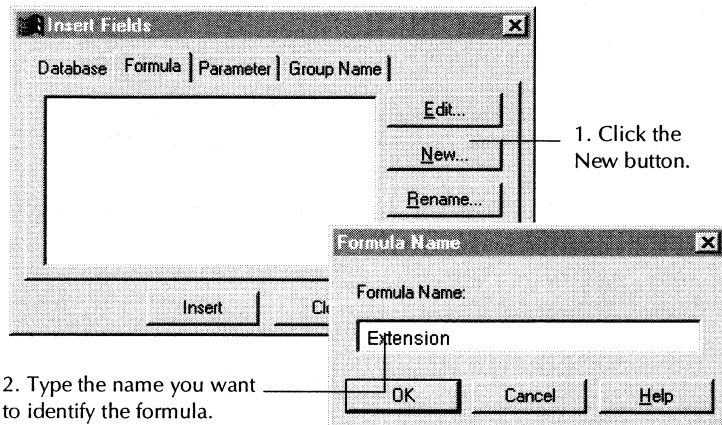
In an orders report or invoice, you may need to extend the prices for individual line items and then subtotal the extensions. You do this in Crystal Reports using a simple formula to extend the prices, and then you subtotal the formula field.

To get started, create a report using the Orders Detail table in CRAZE.MDB, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER ID}
{orders detail.PRODUCT ID}
{orders detail.QUANTITY}
{orders detail.UNIT PRICE}
```

To create the formula for extending the prices, click the **Formula** Tab in the **Insert Fields** dialog box to activate it.

Click *New* to create a new formula. The **Formula Name** dialog box appears. For this example, call the formula "Extension".



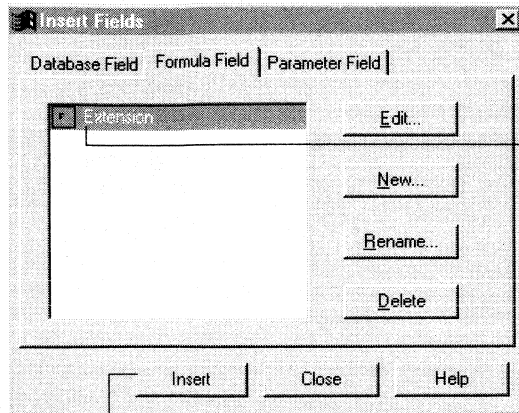
When you click **OK**, the **Formula Editor** appears. Enter the following formula into the *Formula* text box:

```
{orders detail.QUANTITY} * {orders
detail.UNIT PRICE}
```

You can either type it in exactly as it appears, or:

- double-click {orders detail.QUANTITY} in the *Fields* box,
- double-click the asterisk (the multiplication operator) in the *Operators* box, and then
- double-click {orders detail.UNIT PRICE} in the *Fields* box.

Click the *Accept* button when you are finished. The **Formula Editor** disappears and the program returns you to the **Insert Fields** dialog box with the name of your formula highlighted in the list box.



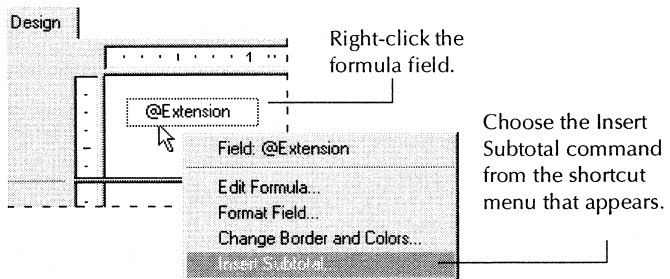
The formula you just created.

Click the Insert button to place the formula in your report.

Place the formula field to the right of the Unit Price field in the Details section of your report. See Chapter 10, *Formulas 101*, Page 249.

Click *Close* to close the Insert Fields dialog box.

To subtotal the extensions, right-click the formula field you just entered and select INSERT SUBTOTAL from the shortcut menu.



Right-click the formula field.

Choose the Insert Subtotal command from the shortcut menu that appears.

The Insert Subtotal dialog box appears, already set to subtotal @Extension (the extended price field). See *Summary functions*, Page 598-599.

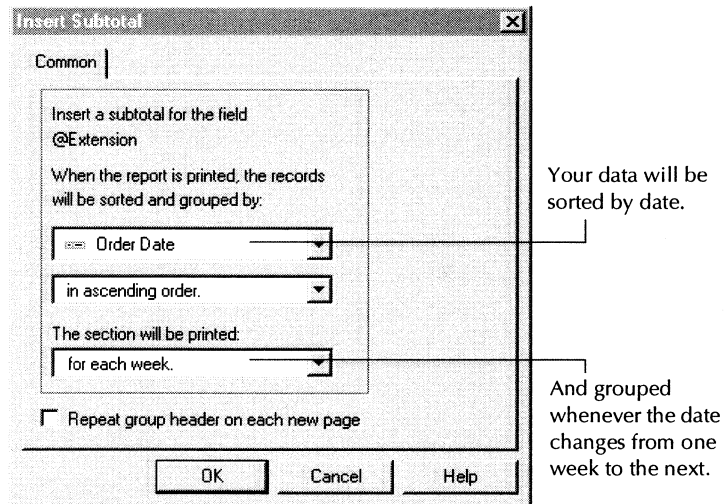
You are interested in weekly sales so you want to create a subtotal whenever the values in the Order Date field change from one week to the next. To do this:

- choose {order.ORDER DATE} as the sort and group by field, and
- choose *for each week* as the date change that is going to trigger the grouping.

NOTE: In order to do this, the Orders table must be linked to the Orders Detail table before you preview the report. CRAZE.MDB comes with the tables already linked. You will only have to relink the tables if you have toggled the Auto-Smart Linking option off in the Layout Tab of the File Options dialog box.

NOTE: You will not see where to choose for each week until you choose the Order Date field.

Your dialog box should look like this:



When you click OK, the program inserts the subtotal in the report.

And, with labels and formatting, your final report should look similar to this:

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

Customer ID	Product ID	Quantity	Unit Price	Amount
2	102181	1	\$1,739.85	\$1,739.85
	103171	1	\$899.85	\$899.85
	202201	2	\$539.85	\$1079.70
Sales for Week				\$3,719.40
2	101172	1	\$2,939.85	\$2,939.85
5	402001	2	\$274.35	\$548.70
	301161	2	\$764.85	\$1,529.70
	102151	1	\$1,739.85	\$1,739.85
Sales for Week				\$6,758.10
3	102171	1	\$1,739.85	\$1,739.85
	103181	1	\$899.85	\$899.85
	101182	1	\$2,939.85	\$2,939.85
Sales for Week				\$5,579.55

NOTE: To see an example of the above report, open the *SGT08.RPT* file in the \CRW directory.

Related Topics

See Chapter 10, *Formulas 101*

See Chapter 6, *Reporting 101*

How to summarize grouped data, Page 219

How to subtotal grouped data, Page 221

How to calculate a percentage of the grand total, Page 238

How to create Group Headers, Page 241

How to create multiple levels of subtotals

Sometimes one level of subtotal will not do. For example, you may want to see sales for each region subtotaled, but within each region you may want to see sales for each sales representative or each postal code subtotaled as well. Crystal Reports enables you to create these multiple subtotals with ease.

There are two keys to effective reporting using multiple subtotals:

- First, you need to make sure you enter the subtotals in the correct order. The rule to follow here is to go from the broad to the narrow. For example, if you want to subtotal by country and, within each country, by region, you enter the country subtotal first, then the region subtotal.
- The other key is to make sure you label the subtotals for clarity. When you have multiple subtotals, it can be difficult to tell which is which unless they are labeled properly.

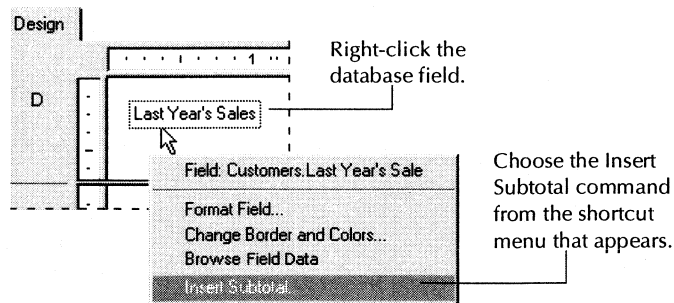
If you want to subtotal last year's sales by country and then by region, here is how you do it:

To get started, create a report using the sample data, CRAZE.MDB, and place the following fields from left to right in the Details section:

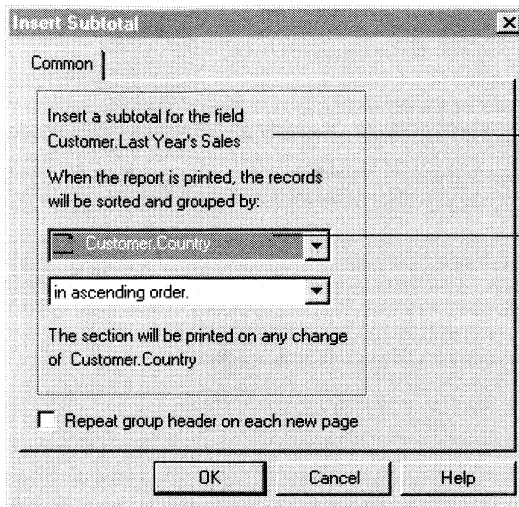
```
{customer.CUSTOMER NAME}  
{customer.REGION}  
{customer.COUNTRY}  
{customer.LAST YEAR'S SALES}
```

Right-click the Last Year's Sales field and select INSERT SUBTOTAL from the shortcut menu.

NOTE: In the screenshot below, menus have been shortened by removing some of the commands that are unrelated to the current discussion.



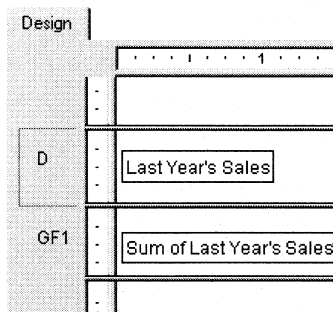
When the Insert Subtotal dialog box appears, select Country as your sort and group by field and click OK.



The field you are going to subtotal.

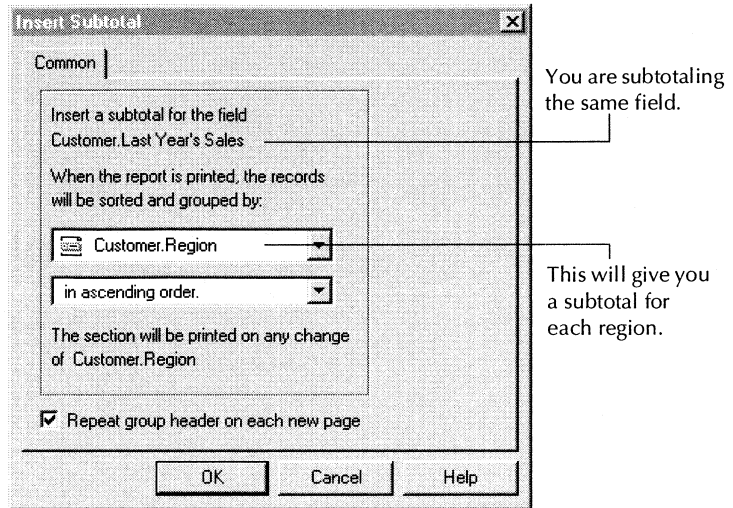
Every time the value in this field changes, this will give you a subtotal for each country.

The program creates a group section (GH1, GF1) and places the subtotal in the footer.

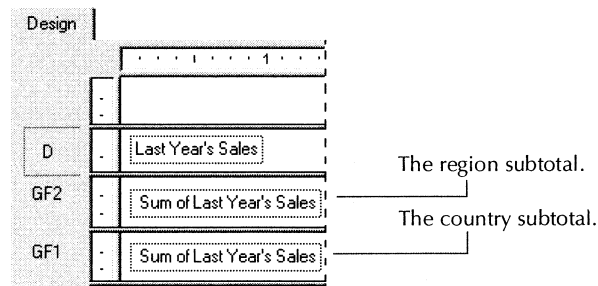


The country subtotal.

Now click the Last Year's Sales field again, right-click, and again select INSERT SUBTOTAL from the shortcut menu.



This time, select Region as your sort and group by field and click OK. The program creates a second group section (GH2, GF2) and places this subtotal in the footer. See *How to group data*, Page 215.



After adding the appropriate labels and formatting, your report should look something like this:

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

Customer Name	Region	Country	Sales
BG Mountain Inc.	BC	Canada	\$29,485.95
Deely MTB Inc.	BC	Canada	\$3,818.25
Total for BC			\$33,304.20
Allez Distribution	PQ	Canada	\$33,180.30
Montreal Mountiain	PQ	Canada	
Hansen MTB Inc.	PQ	Canada	\$5,579.55
Total for PQ			\$38,759.85
Total for Canada			\$72,064.05

Each region is subtotaled.

So is each country.

This report contains multiple levels of subtotals.

NOTE: To see an example of the above report, open the *SGT09.RPT* file in the \CRW directory.

Related Topics

See *Summary functions*, Page 598-599.

How to group based on first letter of company name

You might want to break your data into groups based on the first letter of the company name. In a customer list, for example, you might want all the "A" customers in a group, then all the "B" customers, and so forth. You can not do this by making dialog box choices; it requires the use of a formula.

Don't worry if you are not familiar with formulas yet. We will show you what formula you need here and how to enter it. You can learn more about creating and editing formulas in Chapter 10, *Formulas 101*, Page 249.

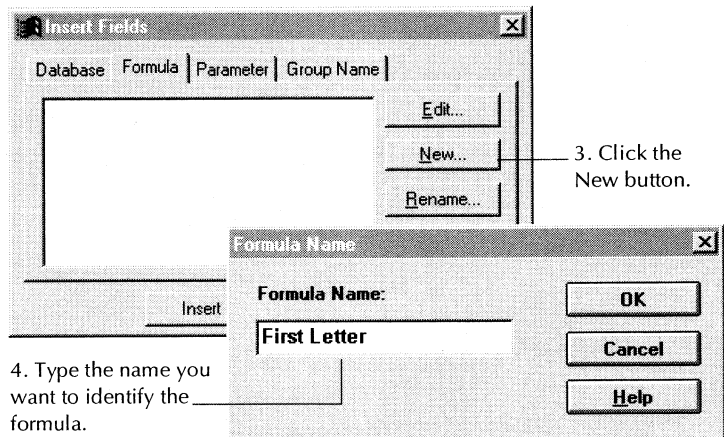
You are going to create a formula that will extract the first letter of each customer's name. Then you are going to group the data using that formula field as the sort and group by field. The program will then sort the data based on the first letter of each customer name and start a new group whenever the letter changes.

1. To get started, create a report using the sample data, CRAZE.MDB, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}
{customer.REGION}
{customer.COUNTRY}
{customer.POSTAL CODE}
```



2. With the Insert Fields dialog box, click the Formula Tab to activate it.



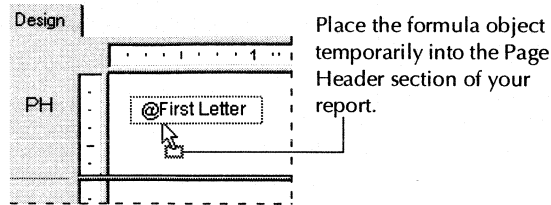
4. Type the name you want to identify the formula.

5. When you click OK, the Formula Editor will appear. Type the following formula into the *Formula* text box:

```
{customer.CUSTOMER NAME} [1]
```

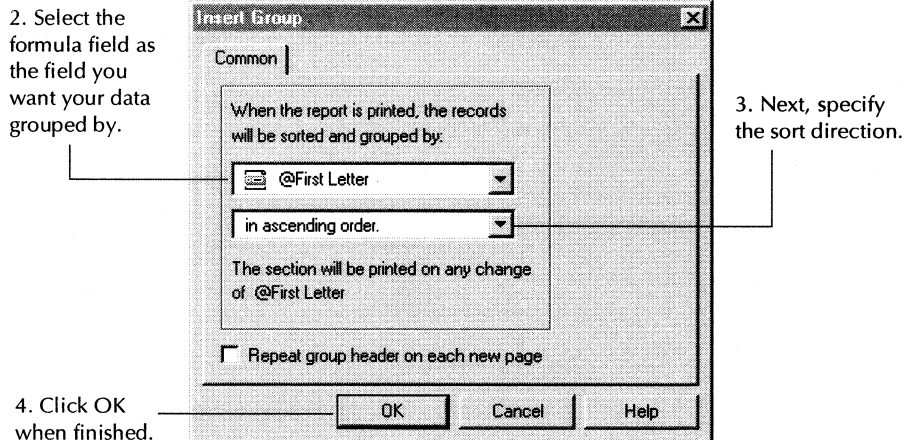
6. Click the *Accept* button. The Formula Editor disappears and Crystal Reports returns you to the Insert Fields dialog box with the name of your formula highlighted in the list box.

7. Click the *Insert* button to place the formula object in the report. Temporarily place the formula field into the Page Header (PH) section of your report.



We are going to move the formula object into a Group Header section, but the report does not have one yet. As soon as we group the data, we can move the formula object into the desired section.

1. Choose the GROUP command from the Insert menu. The Insert Group dialog box appears.



Crystal Reports will return you to your report with the data grouped by the formula field as specified.

5. Finally, move the formula field into the Group Header #1 section where it will serve as a live group header.

With labels and formatting your report should look similar to the following:

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

The data is broken into groups based on the first letter in the Customer's Name.

Customer Name	Region	Postal Code	Country
A			
Active Outdoors	IL	56478	USA
Allez Distribution	PQ	V6M 4G6	Canada
Amadablam		28001	Spain
B			
BG Mountain Inc.	BC	V3F 2K1	Canada
Bicicleta Conexion		41101	Spain
D			
Deely MTB Inc.	BC	V6Z 2H2	Canada
Desert Mountain	NV	89117	USA

The formula also provides a live header for every group.

NOTE: For information of formatting techniques, see Chapter 8, Formatting, Page 183.

NOTE: To see an example of the above report, open the SGT10.RPT file in the \CRW directory.

Related Topics

See Chapter 10, *Formulas 101*, Page 249

See Chapter 6, *Reporting 101*, Page 147

How to group data, Page 215

How to group data in intervals, Page 235

How to create Group Headers, Page 241

How to group data in intervals

You may want to group your data into intervals. Age groups, time periods and sales categories are some of the interval groupings you can create using the process you will learn here. In this example, you will rank customers by the amount of business they did in the previous year.

This example uses a kind of grouping known as "in specified order" grouping. This kind of grouping lets you specify the records that will be included in each group. You define the intervals you want and the program will do the rest.

1. To get started, create a report using the sample data, CRAZE.MDB, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}  
{customer.REGION}  
{customer.COUNTRY}  
{customer.POSTAL CODE}  
{customer.LAST YEAR'S SALES}
```

2. Right click the Last Year's Sales field and choose INSERT SUBTOTAL from the shortcut menu when it appears. The Insert Subtotal dialog box appears. You want to set up intervals based on the previous year's sales so select Last Year's Sales as the sort and group by field.
3. Now click the down arrow in the *Sort Order* list box and choose the *in specified order* option. The program creates a new Specified Order Tab in the dialog box.
4. Click the Specified Order Tab to activate it.
5. Click the *New* button. The program displays the Define Named Group dialog box.
6. Type "Less than \$10,000" in the Group Name box. This is the name that will appear as the Group Name field value for the group.
7. Since the first group is to contain only those records that have a last year's sales figure less than \$10,000, set the three drop down boxes so your condition reads:

is less than 10000

8. Click *OK* to return to the Specified Order Tab.
9. Click *New* again, and again the Define named group dialog box appears. This time we are going to set up our second group, a group that contains values from \$10,000 to \$25,000.
10. Type "\$10,000 to \$25,000" in the Group Name box.
11. Set the first two drop down boxes so your condition reads:

is between

When you make these settings, the program creates a fourth drop down box. There are now two drop down boxes on the right, one above the other, with the word "and" separating them. You use these drop down boxes to specify a range of values.

12. To specify the range,
 - type 10000 in the top drop down box, and
 - type 25000 in the bottom drop down box.

You have now set up the group to contain all values between \$10,000 and \$25,000.

13. Click *OK* to return to the Specified Order Tab.
14. To set up your final group, all those values over \$25,000, click the *New* button again.
15. When the Define Named Group dialog box appears, type "Over \$25,000" in the Group Name box.
16. Set the three drop down boxes so your condition reads:

is greater than 25000

17. Click *OK* to return to the Specified Order Tab.
18. Immediately to the right of the Specified Order Tab is the Others Tab. Use this tab to tell the program how you want it to handle all values that do not fit in any of the groups. Since all of the values in the Last Year's Sales field fall in one of the three groups, you won't have to worry about your settings here, so click *OK*. The program returns you to your report.

19. Only one other thing remains in setting up your interval report and that is to insert live headers to identify each of the groups. If you insert the Group Name field into the Group Header section of your report, the program will use the name you assigned to each group to identify those groups in the report. Click the *Insert Fields* button. The Insert Fields dialog box appears.
20. Click the Group Name Tab. There should only be one group name in the list, Group #1 Name.
21. Drag that name into the Group Header section.

With formatting, your report should look similar to this.

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

Customer Name	Region	Postal Code	Country	Sales
Less than \$10,000				
Active Outdoors	IL	56478	USA	\$624.30
Deely MTB Inc.	BC	T2F 8M4	Canada	\$3,818.25
Montreal Mountain	PQ	H2Z 1S4	Canada	\$4,918.00
Hansen MTB Inc.	PQ	H1J 1C3	Canada	\$5,579.55
\$10,000 to \$25,000				
Sierra Mountain	NV	86521	USA	\$11,842.95
Desert Mountain	NV	89117	USA	\$18,778.80
Over \$25,000				
BG Mountain Inc.	BC	V3F 2K1	Canada	\$29,485.95
Hansen MTB Inc.	PQ	H1J 1C3	Canada	\$33,180.30

This report is grouped by interval in a specified order.

NOTE: To see an example of the above report, open the SGT11.RPT file in the \CRW directory.

Related Topics

See Chapter 10, *Formulas 101*, Page 249

How to group based on the first letter of company name, Page 231

How to create Group Headers, Page 241

How to calculate a percentage of the grand total

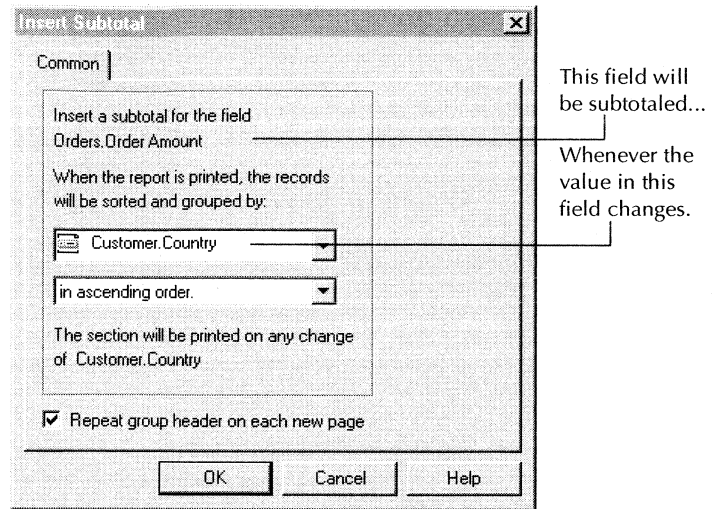
In some reports, you want to see what percentage of the grand total each group contributes. In this example, you will create a report that subtotals orders by country and then determines what percent of total worldwide sales each country generated. The process is simple and straightforward. It uses Crystal Reports' built-in subtotalling techniques plus one simple formula.

1. To get started, create a report using the sample data, CRAZE.MDB, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}  
{customer.REGION}  
{customer.COUNTRY}  
{orders.ORDER AMOUNT}
```

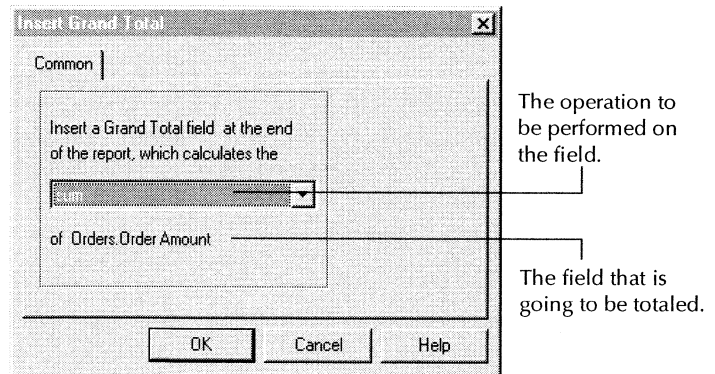
2. Right-click the Order Amount field and select INSERT SUBTOTAL from the shortcut menu.
3. When the Insert Subtotal dialog box appears, specify Country as your sort and group by field and click *OK*.

This calculates the sales for each country. It breaks your data into country groups and subtotals each group.



4. Now right-click the Order Amount field again and choose INSERT GRAND TOTAL from the shortcut menu. When the Insert Grand Total dialog box appears, make sure that the function is set to Sum and then click OK. See *Summary functions*, Page 598-599.

This calculates world wide sales, the total of all sales in the report.



All that is left to do now is to determine the percentage of worldwide sales generated by each country. To do this you will need to create a simple formula.

5. Open the Formula Editor and enter the following

formula:

```
Sum ({orders.ORDER AMOUNT},  
{customer.COUNTRY})/Sum ({orders.ORDER  
AMOUNT}) *100
```

For more information on entering formulas, see Chapter 10, *Formulas 101*, Page 249.

6. Place the formula in the group footer section of the report just to the right of the Country subtotal.

After you add labels, headers, and formatting, and you run the report, the Design Tab should look similar to this:

The screenshot shows a report design view with a table. The table has four columns: Customer Name, Region, Country, and Order Amount. The table is divided into sections: a header section with the column names, a group header section with 'Group #1 Name' and 'Group #2 Name', a data section with 'Customer Name', 'Region', 'Country', and 'Order Amount', a subtotal section with '@SubtotalText', 'SumOfOrder', '@Percent', and '%of total orders', a grand total section with '@GrandTotalText', 'SumOfOrder', '@Percent', and '%of total orders', and a final grand total row with 'GRAND TOTAL FOR ALL' and 'SumOfOrder'. The design view includes a ruler at the top and a 'Design' tab on the left.

Your results should look similar to this:

NOTE: This example report has been designed to illustrate concepts only, not the actual look of your finished report.

The program calculates the percent of total sales each customer and each country contributes.

Customer Name	Region	Country	Amount
SFB Inc.	CA	USA	\$7,911.80
			\$12,093.60
			\$9,606.30
			\$4,739.55
Subtotal for SFB Inc.			\$34,351.25 8.53% of total orders
Sierra Bicycle Group	CA	USA	\$19,766.20
			\$12,763.95
			\$8,398.50
Subtotal for Sierra Bicycle			\$40,928.65 10.16% of total orders
Sierra Mountain	NV	USA	\$8,233.50
			\$3,609.45
			\$13,543.05
Subtotal for Sierra Mountain			\$25,386.00 6.30% of total orders
Subtotal for USA			\$259,024.65 64.33% of total orders

Each customer is subtotaled.

Each country is subtotaled.

NOTE: To see an example of the above report, open the **SGT12.RPT** file in the **\CRW** directory.

Related Topics

See Chapter 10, *Formulas 101*, Page 249

How to create Group Headers

Whenever you create a group, a subtotal, or a summary, the program creates both a Group Footer section (where it places any subtotal or summary value), and a Group Header section (where you can place some text to identify the group if you wish). Group headers are useful, even necessary, if you want your report data to be clear and easily understood. In this section you will learn how to create the five most common kinds of group headers:

- standard headers,
- simple live headers,
- live headers that combine text, fields,
- live headers for groups based on a formula, and
- live headers for custom groups.

STANDARD HEADERS

A standard header is a block of text that is used to identify each group in a rather generic kind of way. "Customer", "State", and "Monthly Orders" are all examples of this kind of header.

Preview	
Region Sales Figures	
Adventure Cycling	AZ
Bob's Bicycle Distribution	AZ
The Edge	AZ
Region Sales Figures	
BG Mountain Inc.	BC
Deely MTB Inc.	BC
Region Sales Figures	
Adventure 16	CA
Nomad Bike Inc.	CA

While the header is somewhat descriptive ("Region Sales Figures" clearly means it is a regional group), you never know what region is in the group without first looking at the details in the group.

Nonetheless, this kind of header has its place and it is easy to create.



1. To create a standard header, click the INSERT TEXT OBJECT button.
2. When the object pointer appears, move the object frame into the Group Header section where you want it to print.
3. Click inside the frame to set the insertion point, then enter the text you want to use for your header.
4. Click outside the frame when finished to complete the process. Now when you run the report, the same header will appear at the beginning of each group.

For more information on inserting text objects see *How to insert text objects*, Page 95.

LIVE HEADERS

A live header is a header that changes based on the content of the group. If you have your data subtotaled by region, for example, a live header would typically identify the region detailed in each group. Thus, the Arizona group would have a header identifying the data as Arizona data, the California group would have a header identifying the data as California data, and so on.

NOTE: When you create a group, the program automatically inserts a group name field in the group header section unless you have toggled the option off using the Options command on the File menu. The information that follows details how you can manually insert such a section (if you don't have the program insert one automatically) and how to create different kinds of live headers for different needs.

GROUP NAME ONLY

The easiest live header to create is simply an identifying field value. To create this kind of live header for region groups, for example, you simply insert a Group Name field in the Group Header section. This prints Arizona (or AZ) at the beginning of the Arizona group, California (or CA) at the beginning of the California group, and so on. To do this:

1. Click the INSERT FIELDS button on the toolbar. The Insert Field dialog box appears.
2. Click the Group Name Tab.
3. Select the Group Name field that matches the group you are working with and drag it into the Group Header section for that group.
4. Format it the way you want it.

Now, when you run the report, the region identifier will appear as the group header for each region group.

GROUP NAME WITH TEXT

A more complex kind of live header combines text and a field value. A typical group header of this kind for data broken

down by region would be, "Sales for California" or "Customers in Postal Code 60606". To create these headers:

- insert a text object in the Group Header section,
- type in the text you want to appear, and
- enter the Group Name field in the text field where you want it to appear in the header.

For example, if you want your header to read "Sales for" and then the name of the region in the current group (Sales for AZ, Sales for CA, and so forth), follow these steps:

1. Choose the TEXT OBJECT command from the Insert menu.
2. Move the frame for that object in the Group Header section for the group.
3. Click inside the object to set the insertion point and type in "Sales for" and a space after it.
4. Choose the GROUP NAME FIELD command from the Insert Menu. The Insert Fields dialog box appears with the Group Name Tab active.
5. Select the Group Name Field that matches the group, and drag it into the text object, immediately after the text and the space you entered.
6. Format the text object as you want it to appear.

Now, when you run the report, the program will create a live header (with text) for each of your groups.

Preview

Sales for AZ	
Adventure Cycling	AZ
Bob's Bicycle Distribution	AZ
The Edge	AZ

Sales for BC	
BC Mountain Inc.	BC
Deely MTB Inc.	BC

Sales for CA	
Adventure 16	CA
Nomad Bike Inc.	CA

LIVE HEADERS FOR GROUPS BASED ON A FORMULA

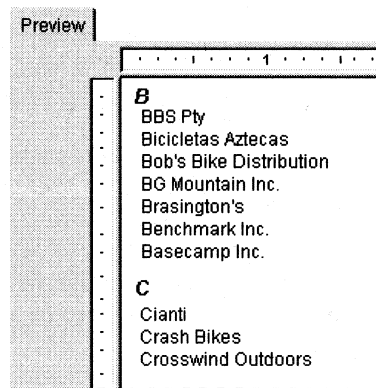
When you create a group and use a formula field as a sort and group by field, the program automatically creates a group name field based on the value returned by the formula.

For example, if you create this formula

```
{customer.CUSTOMER NAME} [1]
```

and then group on the formula, the program will group your data based on the first letter in the Customer Name field.

To create a live group header for a group based on a formula, simply insert the group name field in the group header section.



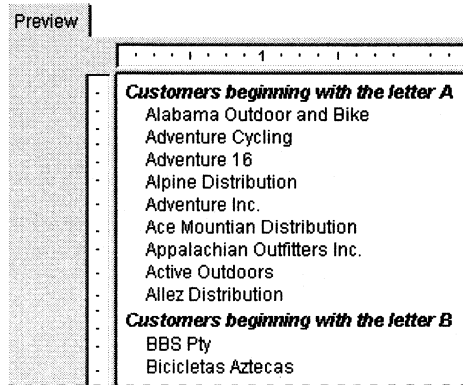
When you run the report, the "A" group will have the letter "A" as a header, the "B" group the letter "B", and so on.

To create a more descriptive header such as "Customers beginning with the letter B", you simply

- insert a text object in the Group Header section,
- type in this text:

```
Customers beginning with the letter
```
- add a space, and then
- insert the group name field into the text object and place it right after the space.

NOTE: If the program automatically inserted a group name field in the group header section, you will need to delete that field so you do not have duplicate headers.



HEADERS FOR CUSTOM GROUPS

The final kind of header is a header for the kinds of custom groups you create when you group things in specified order. When you use specified order grouping, you specify the name for each group and the records that belong in it. As in the other grouping situations, the program again creates a group name field for each group based on the group names you specified.

To insert a live group header for each of these groups,

1. Choose Group Name Field from the Insert menu
2. Insert the group name field for the custom groups into the custom group Group Header section.

The program will automatically apply each of the group names you assigned to the appropriate group.

NOTE: Make certain that when you assign the names to the groups using the Define Named Group dialog box, the names you assign are the names you want to appear as group headers.

Preview

	1	2
\$10,000 or less		
Active Outdoors		\$624.30
Bicicletas Aztecas		\$9,599.10
Deely MTB Inc.		\$3,818.25
Hansen MTB Inc.		\$0.00
La Bomba de Bicicleta		\$1,956.20
Montreal Mountain Sports		\$5,579.55
Between \$10,001 and \$25,000		
Desert Mountain Bikes		\$18,778.80
Sierra Mountain		\$11,842.95
\$25,001 or more		
Allez Distribution		\$33,180.30
BG Mountain Inc.		\$29,485.95

10

Formulas 101

What you will find in this chapter...

What are Formulas? 250

How formulas are created - An Introduction to the Formula Editor 254

Other formula conventions 258

Formula Syntax 261

How formulas are evaluated - Order of precedence 264

How to delete formulas from your report 264

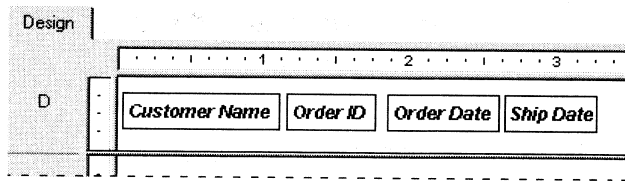
How to copy formulas from Crystal Reports online Help 266

How to copy formulas from one report to another 267

What are Formulas?

In many cases, the data you want to appear on your report already exists in fields in database tables. To prepare an order list, for example, you simply need to place the appropriate fields on your report.

By placing these fields . . .



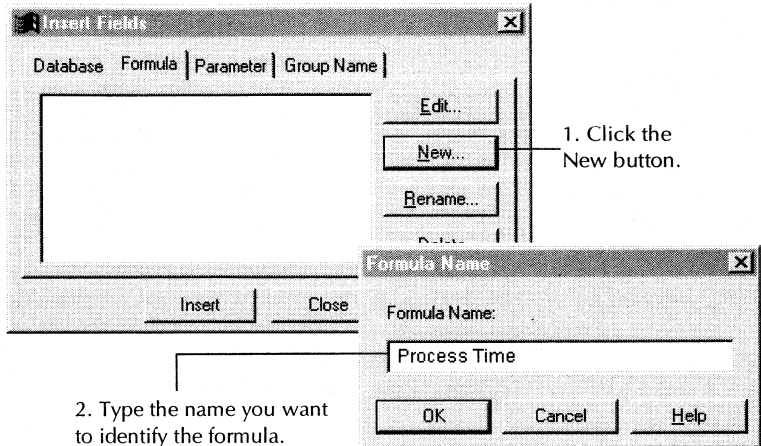
Customer Name	Order ID	Order Date	Ship Date
Bike-O-Rama	1143	1/30/95	1/30/95
ABC Incorporated	1092	2/16/95	2/16/95
The Pedallers	1296	2/27/95	2/28/95
The Cyclists Company	1366	3/1/95	3/1/95
Sporting Wheels Inc.	1387	6/14/95	6/15/95
CycleSporin	1717	6/24/95	6/30/95
Ride 'Em Cowboy	1763	8/8/95	8/9/95
Bob's Bikes Inc.	1952	9/1/95	9/2/95

You get this kind of report.

Sometimes, however, you need to put data on your report that does not exist in any of the data fields. In such cases, you need to create a formula. For example, to calculate the number of days it takes to process each order, you need a formula that determines the number of days between the order date and the ship date.



Crystal Reports makes it easy for you to create such a formula. To create the formula, you simply click the INSERT FIELDS button. The Insert Fields dialog box will appear. Click the Formula Tab to make it active.



When you click *OK* in the Formula Name dialog box, the Formula Editor appears. Use the Formula Editor to create, test and modify your formula.

To build the processing time formula, enter the Ship Date field, the subtractions operator, and the Order Date field.

For more information on the Formula Editor, see *How formulas are created - An Introduction to the Formula Editor*, Page 254.

First, double-click the Ship Date field in the Fields list.

Next, double-click the subtraction operator in the Operators list.

Double-click the Order Date field from the Fields list.

The formula components are entered as you choose them.

Accept Automatically checks syntax and accepts the formula for insertion into report.

Check Checks the formula syntax and identifies any errors.

Select Inserts the highlighted list item into the formula.

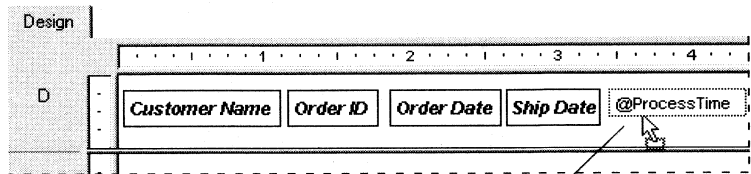
Browse Field Data... Shows the values in a highlighted field and lets you paste values.

NOTE: When you double-click items from the list boxes, Crystal Reports enters them in your formula complete with the brackets, punctuation, and other syntax items the Formula Editor needs to process them correctly. If you enter your formula manually, using the keyboard, you have to make certain you enter those syntax

items yourself. It is generally safer and quicker to build a formula by choosing list box items.

NOTE: The list of records in the Browse Field Data dialog box represents only a subset of the available records.

When you are finished, click the *Accept* button. Crystal Reports checks the formula syntax, closes the Formula Editor if the syntax is correct, and returns you to the Insert Fields dialog box. Click the *Insert* button to place the field object in your report.



When the field object box appears, place it where you want the formula to appear in your report.

Customer Name	Order ID	Order Date	Ship Date	Process Time
Bike-O-Rama	1143	1/30/95	1/30/95	0
ABC Incorporated	1092	2/16/95	2/16/95	0
The Peddlars	1296	2/27/95	2/28/95	1
The Cyclists Company	1366	3/1/95	3/1/95	1
Sporting Wheels Inc.	1387	6/14/95	6/15/95	4
CycleSporin	1717	6/24/95	6/30/95	6
Ride 'Em Cowboy	1763	8/8/95	8/9/95	1
Bob's Bikes Inc.	1952	9/1/95	9/2/95	1

You get this kind of report.

The formula subtracts the Order Date from the Ship Date and then prints the result here.

This is just one of many of ways you can use formulas to create powerful reports.

Some of the other typical uses for formulas are:

- creating calculated fields to add to your report,
`{orders detail.UNIT PRICE}* .85`
«Calculates a price discounted 15%.»
- formatting text on a report,
`UpperCase ({customer.CUSTOMER NAME})`
«Changes all the values in the Customer Name field to upper case. See *How to format text with formulas*, Page 273.»
- pulling out a portion, or portions, of a text string,
`{customer.CUSTOMER NAME} [1]`
«Extracts the first letter of the customer name. See *Subscript (:=)*, Page 628.»
- pulling out a portion of a date,
`Month ({orders.ORDER DATE})`
«Determines what month an order was placed in.»

These examples just scratch the surface. If you have a need for specialized data manipulation, chances are you can do it with a Crystal Reports formula.

Related Topics

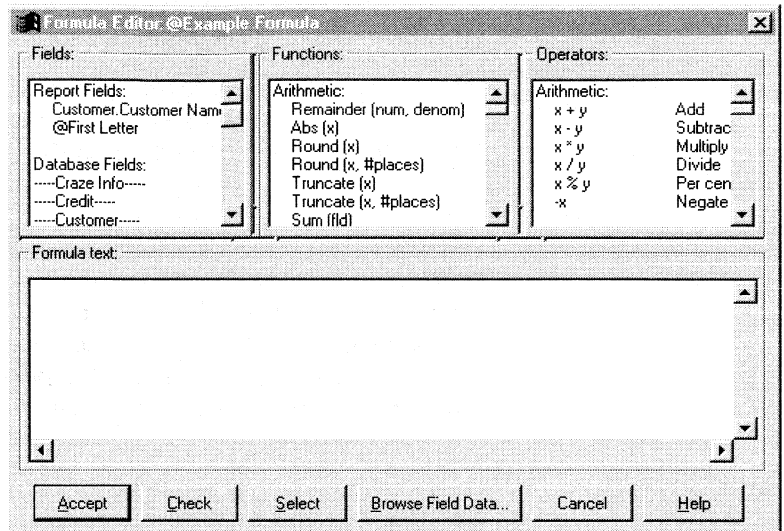
See Chapter 12, *Advanced Totaling*, Page 291.

See *Count function*, Page 574

See *Distinct Count function*, Page 578

How formulas are created - An Introduction to the Formula Editor

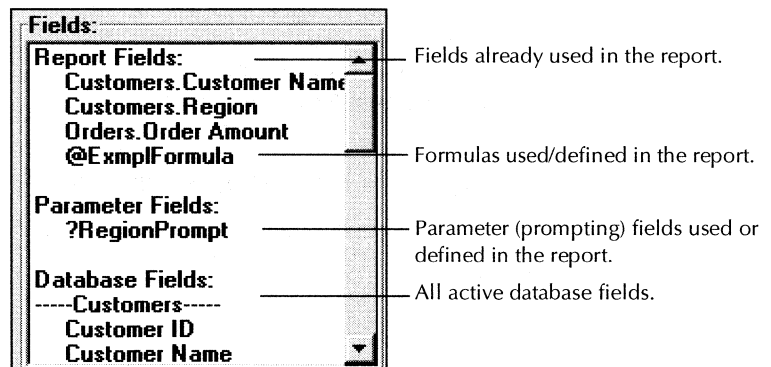
The Formula Editor is a dialog box that contains all the tools you need to create and check your formulas.



Using the Formula Editor you combine elements to create a working formula. The three list boxes at the top of the Formula Editor list the primary formula components.

Fields box

The *Fields* box lists all the fields you can use as formula arguments.



TO INSERT FIELDS IN YOUR FORMULA

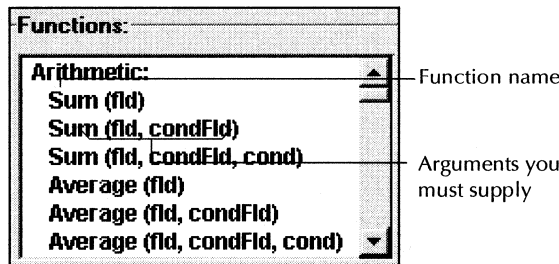
You can insert fields in your formulas in two ways:

- by placing the I-beam cursor where you want the field to appear in the *Formula* text box, setting the insertion point and then double-clicking the desired field in the *Fields* box, or
- by placing the I-beam cursor where you want the field to appear in the *Formula* text box and then typing it in manually.

NOTE: Be sure to review the correct syntax for using fields in your formulas. If you do not include the file name, leave out the separating period, or fail to surround the field in braces, Crystal Reports will generate a *Formula Compiler Error* message detailing your error. See *Formula Syntax, Page 261*.

Functions box

The *Functions* box lists the dozens of functions that are included with Crystal Reports. These functions are prebuilt procedures that return values. All functions available are listed in this box with their arguments and arranged by their use.



TO INSERT FUNCTIONS IN YOUR FORMULA

You can insert functions in your formulas in the same manner as fields:

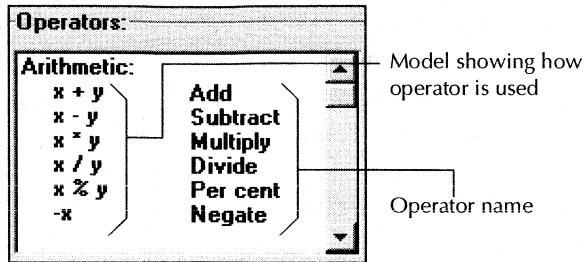
- via the *Functions* box, or
- by typing them in manually.

See *To insert fields into your formula, Page 256*.

NOTE: Be sure to review the correct syntax for using functions in your formula. If a function has required arguments, all arguments must also be entered. If any required arguments are not entered, Crystal Reports will generate a *Formula Compiler Error* message detailing the error. See *Formula Syntax, Page 261*.

Operators box

The *Operators* box lists the "action verbs" you can use in your formulas. (See Chapter 25, *Operators and Variables*, Page 609.)



TO INSERT OPERATORS IN YOUR FORMULA

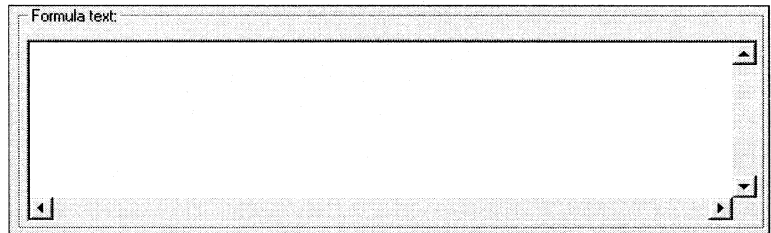
You can insert operators in your formulas in the same manner as fields and functions:

- via the *Operators* box, or
- by typing them in manually.

See *To insert fields into your formula*, Page 256.

Text box

The *Formula* text box is where you actually enter the formula.



You can:

- enter the formula manually, using the keyboard,
- choose your formula components from the *Fields*, *Functions*, and *Operators* boxes, or,
- combine the two methods, choosing some of the components from the list boxes and entering other parts of the formula manually.

TO INSERT TEXT AND NUMBERS IN FORMULAS

You insert text and numbers in formulas by typing them directly into the *Formula text* box. You can also click the *Browse Field Data* button, highlight the value you want to insert from the list that appears, and click *Paste* to paste the value into your formula.

NOTE: Be sure to review the correct syntax for text and numbers in formulas. See *Formula Syntax*, Page 261.

Other formula conventions

The following is a description of the remaining elements available for use in assembling a formula.

Other Formulas

Just as you can enter fields in formulas, you can enter other formulas in formulas. Crystal Reports performs the calculations in the inserted formula, and then uses the value returned by the referenced formula in the same way it uses any other value.

For example, the formula:

$$1 * (2+4*6/3-7*12-8) + 2 * (2+4*6/3-7*12-8) + 3 * (2+4*6/3-7*12-8) + 4 * (2+4*6/3-7*12-8) = -820$$

includes the expression $(2+4*6/3-7*12-8)$ repeated many times.

If you create a formula for the repeated expression ($@F = (2+4*6/3-7*12-8)$) and then reference that formula instead of entering the expression itself, you will get the same result.

$$1 * \{ @F \} + 2 * \{ @F \} + 3 * \{ @F \} + 4 * \{ @F \} = -820$$

Every time Crystal Reports sees the formula @F, it performs the @F calculation and returns the value -82, just as the expression underlying the formula $(2+4*6/3-7*12-8)$ returns the value -82. See *How formulas are evaluated - Order of Precedence*, Page 264.

TO INSERT OTHER FORMULAS IN FORMULAS

You can insert other formulas into your formulas in the same manner as you did other fields:

- via the *Fields* box, or
- by typing them in manually.

See *To insert fields into your formula*, Page 256.

NOTE: Be sure to review the correct syntax for using formula fields in your formula. See *Formula Syntax*, Page 261.

NOTE: A formula cannot reference itself.

Group Field values

Group field values are values that summarize a group (a group subtotal, a group average, and so forth). You can use them in formulas for many reasons. A typical reason would be to find out the percentage of the grand total that each group contributes (for example, what percentage of the \$2,300,000 US sales figure did the Western Region contribute?).

TO INSERT GROUP FIELD VALUES IN FORMULAS

You can insert group field values into your formulas in the same manner as you did other fields:

- via the *Fields* box, or
- by typing them in manually.

NOTE: Due to the syntax complexity of some group fields, it is highly recommended that you enter group fields by selecting them from the *Fields* box versus manually typing it in. See *Formula Syntax*, Page 261.

Related Topics

See Chapter 9, *Sorting, Grouping, and Totaling*, Page 201

Formula Comments

Formula comments are notes that you include with a formula to explain its design and operation.

Comments do not print and they do not affect the formula, but they appear in the Formula Editor. It is always a good idea to include comments with complex formulas, especially those formulas that will be used again and again over time.

TO INSERT COMMENTS IN FORMULAS

Type your comments in the *Formula* text box in the Formula Editor. The comment can be above or below the formula, or it can even follow the formula on the same line.

NOTE: A comment must be preceded by two slashes (//).

Any of the following placements are acceptable:

```
//This is an acceptable
//position for a comment.
//Note that when we force
//the line break, we have to
//begin each new line with
//double slashes.

If {orders.ORDER AMOUNT} in (100.00 to 250.00)
Then
    .10 * {orders.ORDER AMOUNT}
Else
    0

//This is also an acceptable comment position.
If {orders.ORDER AMOUNT} > 10.00 Then
    "" //This position is also acceptable.
Else
    "Flag"
```

See *Comment operator (//)*, Page 613.

Formula comment considerations

The following are considerations when including comments with formulas:

- The proper syntax for a comment is two forward slashes // followed by the comment. When Crystal Reports sees the two slashes, it realizes that the text that follows for the rest of the line is comment only and not to be evaluated as part of the formula itself.
- Crystal Reports treats everything that follows the slashes on the same line as a comment.
- If your comment is long and automatically wraps to the next line, no additional slashes are necessary; Crystal Reports treats it as one continuous comment.
- If you break your comment into two or more lines using the Enter key, you must begin each new line with two slashes. If you do not, Crystal Reports treats each unslashed line as part of the formula itself and displays an error message when you check the formula syntax.

Formula Syntax

Crystal Reports requires you to enter the various components of a formula according to a specific set of rules called syntax. Syntax, like the grammar of any language, takes practice to learn and perfect. Formula components must be written in a specific way and entered in specific order. The program uses syntax items (quotation marks, brackets, parentheses, etc.) to identify the various formula components, so it is very important that you stick to the rules in order for the program to recognize your formulas as well as have them work as you had planned.

The various components of formulas, and their syntax, are listed below:

Text

"Text" or 'Text'

(enclosed within 'single' or "double quotes")

When using text in formulas, it must be enclosed within 'single' or "double" quotes. Whatever text is within those quotes will be printed.

Numbers

23134.7

When using numbers in formulas, they must be entered without any comma separators or currency symbols.

Fields

{file.FIELD}

When using fields in formulas, they must be enclosed within French braces {} with the table name to identify which database table you are referring to followed by a period and then the field name within that table.

EXAMPLE

{customer.REGION}

This identifies the Region field from the Customer table.

Operators

1+1

When using operators in formulas, you simply type in the operator where desired. You may place a space before and after the operator if you desire, but to do so is optional (i.e., 1 + 1).

Functions

FunctionName (x)

When using functions in formulas, you simply type the function and supply the arguments as specified in the parentheses. For example, the Sum(x) function requires a field or formula as its (x) argument.

EXAMPLE

Average({order.AMOUNT})

This formula will calculate the average of all values in the Amount field.

Brackets in formulas

Crystal Reports uses three different types of brackets in writing formulas. Each one has a specific purpose and can be used only with certain formula elements.

- {} are called French braces and are placed around database, formula, and parameter fields:

{customer.REGION}, {@sum}, {?Region}

- [] are called Square brackets and are placed when using the Subscript or Array Operator:

{customer.CUSTOMER NAME} [1]

- () are called Parentheses and are placed around the arguments of a function:

Round(x, # places), Abs(x)

NOTE: Parentheses can also be used to control the order in which the formula elements are evaluated. See Order of precedence, Page 264, for further information.

HINT: A useful way to remember which brackets are used where is the following:

- French {} = Fields
- Square [] = Subscript (and Array)
- Parentheses () = Parameters

Whether you enter the formula manually or by double-clicking formula components from the component list boxes, you must use the correct syntax if you want your formula to work.

The diagram illustrates the process of selecting a function and its resulting syntax in Crystal Reports. It consists of three main parts:

- Functions List:** A window titled "Functions:" with a sub-section "Arithmetic:". It lists several functions: "Sum (fld)", "Sum (fld, condFld)", "Sum (fld, condFld, cond)", "Average (fld)", "Average (fld, condFld)", and "Average (fld, condFld, cond)". A horizontal line is drawn under "Sum (fld, condFld, cond)".
- Formula text:** A window showing the selected function and its syntax: "Sum (, ,)". Arrows point from the text to labels: "Commas between arguments" (pointing to the first comma), "Quotes for arguments" (pointing to the double quotes), and "Parentheses" (pointing to the opening and closing parentheses).
- Final Formula text:** A window showing the final formula: "Sum ({Orders.Order Amount},{Orders.Ship Date} , "weekly")". Arrows point from the text to labels: "Sums the values in the Order Amount field..." (pointing to the first field name), "whenever the date changes..." (pointing to the second field name), and "from one week to the next." (pointing to the "weekly" string).

Related Topics

See *How to subtotal grouped data*, Page 221.

How formulas are evaluated - Order of precedence

If you are creating formulas that contain different kinds of operators, it is important to consider the order in which Crystal Reports evaluates the separate parts of your formula. This order is called order of precedence.

Simple order of precedence follows basic math rules of precedence. Multiplication and division are performed first, from left to right, then addition and subtraction are performed. For example:

$$5 + 10 * 3 = 35$$

The multiplication $10 * 3$ is performed first to get 30. Then, 30 is added to 5 to arrive at the final answer, 35.

Now, if your intention is to add 5 to 10 and then multiply the sum by 3, you have to modify the order of precedence with parentheses. You can do that as follows:

$$(5 + 10) * 3 = 45$$

It is clear that parentheses have a higher precedence than the add, subtract, multiply, and divide operators. They redirect the order of calculation.

NOTE: When a formula contains other formulas such as:

`@Extension * 107.5%,`

the program first evaluates the embedded formula, @Extension, following order of precedence rules and then evaluates the rest of the primary formula.

How to delete formulas from your report

When you create a formula and enter it in your report, Crystal Reports:

- stores the specification for creating the formula, using the name you assigned it, and
- places a working copy of that formula at the point you specify in the report. A working copy is any occurrence of the formula in the report.

In order to delete formulas, you must delete the specification and all working copies of the formula.

NOTE: *You cannot delete the specification without deleting all working copies of the formula.*

To delete individual working copies of a formula

1. Select the formula copy you want to delete in your report.
2. Press the Delete or Del key on your computer's keyboard.

NOTE: *Even after you have deleted all of the working copies of a formula from your report, the formula specification remains unchanged. The specification is listed in the Formula Tab of the Insert Fields dialog box, and it is available for immediate use should you wish to enter the formula in your report again.*

Search for the Insert Fields dialog box in Crystal Reports online Help.

To delete the specification



NOTE: *This step can only be completed after you have deleted all working copies of the formula from your report.*

1. Once you have deleted all working copies of the formula, click the INSERT FIELDS button on the toolbar. The Insert Fields dialog box appears with the Database Tab active.
2. Click the Formula Tab to activate it.
3. Select the formula specification you want to delete from the *Formula name* list. Crystal Reports activates the *Delete* button.
4. Click the *Delete* button, and Crystal Reports deletes the formula specification.

NOTE: *If the formula is being used in another formula the program will delete the specification nonetheless.*

NOTE: *If you have not deleted all working copies of the selected formula, the program displays the message: Formula field in use, cannot be deleted!*

How to copy formulas from Crystal Reports online Help

Windows allows you to copy text from online Help topics to the Clipboard. You can then paste this text wherever it is needed. Since the formulas you develop using the Formula Editor are simply text, you can save yourself a lot of time by copying useful formulas directly into the Formula Editor and then modifying them to fit your needs.

To copy formulas from online Help

1. With the Formula Editor active, call up the Crystal Reports online Help in any of the standard ways (Help menu, F1 function key, etc.).
2. Regardless of the topic that first appears, use the Search or Find facility to find the topic that contains the formula of interest.
3. Call up the topic.
4. Choose the COPY command from the Edit menu. The Copy dialog box appears with the topic text displayed in a scrolling edit window.

NOTE: To do this you highlight the text and choose COPY from the Edit menu to copy the text to the Clipboard.

5. Scroll through the topic until you locate the formula you want to copy.
6. Select the formula by dragging the I-beam cursor over it, and click the *Copy* button when finished. Windows puts a copy of the selected text on the Clipboard.
7. Place the insertion point where you want the text to appear in the Formula Editor and press Ctrl-V to paste the text from the Clipboard.
8. Modify the formula to fit your needs.

NOTE: In Windows 95 and NT, highlight the text you want to copy and choose COPY from the Edit menu or press Ctrl-C to copy the text to the Clipboard.

NOTE: Change the fields, formulas, group fields, conditional statements, and text strings as necessary for use with the data used in the new report.

How to copy formulas from one report to another

You may find yourself wishing to copy a formula created in one report for use in another report. Copying formulas from one report to another is a simple procedure, but it requires careful attention to detail.

Since formulas are stored as text, it is a simple matter to copy the text formula from one report to another via the Clipboard.

To copy the contents of a formula from one report to another

1. In your report, select the formula field you want to copy.
2. Choose COPY from the Edit menu or press Ctrl-C.
3. Open the report you want to copy the formula to.
4. Choose PASTE from the Edit menu or press Ctrl-V.
5. When the program displays a rectangle placement frame, drag the formula where you want it.
6. Change the fields, formulas, group fields, conditional statements and text strings if necessary for use with the data used in this report.
7. To make these changes (if necessary), right-click the formula and choose Edit Formula from the shortcut menu. The Formula Editor appears.
8. Delete the old values and type in the new values, or select them from the *Fields*, *Functions*, and / or *Operators* lists. When making changes, use the following points as a guide:
 - All fields, formulas and group fields referenced in the formula copy must actually exist in the new report. This means that any database referenced in the original formula (or a database with the same structure, field names, and alias) must be active in the new report.
 - If such a database is not active, you must change the field, formula, and group field references in the formula copy to correspond to elements in your new report.
 - If your formula contains conditional elements, make certain that the conditions apply to the data in the new report. For

example, if the formula in your old report performed an action when the quantity was greater than 100, make sure that the greater than 100 condition makes sense in the new formula. When modifying a formula, you may find that greater than 10 or greater than 2000 makes more sense with your new data.

- If you are using the formula with new data, and if your report contains statements similar to the following:

```
If {file.FIELD} = "text string"
```

make certain that the text strings used in the formula match values that actually exist in the new data.

9. Click *Accept* when finished.

Related Topics

To insert fields in your formula, Page 256

To insert functions in your formula, Page 256

To insert operators in your formula, Page 257

11

Advanced Formulas

What you will find in this chapter...

How to create If-Then-Else formulas 270

How to create multi-condition If-Then-Else formulas 272

How to format text with formulas 273

How to use variables in formulas 273

How to declare a variable 277

How to assign a value to a variable 278

How to combine a variable declaration and assignment expression 280

How to declare and assign values to multiple variables 280

How to conditionally assign values to variables 281

How to use an array in a formula 282

How to use a range in a formula 284

How to use semicolons in formulas 286

How to debug a formula 287

This chapter covers many of the advanced aspects of working with the Crystal Reports formula language.

How to create If-Then-Else formulas

If-Then-Else formulas are conditional formulas: if a condition is met, then a certain consequence, an action, takes place. If the condition is not met, a different action takes place.

Example

- If a sales representative has already earned the maximum allowable bonus, print the amount of the maximum bonus allowed; if he has not yet earned the maximum, calculate the bonus actually earned and print it.
- If the value in the title field is "Mr.", print "Dear Mr." as the beginning of the salutation; if it is not "Mr.", print "Dear Ms." as the salutation.
- If the quantity on hand of a part is less than or equal to the reorder amount, reorder according to the reorder instructions; if the quantity is greater than the reorder amount, do nothing.

These are just a few of the kinds of conditional formulas you can create using the If-Then-Else operator.

When using the If-Then-Else operator, remember that there must be three separate parts to any If-Then-Else formula:

the <i>If</i> portion	This portion sets the condition.
the <i>Then</i> portion	This portion sets the action that takes place if the <i>If</i> condition is met.
the <i>Else</i> portion	This portion sets the action that takes place if the <i>If</i> condition is not met.

NOTE: The data types (text, number, currency, date, time, dateTime, or Boolean) for the *Then* and *Else* parts must be the same.

Thus, if the action that takes place if the condition is met (*Then*) is to print a text string, the action that takes place if the condition is not met (*Else*) must also be to print a text string, even if that text string is empty. For example:


```

If {file.FIELD} = 5 Then
    "Text String"
Else
    ""

If {file.FIELD} = 5 Then
    0
Else
    1

```

NOTE: You cannot create If-Then-Else formulas that use a Time or dateTime data type as a Then action and a null or empty Time or dateTime as an Else action because there are no null or empty values for those data types. To generate this same kind of result, follow the procedure outlined in the next section.

How to print Time or dateTime values conditionally

Assume that you want to print a string field only when the employee last name is Fuller. To do that, create a formula like this:

```

If {employee.LAST NAME} = "Fuller" Then
    {file.STRINGFIELD}
Else
    ""

```

Place the formula where you want the string field to print on your report. The program prints the value in the string field if the employee's last name is Fuller and it prints an empty string (nothing) for any other last name.

But if you want to print Time or dateTime values conditionally and print nothing if the condition is not met, you can not do it using an If-Then-Else formula because there is not such thing as an empty or null value for the Time or dateTime data type. Instead, you insert the field itself in your report and set its Suppress property conditionally. To do this

1. Place the dateTime field where you want it to print on your report.
2. Right-click the field and choose the FORMAT FIELD command from the shortcut menu that appears.
3. Click the Common Tab in the Format Editor to activate it,
4. Click the *Formula* button to the right of the *Suppress* property.

5. Type in the following formula in the Formula Editor when it appears.

```
{employee.LAST NAME} <> "Fuller"
```

Now when you run the report, the program will print the date`Time` field whenever the employee last name is Fuller and it will suppress the field when the last name is anything but Fuller.

NOTE: *You can also print string fields conditionally (like in the example at the beginning of this section) or fields of any other data type using this technique.*

How to create multi-condition If-Then-Else formulas

Crystal Reports allows you to create powerful multi-condition formulas using the If-Then-Else operator.

Multi-condition and nested If-Then-Else formulas can be set up in this general pattern: *If* the X (first) condition is met, *Then*, go to the Y (second) condition. *If* the Y condition is met, *Then* perform the Y action; if the Y condition is not met (*Else*), perform the Y alternative. If the X condition is not met (*Else*), perform the X alternative. Thus:

```
If {file.FIELD1} = "X" Then
  If {file.FIELD2} = "Y" Then
    "Y Action"
  Else
    "Y Alternative"
Else
  "X Alternative"
```

See *How to create If-Then-Else formulas*, Page 270.

This formula checks the field FIELD1 first.

- If the value of that field is "X," the FIELD2 field is checked.
 - If the value of that field is "Y," "Y Action" is printed.
 - If the value of FIELD2 is not "Y," "Y Alternative" is printed.
- If the value of FIELD1 is not "X," "X Alternative" is printed.

While multi-condition formulas look complex at first, after you have worked through one or two you will find that they are not as intimidating as they seem, especially given the work they perform.

How to format text with formulas

You can use Crystal Reports formulas to format text. For instance, Crystal Reports includes functions for removing unnecessary leading or trailing spaces from text strings and converting text entirely to upper or lower case.

Example

```
TrimLeft("  A1/4520/B12 ") = "A1/4520/B12"  
TrimRight ("A1/4250/B12  ") = "A1/4520/B12"  
LowerCase("Ronald Black") = "ronald black"
```

Related Topics

See Chapter 8, *Formatting*, Page 183.

How to use variables in formulas

Variables can be used to solve many formula problems, but they have two primary uses:

- streamlining formulas, and
- expanding formula capabilities.

Unlike a constant value which is fixed and unchanging, a variable can be repeatedly assigned different values. You assign a value to a variable, and the variable maintains the value until you later assign a new value. Then the variable maintains the new value until you later assign a newer value, etc.

Using variables to streamline formulas

With variables, you can write formulas much more efficiently than you can without using variables. For example, to evaluate the {customer.FAX} field to determine if the area code is for Washington state (206, 509) or British Columbia, Canada (604), and without the benefit of variables, you must write a formula similar to the following:

```

If {customer.FAX}[1 to 3] = "604" Then
    "BC"
Else
    If {customer.FAX}[1 to 3] = "206" or
    {customer.FAX}[1 to 3] = "509" Then
        "WA"
    Else
        " "

```

See *How to create an If-then-Else formula*, Page 270, and *Subscript operator()*, Page 628.

You have to write out the instructions for extracting the area code from the telephone number field ({customer.FAX} [1 to 3]) every time you want the formula to use the area code from the current record.

By using a variable (for example, AreaCode), you write those instructions one time. Crystal Reports, using those instructions, automatically extracts the area code from the {customer.FAX} field, and stores it in the variable each time it reads a new record. You simply reference the variable AreaCode whenever you want to use the area code from the current record in your formula. Here's an example of the formula using a variable:

```

StringVar AreaCode:={customer.FAX}[1 to 3];
If AreaCode = "604" Then
    "BC"
Else If AreaCode = "206" or AreaCode = "509"
Then
    "WA"
Else
    " "

```

See *How to create If-Then-Else formulas*, Page 270.

Not only does the streamlined version take less time to write, but it takes less time to process by Crystal Reports as well, so your report prints more quickly.

Using variables to expand formula capabilities

Besides their impact on streamlining formulas, variables allow you to expand your formula writing capabilities. Before discussing the specifics of using variables in formulas, it is important to understand some things about the way the Formula Editor reads formulas.

SPECIAL REQUIREMENTS FOR USING VARIABLES IN FORMULAS

The previous sections demonstrated formula elements that were pretty narrowly defined:

- a given operator only works in certain situations and with certain kinds of text and/or data,
- a function only works with a specific number of arguments, and each argument must be a specific data type, and
- If-Then-Else formulas work only if the data type of the *Else* part of the formula matches the data type of the *Then* part.

Such narrow definition allows you to create formulas, in many cases, simply by filling in the blanks, with the formula checker pointing out any mistakes you make.

Variables, however, are not narrowly defined. They are extremely flexible; you make them what you want them to be. You create them at will, and you include them in formulas as needed.

Because of this flexibility, it is necessary for you to define (declare) the variables before you use them so that Crystal Reports:

- is aware of them,
- understands how you intend to use them, and
- can set aside and set up the memory space they require.

You also need to assign values to the variables so Crystal Reports knows what value they are to return.

Crystal Reports knows only what you tell it about the variables. The fail-safe formula-checker routines that work automatically with the other formula elements work with variables only after you define them.

To use a variable in a formula, you must do three things:

1. declare the variable,
2. set the value of the variable, and
3. enter the variable in the formula.

Declaring the variable

Crystal Reports requires you to declare all variables prior to using them. When you declare a variable, you tell the program:

- the name you intend to use for the variable, and
- the type of data you want the variable to hold.

The program uses this information to set aside a piece of memory for receiving and storing the values that are assigned to the variable.

NOTE: If you declare a variable with the same name and data type in two or more formulas, the formulas share the same variable. Thus, if one formula sets the value of the variable, the variable in the second (and additional formulas) reflects the change.

Naming the variable

You can name the variable anything you wish with the following qualifications:

- the variable name must not exceed 254 characters, and
- it cannot have the same name as a Crystal Reports operator or built-in function.

NOTE: As a general rule, you are probably better off if you keep the variable name short, easy to remember, and unique (not so close to the name of another variable as to cause confusion).

Variable Data Types

The data type of a variable determines the type of data that can be stored as a value in that variable. With Crystal Reports, you can create a variable with one of seven data types:

- number (100000)
- currency (\$30,000.00)
- Boolean (TRUE)
- date (January 1, 1996)

- string ('Hello')
- time (11:59:01)
- dateTime (96/12/31 11:59:59 P.M.)

The data types correspond to the data types used throughout Crystal Reports. Search for *data types* in Crystal Reports online Help.

How to declare a variable

You declare a variable at the beginning of the formula that uses the variable.

NOTE: *You can not declare variables globally. If you are using a variable that was declared in another formula, you must declare it again.*

NOTE: *If you declare a variable with the same name and data type in two or more formulas, the formulas share the same variable. Thus, if one formula sets the value of the variable, the variable in the second (and additional formulas) reflects the change.*

Variables must be declared in the following manner: The datatype and the variable name, then a semicolon to mark the end of the declaration.

For example, to declare a number variable named Amount, enter the following declaration statement:

```
NumberVar Amount;
```

If you want to declare a Boolean variable named Outstanding, enter the following declaration statement:

```
BooleanVar Outstanding;
```

If you want to declare more than one variable, you can list them accordingly. Each variable is separated by a semicolon. For example:

```
NumberVar Amount;  
BooleanVar Outstanding;  
DateVar MonthEnd;
```

Crystal Reports uses your declaration statement to set aside a block of memory to hold each of the variable values, and to assign a default value to each memory block. The default value assigned depends on the data type you declared for the variable. The default values assigned are as follows:

<i>Data type</i>	<i>To declare</i>	<i>Default value</i>
number	NumberVar	0
currency	CurrencyVar	0
Boolean	BooleanVar	False (No, 0)
date	DateVar	date (0, 0, 0)
string	StringVar	empty string ("")
dateTime	DateTimeVar	No default ¹
time	TimeVar	No default ¹

¹ Since Time (0 0:00:00) is midnight, you cannot use 00:00:00 as a default time and you can not use this time value as part of a default dateTime either. To avoid confusion, a default is not assigned here.

How to assign a value to a variable

You assign a value to a variable using an assignment statement. The assignment statement consists of:

- the variable name,
- the assignment operator, and
- the value you want to assign to the variable.

VARIABLE NAME

The variable name is the name you used to declare the variable.

ASSIGNMENT OPERATOR

The assignment operator is a colon followed by an equal sign (:="). See *Assignment operator (:=)*, Page 611.

VARIABLE VALUE

The variable value is any value you want to assign to the variable.

Example assignment statement

A variable value can be a constant, an expression, or a sequence of expressions.

Here are the assignment statements for assigning different kinds of values to variables:

```
Amount := 0
```

«Initializes the variable Amount to zero.»

```
Amount := 100
```

«Assigns the value 100 to the variable Amount.»

```
Amount := Amount + {orders detail.QUANTITY}
```

«Assigns the result of a calculation to the variable Amount. The calculation adds the value of the quantity field {orders detail.QUANTITY} to the current value of the Amount variable. This type of expression is useful in running total situations where each running total consists of the current amount plus an additional value.»

```
Amount := {file.QUANTITY1} + {file.QUANTITY2} +  
{file.QUANTITY3}
```

«Adds the three quantity fields and assigns the result to the variable Amount.»

```
Customer := "Westside Motors"
```

«Assigns the string "Westside Motors" to the variable Customer.»

```
Customer := {customer.FIRST NAME} +  
{customer.LAST NAME}
```

«Concatenates two fields and assigns the concatenated value to the variable Customer.»

```
Customer := TrimRight({customer.FIRST NAME}) +  
{customer.LAST NAME}
```

«Trims the trailing blanks from the first name field ({customer.FIRST NAME}), concatenates that field to the last name field ({customer.LAST NAME}), and assigns the concatenated value to the variable Customer.»

```
Customer := "Mr. " + {customer.LAST NAME}
```

«Concatenates the string "Mr. " with the value of the last name field {customer.LAST NAME}, and assigns the concatenated value to the variable Customer.»

```
Amount:= 100; Customer:= "Westside Motors";
```

«Assigns the constant 100 to the number variable named Amount, and assigns the string "Westside Motors" to the variable Customer. You can assign values to multiple variables by separating the assignment statements with semicolons.»

How to combine a variable declaration and assignment expression

For efficiency, Crystal Reports gives you the ability to declare a variable and assign it a value in a single line of formula code. To do this, simply declare the variable, allow a blank space, enter the assignment operator, and assign the value. For example, to declare a currency variable SellPrice and assign the value of the Cost field ({product.PRICE (SRP)}) times two (a 100% markup), you would use this expression:

```
CurrencyVar SellPrice:={product.PRICE (SRP) }  
* 2;
```

To declare a Boolean variable OverQuota and assign the result of the comparison {file.SALES}>{file.QUOTA}, use the following expression:

```
BooleanVar OverQuota:=  
{file.SALES}>{file.QUOTA};
```

How to declare and assign values to multiple variables

In the previous section, you learned how to declare a variable and assign a value to it in a single step. There may be times, however, that you want to declare multiple variables and assign values to each of them in the most efficient manner possible. To do this, simply chain the declaration/assignment expressions together, separating them with semicolons. For example, to declare two variables (a number variable Quantity, and a currency variable SellPrice) and then to assign values to each variable (the number

5 to the variable `Quantity`, and `{file.COST} * 2` to the variable `SellPrice`), you use chained expressions similar to the following:

```
NumberVar Quantity:= 5;  
CurrencyVar SellPrice:= {file.COST} * 2;
```

See *Assignment operator (:=)*, Page 611.

How to conditionally assign values to variables

Crystal Reports formula language gives you the ability to assign different values to variables based on conditions being met or not met. Consider the following formula:

```
NumberVar Total;  
NumberVar Result;  
  
Total:= Total + {invoices.ITEM TOTAL};  
  
If Next ({invoices.CUST#}) <> {invoices.CUST#}  
Then  
    (Result:= Total; Total:= 0)  
Else  
    Result:= 0;  
  
Result;
```

See *How to create If-Then-Else formulas*, Page 270.

The If-Then-Else part of this formula says that if the *If* condition is met (if the customer numbers [`{invoices.CUST#}`] are not equal), Crystal Reports is to do two separate things:

- assign the value stored in the variable `Total` (the running total) to the variable `Result`, and
- reset the value in the variable `Total` to 0.

If the *If* condition is not met (if the customer numbers are equal), Crystal Reports is to assign the value 0 to the variable `Result`.

How to use an array in a formula

An array is a special type of variable that can hold several values at once. The entire array can be passed to a summary function for evaluation, or separate elements of the array can be extracted using the Subscript operator (Page 628).

A common use for an array is to store the names of the days of the week:

```
StringVar array Weekdays:= ["Sunday",  
"Monday", "Tuesday", "Wednesday", "Thursday",  
"Friday", "Saturday"];
```

The array variable is declared using a variable declaration operator specific to the type of data the array will hold, `StringVar` in this example. The word *array* and the name you assign to the array follow the variable declaration operator.

The Assignment operator (Page 611) is used to assign values to the array variable, and square brackets are used to enclose the values (elements) stored by the array. Each element is separated by a comma.

Crystal Reports assigns an index value to each element in the array according to the order of the elements. The first element is assigned index 1, the second element is assigned index 2, and so on. To extract an element from the array in your code, use the subscript operator with the index number for the element you want to extract:

```
Weekdays[5] = "Thursday"
```

Negative numbers can also be used to extract array elements:

```
Weekdays[-4] = "Wednesday"
```

You can create an array of values for any valid Crystal Reports data type. However, arrays have the following restrictions:

- All elements must be of the same data type.
- You must declare the array with one of the variable declaration operators.
- There can be a maximum of 100 elements in an array.

- Each element in an array of string values can have a maximum of 254 characters (the standard limit of any string value in Crystal Reports).

Examine the following examples of array declarations to become more familiar with arrays:

```
NumberVar array x := [1, 10, 44];

CurrencyVar array Cost := [$19.95, $79.50,
$110.00, $44.79, $223.99];

DateVar array PayDays := [Date(1999, 05, 15),
Date(1999, 05, 31)];
```

Arrays can also be used without being assigned to array variables. For example:

```
["One", "Two", "Three"][2]
```

«Returns "Two" because it is the second item in the array.»

In some situations, you may prefer to use arrays dynamically like this. However, most formula situations that require arrays can be handled more easily by defining the array as an array variable.

Using arrays with summary functions

Summary functions accept arrays as parameters without requiring the array be declared as a variable. For example:

```
Average([5, 10, 15]) = 10
```

Use the square brackets to indicate that an array is being used with the function. See *Summary (x)*, Page 599.

A formula example

To better understand how arrays can be used in formulas, examine the following formula code:

```
StringVar array Weekdays := ["Sunday",
"Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"];

Weekdays[DayOfWeek({orders.SHIP DATE})]
```

«If DayOfWeek is 2, Monday will be returned because it is the second item in the array.»

This formula prints the name of the day of the week that each order was shipped. First, the Weekdays array is declared and

assigned string values for each day of the week. See *DayofWeek*, Page 577.

Next, the *DayOfWeek* function (Page 577) evaluates the date stored in the {orders.SHIP DATE} field and returns a number representing the day of the week (1 for Sunday, 2 for Monday, etc.).

Finally, the subscript operator is used with the *Weekdays* array to retrieve the name of the day of the week according to the number returned by the *DayOfWeek* function. The name of the day is returned by the formula and appears in the report.

How to use a range in a formula

A range is designed to conveniently handle entire ranges of values, values that fall between a minimum and a maximum value. A range variable is declared much like an array. A variable declaration operator appropriate to the type of values stored by the range is used. The word "Range" follows the variable declaration operator, along with the name of the range variable. For example, you would declare a number range variable as follows:

```
NumberVar Range
```

Finally, the assignment operator is used to assign the range of values the range variable will store. The range is indicated using the *Make Range* operator (Page 622) with a minimum and a maximum value for the range.

Ranges have two principle uses:

- To extract a range of values from all possible values, and
- To extract a range of characters from a string value.

Extracting a range of values

Consider the following example:

```
NumberVar range GradeA := 90 to 100;  
NumberVar range GradeB := 80 to 89;  
NumberVar range GradeC := 70 to 79;  
NumberVar range GradeD := 60 to 69;
```

```

If {student.TEST SCORE} in GradeA Then
    "A"
Else If {student.TEST SCORE} in GradeB Then
    "B"
Else If {student.TEST SCORE} in GradeC Then
    "C"
Else If {student.TEST SCORE} in GradeD Then
    "D"
Else
    "F"

```

See *How to create If-Then-Else formulas*, Page 270.

This formula starts by creating four range variables. Each contains a range of possible test scores. Notice that, unlike an array, no brackets are used to set off the range of values assigned to each range variable. Only the range operator (*to*) is used with the minimum and maximum test scores for each range.

The multiple If-Then-Else statement repeatedly evaluates the value in the {student.TEST SCORE} field to determine if it falls within a specific grade range. The formula prints letter grades in the report that are appropriate to the test scores received by each student.

Extracting a range of characters

The following formula demonstrates how to use ranges to extract characters from a string value:

```

StringVar AreaCode := {customer.FAX}[1 to 3];
If AreaCode = "604" Then
    "BC"
Else If AreaCode = "206"
    or AreaCode = "509" Then
    "WA"
Else
    ""

```

See *Subscript*, Page 628, and *How to create an If-Then-Else formulas*, Page 270.

This formula creates a variable that holds the first three characters in the string value of the {customer.FAX} field. For instance, if the value in {customer.FAX} is "6045551234", then:

```
{customer.FAX}[1 to 3]
```

returns "604". Notice that square brackets are used to indicate a range of characters in a string (unlike the range of numeric values seen in the previous example).

The variable AreaCode is assigned the value "604". The multiple If-Then-Else statement evaluates the value in the AreaCode variable to determine which region the fax number is in.

A range can be applied to a constant string value, as well:

```
"6045551234"[1 to 3]
```

«Returns "604".»

When referring to characters in a string, negative numbers can also be used:

```
"abcdef"[-3 to -1]
```

«Returns "def".»

NOTE: *You can not use a range as an element or as part of an element in an array.*

How to use semicolons in formulas

In a formula with multiple statements, the result of the final statement is the result that is returned (gets printed). When you have multiple statements in a formula, you must separate the statements using a semicolon so that Crystal Reports knows where one statement ends and the next begins. Without semicolons, Crystal Reports treats the entire formula as a single statement. In a multiple statement formula, this can result in an incorrect result or an error message.

The general rule that best describes the use of semicolons in multiple statement formulas is that every statement needs to end in a semicolon with two exceptions:

- the last statement in a formula does not need one, and
- the last statement before an *Else* (when there are multiple statements before an *Else*) does not need one.

How to debug a formula

Example

You have created a formula and you check it for errors using the *Check* button. Crystal Reports finds some errors. Now you need to know how to debug the formula.

Solution

Follow the example below to practice the necessary steps to debug a formula. After completing this exercise, use the same principles to debug your own formula.

NOTE: Be sure that you have the CRAZE.MDB database included to complete this exercise.

Debugging Steps

Below is the formula you can test for errors:

```
If ({customer.CUSTOMER NAME} [1 to 2 = "Co"  
and {customer.CUSTOMER ID} [1] = "2") Or  
({customer.CUSTOMER NAME} [1] = "Ha" and  
{customer.CUSTOMER ID} [1] = "2")  
    "PREFERRED CUSTOMER"  
Else  
    "DOESN'T FIT CRITERIA"
```

If correct, our formula should pick out all those companies whose names that begin with "Co" and whose customer numbers begin with "2" and those companies whose names begin with "Ha" and whose customer numbers begin with "3". When printing the field, those selections will read "PREFERRED CUSTOMER" while the rest will read "DOESN'T FIT CRITERIA."

We will now break down the formula to check and see that each condition of the formula is working individually.

To create the formula field (@Formula1)

Begin by inserting the fields {customer.CUSTOMER NAME} and {customer.CUSTOMER ID} on one line in the Details section for reference. For each portion of the formula you will place a new formula field next to these two fields in the report to check the condition.

1. Type the following in the Formula Editor:

```
If {customer.CUSTOMER NAME} [1 to 2 = "Co"  
Then  
    "TRUE"  
Else  
    "FALSE"
```

2. Click the *Check* button to test for errors.
3. You should receive an error message indicating: "A] is missing." Correct the formula by inserting the missing "] " after the 2.
4. Click the *Check* button again. You should receive a "No errors found" message.
5. Place the corrected formula field to the right of the two data fields in the Details section of your report.
6. Click the Preview Tab to see the values in the report and compare the fields to see if the field values returned by @Formula1 are correct. You should find "TRUE" listed next to the customer names that begin with "Co" and "FALSE" next to all the others.

Now we will check the other portions of our formula. Create @Formula2, @Formula3, and @Formula4 following steps 1 - 6 using the formulas specified below for each. Insert each formula field on the same line for easy comparison. Check each one for errors, fix as needed, and make sure the values returned are correct before moving on to the next formula.

To create the formula field (@Formula2)

1. Enter the following formula in the Formula Editor:

```
If {customer,CUSTOMER ID} [1] = "2" Then  
    "TRUE"  
Else  
    "FALSE"
```
2. When checking this formula, you should receive the following error message: "This field name is not known." This message is received because the field {customer.CUSTOMER ID} was not found.
3. Correct the formula by replacing the comma "," with a period "." in the field name.

4. Check the formula again. Your formula should now be error-free. Place the formula to the right of the @Formula1 field. You should see "TRUE" next to all customer numbers that begin with 2 and "FALSE" next to all customer numbers that do not begin with 2.

To create the formula field (@Formula3)

1. Enter the following formula in the Formula Editor:

```
If {customer.CUSTOMER NAME} [1 to 2] = 'Ha'  
Then  
    "TRUE"  
Else  
    "FALSE"
```

2. When checking this formula, you should receive the following error message: "The matching ' for this string is missing."
3. Correct the formula by changing the single quote before 'Ha' into a double quote.
4. Check the formula again. Your formula should now be error-free. Place the formula to the right of the @ Formula2 field. You should see "TRUE" next to all Customer names that begin with "Ha" and "FALSE" next to all Customer names that do not begin with "Ha".

To create the formula field (Formula4)

1. Enter the following formula in the Formula Editor:

```
If {customer.CUSTOMER ID} [1] = "3"  
    "TRUE"  
Else  
    "FALSE"
```

2. When checking this formula, you should receive the following error message: "The word 'then' is missing."
3. Correct the formula by typing in the word "then" at the end of the first line after "3".
4. Check the formula again. Your formula should now be error-free. Place the formula to the right of the @Formula3 field. You should see "TRUE" next to all Customer numbers that begin with 3 and "FALSE" next to all Customer numbers that do not begin with 3.

Now, if all of the formulas check error-free and the field values returned are correct, the next step is to create a formula that links the separate components together. We will begin by linking the first two formulas (@Formula1 and @Formula2) together in the formula field @Formula1+2 and then add @Formula3 and @Formula4 to create the final formula @FinalFormula.

To create the formula field (@Formula1+2)

1. Enter the following formula into the Formula Editor:

```
If {customer.CUSTOMER NAME} [1 to 2] = "Co"
and {customer.CUSTOMER ID} [1] = "2" Then
    "TRUE"
Else
    "FALSE"
```

2. Place the formula to the right of the @Formula4 field.
3. If this formula is working correctly, you can create one last formula adding the code from @Formula3 and @Formula4.

To create the formula field (@FinalFormula)

1. Enter the following formula into the Formula Editor:

```
If ({customer.CUSTOMER NAME} [1 to 2] = "Co"
and {customer.CUSTOMER ID} [1] = "2") or
({customer.CUSTOMER NAME} [1 to 2] = "Ha" and
{customer.CUSTOMER ID} [1] = "2") Then
    "PREFERRED CUSTOMER"
Else
    "DOESN'T FIT CRITERIA";
```

2. Place the formula where you want it to appear in the Details section of your report. (You can now delete all other formula field from your report.)

Now, your formula field should be error-free, field values returned correct, and we have debugged our formula. Use this same process of condition-by-condition testing for any formulas as a means of systematically checking them.

12

Advanced Totaling

What you will find in this chapter...

Introduction 292

How to maintain running totals in a list 292

How to maintain/subtotal running totals within groups 295

How to subtotal without grouping 298

How to subtotal true A to B, A to C reports 301

Introduction

Advanced totaling is any kind of report totaling that requires the extensive use of formulas. This chapter demonstrates the techniques needed to produce many of the most common reports that require advanced totaling. Each section takes you step-by-step through the process of creating the report and adding the necessary formulas.

The topics covered in this chapter are advanced techniques. Before starting, you should be familiar with how to design reports in Crystal Reports, how to group values in a report and summarize data. You should always know how to use the formula language, including variables, to create formulas and add them to a report.

How to maintain running totals in a list

Running totals are totals that are displayed generally on a record by record basis. They total all records (in the report, in the group, and so forth) up to and including the current record. For example, if your first three records have values of 2, 4, and 6, the running total for each of the three records would be 2, 6 and 12, respectively.

The most basic form of a running total is a single running total maintained throughout a list. In this example you will create that kind of report, setting up a running total for a list of order amounts.

NOTE: To see an example of this report, open the report file *RTO1.RPT* located in the *\CRW* directory.

1. To get started, create a report using the sample data, *CRAZE.MDB*, link the Customers and Orders tables, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}  
{orders.ORDER ID}  
{orders.ORDER AMOUNT}
```

2. Create a formula named @RunningTotal, and enter the following in the Formula Editor:

```
WhilePrintingRecords;  
CurrencyVar Amount;  
  
Amount := Amount + {orders.ORDER AMOUNT};
```

This formula says:

Add the value in the {orders.ORDER AMOUNT} field to the Amount variable.

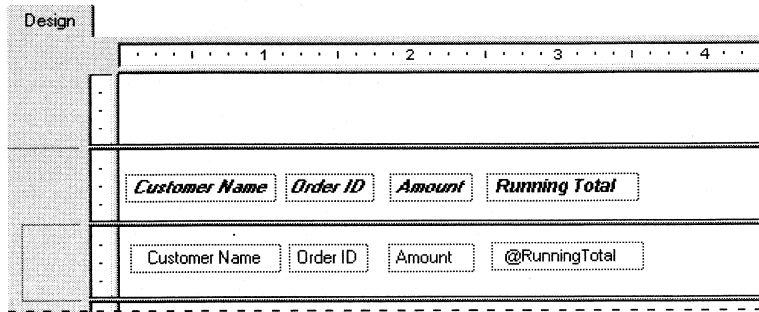
WhilePrintingRecords forces the formula to be evaluated while records are being printed in the report. This makes sure the Amount variable gets updated while each value in the Order Amount field is obtained.

At the beginning of the report, the Amount variable is created and automatically initialized to \$0.00. As the formula iterates through records, the value in the {orders.ORDER AMOUNT} field is added to the Amount variable. As a result, the Amount variable holds the current running total at any given time.

NOTE: If you have NULL values in the field you are doing a running total on, you will have to add code to check for NULL values or turn on the Convert NULL field value to default check box on the Reporting Tab of the File Options dialog box.

3. Place this formula in the Details section of your report, just to the right of the Order Amount field.

NOTE: When you place @RunningTotal in the Details section of the report, the formula executes for every record, thus keeping a running total of the Order Amount field.



When you add labels and formatting and run the report, it should look similar to the following:

Customer Name	OrderID	Amount	Running Total
Allez Distribution	1044	\$1,439.55	\$1,439.55
Allez Distribution	1044	\$1,439.55	\$2,879.10
Allez Distribution	1044	\$1,439.55	\$4,318.65
Allez Distribution	1044	\$1,439.55	\$5,758.20
Allez Distribution	1044	\$959.70	\$6,717.90
Allez Distribution	1044	\$959.70	\$7,677.60
Allez Distribution	1044	\$959.70	\$8,637.30
Allez Distribution	1044	\$959.70	\$9,597.00
Allez Distribution	1044	\$1,529.70	\$11,126.70
Allez Distribution	1044	\$1,529.70	\$12,656.40
Allez Distribution	1044	\$1,529.70	\$14,186.10
Allez Distribution	1044	\$1,529.70	\$15,715.80
Allez Distribution	1044	\$1,529.70	\$17,245.50
		\$658.00	\$17,903.50

The value in each record is added to the sum of the previous value in the report.

The total continues, unbroken, throughout the list.

Related Topics

How to maintain/subtotal running totals within groups, Page 295

See Chapter 10, *Formulas 101, Page 249*

See Chapter 11, *Formulas 201, Page 269*

See Chapter 6, *Reporting 101, Page 147*

How to maintain/subtotal running totals within groups

Another common use of a running total is for tallying items in a group. The running total starts with the first item in the group and ends with the last. Then it starts all over again for the next group. Then the next, and so on.

In this example, you will create a report that:

- maintains a running total of customer orders,
- groups customer orders and resets the running total for each group, and
- displays the subtotal for each order (the last running total for that order).

NOTE: To see an example of this report, open the report file *RTO2.RPT* in the *\CRW* directory.

1. To get started, create a report using the sample data, *CRAZE.MDB*, link the Customers and Orders tables as you did in the previous example, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}  
{orders.ORDER ID}  
{orders.ORDER AMOUNT}
```

2. Begin by creating a *@RunningTotal* formula just as you did in the previous example.

```
WhilePrintingRecords;  
CurrencyVar Amount;  
  
Amount := Amount + {orders.ORDER AMOUNT};
```

3. Place this formula in the Details section of your report, just to the right of the *{orders.ORDER AMOUNT}* field. This formula prints the running total of the values in the Order Amount field.
4. Choose the *GROUP* command from the Insert menu, and group the report on the *{customer.CUSTOMER NAME}* field.

5. Call up the Formula Editor and create @AmountReset.

```
WhilePrintingRecords  
CurrencyVar Amount :=0;
```

This formula says:

Set the value in the Amount variable to 0.

Place this formula in the Group Header #1 section of your report. Because the Group Header #1 section appears once for every group, @AmountReset will execute each time the group changes. Thus, the Amount variable is reset to 0 each time a new group begins.

NOTE: It is important to place this formula in the Group Header #1 section of the report.

6. Select the @AmountReset formula on the report and use the Format Editor to suppress it so that it will not appear in the final print-out.
7. Finally, call up the Formula Editor and create @AmountDisplay.

```
WhilePrintingRecords;  
CurrencyVar Amount;
```

This formula simply displays the current value of the Amount variable at any time. Place this formula in the Group Footer #1 section of your report. Because the Group Footer #1 section appears once for every group, @AmountDisplay will execute each time a group ends. Thus, the value stored in the Amount variable will be printed each time the group changes.

NOTE: This formula prints the same value that @RunningTotal prints as the running total for the last record in each group. But since it is printing it in the Group Footer section, it acts as a group subtotal, not as a running total.

NOTE: It is important to place this formula in the Group #1 Footer section of the report.

At this point the Design Tab should look like this:

Design	1	2	3	4
				@AmountReset
D	Customer Name	Order ID	Amount	@RunningTotal
GF				@AmountDisplay

When you add some labels and headings and format your report, your results should look similar to this.

Customer Name	Order ID	Amount	Running Total
CyclePath Corp.	10875	\$1,739.85	\$1,739.85
CyclePath Corp.	10898	\$1,799.70	\$3,539.55
CyclePath Corp.	12702	\$2,899.55	\$6,239.10
CyclePath Corp.	13635	\$1,799.70	\$8,038.80
		Total	\$8,038.80
Feel Great Bikes Inc.	10887	\$1,739.85	\$1,739.85
Feel Great Bikes Inc.	11792	\$1,739.85	\$3,479.70
Feel Great Bikes Inc.	12566	\$2,939.85	\$6,419.55
Feel Great Bikes Inc.	14470	\$548.70	\$6,968.25
		Total	\$6,968.25
Paris Mountain Sports	10890	\$1,739.85	\$1,739.85
Paris Mountain Sports	12545	\$2,939.85	\$4,679.70
Paris Mountain Sports	12756	\$1,664.70	\$6,344.40
		Total	\$6,344.40

The value in each record is added to the sum of the previous value in the report.

The running total starts fresh with each new group.

The final running total for each group becomes the subtotal for that group (Header total).

This report displays the running total within each group.

Related Topics

See *How to create Group Headers*, Page 241, See Chapter 10, *Formulas 101*, Page 249, Chapter 11, *Formulas 201*, Page 269, and Chapter 6, *Reporting 101*

How to subtotal without grouping

You may have times when you have an ungrouped list of values, and you only want to subtotal some of the values in the list. For example:

- you have a list that contains both Canadian and US customers,
- you want to keep customer records sorted alphabetically based on customer name,
- you do not want to break the data into groups based on the country, but
- you want a total of the values from just the Canadian records, and
- you also want a total of the values from just the US records.

You can do this kind of report easily using Crystal Reports formulas.

NOTE: To see an example of this report, open the *SUBTOTL.RPT* file in the *\CRW* directory.

1. To get started, create a report using the sample data, *CRAZE.MDB*, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}  
{customer.COUNTRY}  
{customer.LAST YEAR'S SALES}
```

2. Sort the records based on the {customer.CUSTOMER NAME} field.
3. Create the following formula, *@Build Canada vs. US totals*, and place it in the Details section, just to the right of the {customer.LAST YEAR'S SALES} field.

```
WhilePrintingRecords;  
CurrencyVar USTotal;  
CurrencyVar CanadaTotal;  
If {customer.COUNTRY} = "Canada"  
Then CanadaTotal := CanadaTotal +  
{customer.LAST YEAR'S SALES}
```

```
Else If {customer.COUNTRY} = "USA" Then
USTotal := USTotal +
{customer.LAST YEAR'S SALES}
```

This formula says:

If the COUNTRY value is Canada Then
add the value of LAST YEAR'S SALES to the
current value of the CanadaTotal variable
and assign that new value to CanadaTotal
Else If the COUNTRY is the USA
add the value of LAST YEAR'S SALES to the
current value of the USTotal variable and
assign that new value to USTotal

This formula reads each record and, based on the value in the Country field, adds the value in the Last Year's Sales field to the current total of a variable. If the country is Canada, the value is added to CanadaTotal. If the country is the USA, the value is added to USTotal. If the value is neither Canada nor USA, the formula does nothing.

NOTE: It is critical that you place this formula in the Details section. Since the formula has to evaluate each record it has to run every time the program prints a record and that happens only if you place it in the Details section.

4. You do not want the value returned by *@Build Canada vs. US totals* to appear in the report. Suppress the formula in the Format Editor so it does not print.
5. To display the Canadian totals and the US totals, you will need to create two formulas that print the current values in each of those variables. Create the following formula, *@Display US Total*, and place it in the Report Footer section of your report, immediately below the {customer.LAST YEAR'S SALES} field.

```
WhilePrintingRecords;  
CurrencyVar USTotal;
```

The first line forces the program to evaluate this formula when it is printing records (to make certain all the grouping and record selection is completed) and the second line prints the current value of the variable holding the US total (USTotal).

NOTE: It is critical that you place this formula in the Report Footer section so it gets evaluated and printed just once, at the end of the report. This will provide you with a total of all the US values for the entire report.

- Now create this formula, *@Display Canada Total*, and place it in the Report Footer section of your report, immediately below *@Display US Total*.

```
WhilePrintingRecords;
CurrencyVar CanadaTotal;
```

NOTE: Just like for @Display US Total, it is critical that you place this formula in the Report Footer section.

The Design Tab should look similar to this:

Design 1 2 3 4			
RH				
PH	Customer Name	Country	Last Year's Sales	
D	Customer Name	Country	Last Year's Sales	@Build Canada vs US
RF	USA Total		@Display US Total	
	Canada Total		@Display Canada Total	

When you add labels and formatting and run the report, your report should look similar to this:

Customer Name	Country	Last Year's Sales
The Cyclists Co.	Canada	\$38,199.10
The Pedallers Co.	USA	\$25,162.05
Ride 'Em Cowboy	USA	
Bob's Bikes Ltd.	USA	\$28,190.52
CycleSporin	USA	\$28,681.53
Bike-O-Rama	Canada	\$38,280.53
Biking's It	Canada	\$30,348.92
	USA Total	\$82,034.10
	Canada Total	\$68,548.02

How to subtotal true A to B, A to C reports

Throughout this manual, the term A to B, A to C report has been used to refer to any report in which a primary table is linked to two lookup tables. However, in a true A to B, A to C link, a single field in the primary table is used to link to both of the lookup tables.

In a true A to B, A to C relationship, one of the two lookup tables usually has more records than the other. If you group these records based on a field in the primary table, values in the smaller lookup table are repeated for each value in the larger lookup table. The following table shows data for an A to B, A to C relationships. The Customer table is linked to the Credits table and again to the Orders table. Notice that Jones has only one Credit ID, but that credit and its amount are repeated, once for each of Jones' two orders.

<i>Name</i>	<i>Credit ID</i>	<i>Credit Amount</i>	<i>Order ID</i>	<i>Order Amount</i>
Jones	1	-10.00	1	10.00
Jones	1	-10.00	2	12.00
Smith	2	-23.00	3	20.00
Smith	3	-45.00	4	30.00

NOTE: Notice that the Credit ID and Credit Amount for Jones is repeated.

If we use a standard subtotal on these groups for both the Order Amount field and the Credit Amount field, the single credit for Jones is counted twice, and the subtotal displays an inaccurate value of -20.00. It should only count this value once and display a subtotal of -10.00 because that is the total credit Jones has.

<i>Name</i>	<i>Credit ID</i>	<i>Credit Amount</i>	<i>Order ID</i>	<i>Order Amount</i>
Jones	1	-10.00	1	10.00
Jones	1	-10.00	2	12.00
		-20.00		22.00
Smith	2	-23.00	3	20.00
Smith	3	-45.00	4	30.00
		-68.00		50.00

NOTE: Notice that the Credit Amount for Jones is incorrect.

This problem would also occur in the Order Amount field if, for instance, Jones had two different Credit Amounts and only one Order Amount. We can avoid this problem by creating a formula for each field we want subtotaled, and placing the formula in the group footer.

NOTE: For an example of this report, open the TRUEABAC.RPT file in the \CRW directory.

1. To get started, create a report using the sample data, ORDRCR.MDB, and place the following fields from left to right in the Details section:

```
{cust.NAME}
{credits.CREDIT ID}
```



```
{credits.CREDIT AMOUNT}  
{orders.ORDER ID}  
{orders.ORDER AMOUNT}
```

NOTE: The sample data in *ORDRCR.MDB* has been specially designed to demonstrate a true A to B, A to C link.

2. In the Design Tab, select the {orders.ORDER AMOUNT} field and insert a subtotal, grouping on the {cust.NAME} field.
3. Create a second subtotal for {credits.CREDIT AMOUNT}, again grouping on {cust.NAME}.
4. Click the Preview Tab, and notice that the {credits.CREDIT AMOUNT} subtotal for Jones is twice as large as it should be (there is only one credit for Jones, but it appears twice, once for each order).
5. Create a new formula, named *Credits Subtotal*. Enter the following formula in the Formula Editor:

```
DistinctCount({credits.CREDIT ID},  
{cust.NAME}) *  
Average({credits.CREDIT AMOUNT},  
{cust.NAME})
```

This formula says:

Count the number of actual credits given to the customer in {cust.NAME}. Multiply this value by the average of all credit amounts displayed in the report for the customer in {cust.NAME}.

Jones has only one credit. The average of all of the credit amounts displayed for Jones is -10.00 (-10.00 + -10.00 = -20.00, -20.00 / 2 = -10.00).

6. Click *Accept* in the Formula Editor and place the formula in the group footer next to the subtotal on the {credits.CREDIT AMOUNT} field.
7. Notice that our new formula-based subtotal produces the correct result. Delete the original, incorrect subtotal for the {credits.CREDIT AMOUNT} field.

Related Topics

See Chapter 10, *Formulas 101*, Page 249

See Chapter 11, *Formulas 201*, Page 269

See Chapter 6, *Reporting 101*, Page 147

13

Parameter (prompting) Fields

What you will find in this chapter....

Parameter Field Objects Overview 306

How to create a parameter field 308

How to set record selection using parameter fields 309

How to use a parameter field in a formula 309

How to respond to parameter field prompts 310

How to use wildcards with parameter fields 311

How to do conditional formatting using parameter fields 312

How to set a report title using parameter fields 313

How to set sort order using parameter fields 314

Parameter Field Objects Overview

Parameter fields are fields that prompt you to specify a value each time you refresh the data in your report. When you supply a value, the program runs the report using that value. By using parameter fields in formulas, selection formulas, and in the report itself, you can create one report that you can modify quickly as your needs change. See Chapter 10, *Formulas 101*, Page 249.

Imagine that you are creating a report and you want to include only California records. Without parameter fields, you would enter a record selection formula similar to this:

```
{customer.REGION} = "CA"
```

This formula always tests the {customer.REGION} field to see if it holds the value "CA". If it does, it uses the record in the report. If it does not, it rejects the record. The report runs exactly the way you want it. This is fine if you always want to run the report using only California records. But if you want to run it using records from other states, you have to edit the formula and hard code your changes (for example, if you want to run the report using Arizona records, you would have to change "CA" to "AZ" in the Record Selection Formula Editor or the Select Expert). See *How to set up record selection using the Expert*, Page 426.

Using a parameter field in place of the state field value, however, lets you make changes "on the fly", without hard coding a new value. Here is how it works...

1. First, set up a parameter field using the PARAMETER FIELD command on the Insert menu. For the purposes of this example, we will call that parameter field *Region*.
2. Next, use the parameter field in your selection formula. Instead of using the formula:

```
{customer.REGION} = "CA"
```

Create a formula similar to this:

```
{customer.REGION} = {?Region}
```

NOTE: *{?Region}* is the parameter field you created in Step 1. The program uses the *{?ParameterFieldName}* format for parameter fields in formulas and Experts.

3. Finally, whenever you refresh the data in your report, the program prompts you to supply a value for the parameter field or to accept the default value. When you do, the program selects the records for the report using the value you specified.

Using parameter fields, you can create one report that can be customized quickly to meet a variety of needs.

Multiple Parameter Fields

Formulas and record selection formulas can contain multiple parameter fields. When you use multiple parameter fields, the program prompts you for each parameter before it refreshes the data. Thus, you can use a selection formula similar to the following:

```
{customer.REGION} = {?Region} AND  
{customer.LAST YEAR'S SALES} <= {?Sales}
```

to prompt you first for the region you want to report on and then, for the amount that you want to compare last year's sales against.

Parameter field considerations

There are a number of things to keep in mind when you are working with parameter fields.

- You do not have to place a parameter field in your report in order to use it in a record or group selection formula. You just create the parameter field and then enter it in your formula as you would any other field.
- The program supports parameter fields in the following data types: string, number, currency, Boolean, and date.
- Parameter field prompts can be up to two lines long with approximately 40-50 characters per line (depending on character width). The program performs automatic word wrap on prompts more than one line long.
- A parameter field can have only a single default value.
- You can use parameter fields in compiled reports to prompt for record selection criteria that you normally would not be prompted for in a compiled report. For example, if you activate the *Modify Selection Formula* check box in the Execute Compiled Report dialog box, the program gives you the opportunity to specify values in compiled report selection formulas that

contain fields and operators only. But if you place a parameter field in your selection formula, the program will prompt you for its value, even if that value is not a field.

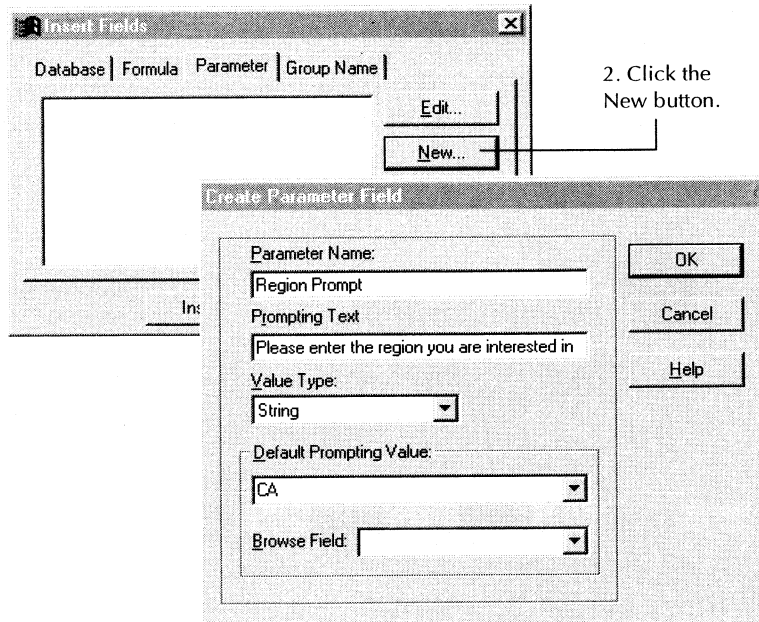
HANDS-ON

This section contains hands-on tutorials for performing a number of different tasks using parameter fields. The first three topics (*How to create a parameter field*, *How to use a parameter field in a formula*, and *How to respond to parameter field prompts*) are sequential. They provide an overview of a typical process using parameter fields. The remaining topics can be read individually, as needed.

How to create a parameter field



1. Click the INSERT FIELDS button on the toolbar. The Insert Fields dialog box appears. Click the Parameter Tab to make it active.



Your dialog box should look like this...

3. This information should be entered in the fields:
 - *Parameter Name*: Enter the name you want to use to identify the parameter field.
 - *Value Type*: Enter the data type of the parameter field.
 - *Prompting Text*: Enter the text you want to appear when the program prompts you.
 - *Default Value*: Enter the value you want the program to use if you do not supply a new value.

When you click *OK*, the program returns you to the Insert Fields dialog box with the new parameter field name displayed in the *Parameter Fields* list.

How to set record selection using parameter fields

1. Create a parameter field for each of the record selection criteria you want to be prompted to change.
2. Using the Select Expert, the Record Selection Formula Editor, or the Group Selection Formula Editor, create a record selection request that uses the parameter fields in place of fixed fields.

Now, when you refresh your data, the program will prompt you for the parameter values and base record selection on the values you provide. See *How to create a record or group selection formula*, Page 331.

How to use a parameter field in a formula

1. While still in the Insert Fields dialog box, click the Formula Tab and then click the *New* button, type a name for the formula in the Formula Name dialog box, and click *OK* to call up the Formula Editor.
2. Create your formula using the parameter field as you would any constant value. For example, instead of creating a formula that hard-codes the region name:

```
{customer.REGION} = "CA"
```

Use a parameter field in place of "CA":

```
{customer.REGION} = {?ParameterFieldName}
```

When you click *Accept*, the program returns you to the Insert Fields dialog box with the name of the formula you just created highlighted in the *Formula* list box.

3. Click the *Insert* button and place the formula where you want it to appear in your report.
4. Click *Close* to exit the Insert Fields dialog box.
5. To see how this field works, click the PRINT PREVIEW button to run the report.



NOTE: *The program automatically puts a question mark before the field name to identify it as a parameter field. See Chapter 10, Formulas 101, Page 249.*

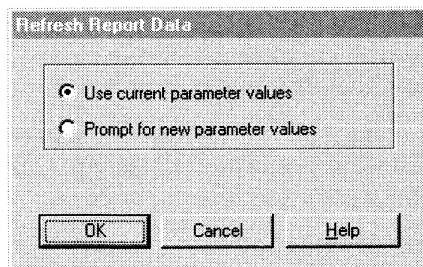
How to respond to parameter field prompts

When you run the report the first time (or you refresh the data from the Design Tab), the Enter Parameter Values for Main Report dialog box appears prompting you for a value. A picture of this dialog box appears below.

- If you specified a default value when you created the parameter field, the program will use that value unless you specify a new one.
- If you did not specify a default value, you must supply a value before the program will refresh the data.

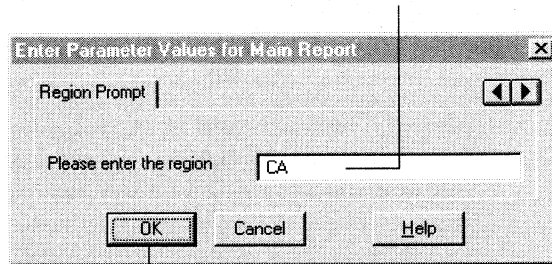


When you refresh data from the Preview Tab, the Refresh Report Data dialog box appears.



- Click the *Use current parameter values* option if you want to use the current parameter value.
- Click the *Prompt for new parameter values* option if you want to enter a new parameter value. If you choose this option and click OK, the Enter Parameter Values for Main Report dialog box appears.

If you want to use a different value than the default displayed, type a new value in the text box, and click OK.



If you want to use the default value, simply click OK.

The program will now run the report using the new value you specify.

How to use wildcards with parameter fields

You can use wildcards in your parameter field values to increase their flexibility. When prompted for a value, simply respond with a value using a wildcard.

NOTE: You can only use a wildcard in a parameter field value if the wildcard would have been appropriate in the non-parameterized value (for example, in formulas using the *Like* operator or the *LooksLike* function). See Chapter 24, *Functions*, Page 569.

Putting your parameter field in a selection formula using the *Like* operator, for example, and then responding to the prompt using a wildcard, you can create record selection requests like this:

```
{customer.REGION} like A?
```

«The program uses all the records that have a value in the Region field beginning with A, regardless of what the second

character is. This kind of request would return records from Alaska (AK), Alabama (AL), Arizona (AZ), and so forth.»

```
{customer.REGION} like ?A
```

«The program uses all the records with Region values ending in A, regardless of what the first character is. In this case, the request would return records from California (CA), Pennsylvania (PA), Washington (WA), and so forth.»

You can also use the * wildcard to create record selection requests like this:

```
{customer.POSTAL CODE} like 8*
```

Here the program uses all the records with Postal Code values beginning with 8. Because you are using a wildcard to designate any and all missing characters, the request would return both five digit and nine-digit (ZIP plus 4) postal codes.

You can restrict your report to a smaller postal code range by increasing the number of constant characters before the wildcard. For example:

```
{customer.POSTAL CODE} like 84*
```

In this case the program uses all the records with Postal Code values beginning with 84.

See Chapter 14, *Record and Group Selection*, Page 315, and search for *Selection Formula Templates* topic in Crystal Reports online Help.

How to do conditional formatting using parameter fields

You can create conditional formatting formulas using parameter fields that you can customize whenever you refresh the data in the report. A typical use for such a formula would be for color-flagging data if it meets certain conditions. For example:

- sales representatives who have sales more than 10% over quota,
- customers who have not ordered in the last quarter, and
- inventory items that have not had any movement in the last month.

If the conditions under which you flag these items never changes, you do not need to use parameter fields. You can just use formulas (for text flags) or conditional formatting (for color or border flags). But if you want to change the conditions from report to report, you can use parameter fields in formulas and conditional formatting formulas to do it.

1. Create the parameter field in the data type you need for the formula.
2. Create the formula and use the parameter field in place of the fixed value you would normally use. For example, to print the names in red of all the customers who had sales last year over a certain value (that you want to be prompted for), you can format the Customer Name field using a conditional formatting formula like this:

```
If {customer.LAST YEAR'S SALES} >
  {?ParameterFieldName} Then
  Red
Else
  Black
```

Now when you refresh the data, the program will prompt you for the threshold value (the value that triggers the color flag). It then runs the report and flags all the customers that had sales last year over the threshold figure. You can change the figure if you wish every time you run the report and the program will flag a different set of Customer Names each time. See *How to create If-Then-Else formulas*, Page 270, *Conditional Attribute Properties*, Page 189, *Conditional Formatting*, Page 188, and *How to format objects conditionally*, Page 178.

How to set a report title using parameter fields

To set a report title using parameter fields, you simply create a parameter field with the String data type and place it directly in the Report Header section of your report. Whenever you refresh the data in the report, the program prompts you to enter a title. Type in the title you want the program to use for your report. See *How to add a title page to your report*, Page 108.

How to set sort order using parameter fields

To set the sort order using parameter fields, you need to create a formula that includes a parameter field and then sort on that formula. For example, assume that you have a customer list report based on the Customer table. For each customer, you show the Customer Name, City, Region, Country, and Phone Number. You want to be able to sort the report by Country, by Region, or by City, depending on your needs at the time. To do this:

1. Create a parameter field and name it "SortField".
2. In the *Prompting Text* edit box, type in a prompt similar to this:

Type R to sort by Region or C to sort by City, otherwise data will be sorted by Country.

3. Create a formula similar to this and call it "Sort":

```
If {?SortField} = "C" Then
    {customer.CITY}
Else
    If {?SortField} = "R" Then
        {customer.REGION}
    Else
        {customer.COUNTRY}
```

This formula prompts for a value for the parameter field?SortField. If you enter "C", the formula returns the City field. If you enter "R" it returns the Region field. If you enter anything else, or do not enter anything at all, the formula returns the Country field. See *How to create multi-condition If-Then-Else formulas*, Page 272.

4. Place the formula in the Report Header section of your report and toggle its *Suppress* property on so it does not print.
5. Using the SORT RECORDS command on the Report menu, choose your formula, @Sort, as your sort field. See *How to do a single field sort*, Page 210.

Now when you run the report:

- the program will prompt you for a sort field,
- the formula will return a value based on your selection, and
- the sort facility will use that value as your sort field.

14

Record and Group Selection

What you will find in this chapter...

Record Selection 316

Group Selection 320

Record selection formula templates 322

How to set up record selection using the Expert 328

How to set up group selection using the Expert 329

How to create a record or group selection formula 331

How to use record/group selection templates 332

How to select the top or bottom N groups 333

Record Selection

Specifying records/groups to be included

When you select a field to appear on your report, Crystal Reports, by default, prints field values from every record in the active table(s). In many cases, you may not want to include all the values, but only a subset of those values. For example, you may want to include records only for a specific group of customers or for a specific range of account numbers out of the total number of records in the database. Or you may want to include values from only those records that fall within a particular date range. With Crystal Reports this is easy.

You can select records in one of two ways:

- using the Select Expert, and
- using the Formula Language.

Setting up record selection

USING THE EXPERT

Crystal Reports includes a very sophisticated formula language that you can use to specify virtually any kind of record selection that you want. Many times, however, you may not need the flexibility in record selection that the formula language provides. The Select Expert is designed for times like that.

NOTE: You can use the Select Expert to set up record selection and group selection requests. When you select either a group name or a summary field, the program knows that the selection criteria you set up is intended for group selection. In all other cases, the program knows that you are setting up record selection.

The Select Expert makes it easy to specify the records you want included in your report. You simply select the field that you want to base your selection on and then set the selection criteria. If you want to set additional criteria for the selected field or if you want to base record selection on additional fields, the Expert provides the tools you need to do it. See Chapter 10, *Formulas 101*, Page 249.

Using the Expert you can set up simple record selection requests:

- customers from Arizona,

- orders in the first quarter, or
- sales over \$10,000.

You can also use it to set up some very sophisticated requests:

- customers whose names start with "A", "M", or "S", or
- customers from California or Florida who ordered in July.

These are all range limit requests. One or more constants define the range. The program compares the field value in each record to the constant(s) and rejects records with values outside the range. The report is thus limited to values within the range. The Select Expert handles requests like this with ease.

NOTE: If you want to create a range limit request based on part of a field value, you are limited as to what you can do in the Expert. You can create a request that bases record selection on the first character in a field value using the Expert, for example. If you want to base it on the last character or two or more characters within the value, however, you will need to use the formula language to create a formula that subscript those characters out of the field values. (See Subscript (I), Page 628.)

You can set up all of these kinds of record selection requests without knowing anything about the Crystal Reports formula language.

NOTE: You can use the formula language from within the Select Expert if you wish, but that masks the Expert's main purpose. That purpose is to create powerful record selection requests using simple point-and-click technology.

USING THE FORMULA LANGUAGE

As discussed in the last section, you can use the formula language from the Select Expert. But if you are planning to create a selection formula using the formula language, you can go directly to the appropriate Selection Formula Editor using the EDIT SELECTION FORMULA command on the Report menu and then choosing RECORD or GROUP from the submenu when it appears.

When you are in the Formula Editor, you can build your record selection request using fields, functions, operators, and other formulas. Your only restriction is that the resulting formula must be a Boolean, that is, it must return either a True or False value. For

complete instructions on creating formulas, see Chapter 10, *Formulas 101*, Page 249. For sample record selection templates that you can modify for use in your report, see *Record/Group Selection Formula Templates*, Page 322. To copy and paste the templates directly into the Formula Editor, see the *Record/Group Selection Templates* topic in Crystal Reports online Help.

HOW TO DETERMINE WHAT FIELD(S) TO USE

When you select records, you are telling Crystal Reports to base your report only on those records that meet some conditions that you set. You base those conditions on what kind of information you want in your finished report.

Assume, for example, that you want a report that only shows California data. Your challenge is to find the best way to identify those records that come from California.

- If the table that you are using for your report has a state field or a region field, you can specify in your request that the program use only those records where the value in the state field is equal to California (Region is equal to CA). This is clearly the easiest way.
- If the table does not have a state field and you still want to report only on California data, do not lose hope. You may be able to identify that data in some other way.
 - If the table has a Postal Code or Area Code field, you could base your record selection
 - on the range of ZIP codes that apply to California (Postal Code is between n and N), or
 - on California Area Codes (Area Code is one of x,y,\dots,z).

NOTE: *If the Area Code is stored in the telephone number field, you will not be able to do record selection in the Expert based on the Area Code. You will have to create a record selection formula using the formula language to extract the Area Code part of the phone number and then do record selection on that. See Subscript [], Page 628.*

Clearly you are not locked into any one method of record selection. Just because you are not locked into it, however, you should still use care when you set up your selection criteria. The next section, *Selection Performance Tips*, Page 325, details some of the things you should consider.

Interaction of the Expert and the Selection Formula Editor

NOTE: As a general rule, if you can base your record selection on a number of fields (as in our example), select an indexed field instead of a field that is not indexed.

The Select Expert and the Selection Formula Editor are interactive. That is, record selection criteria you enter via the Expert automatically generates a record selection formula that you can review and modify using the *Show Formula* button in the Select Expert or the Record Selection Formula Editor. Likewise, record selection formulas and modifications to existing record selection formulas automatically update the selection criteria in the Select Expert.

Because of this interactivity, you can use the two facilities together as a tutorial for learning the Crystal Reports formula language. To do this:

1. Set up your selection criteria using the Select Records Expert.
2. Click the *Show Formula* button and the Expert expands so you can review the formula the program generated based on your criteria.
3. Click the *Hide Formula* button when you are done with your review.
4. Change your selection formula using the Select Records Expert.
5. Review the updated formula by clicking the *Show Formula* button again.
6. As you gain confidence and want to make changes using the formula language, click the *Formula Editor* button in the expanded Expert and make your formula changes using all the tools in the Record Selection Formula Editor.
7. Review the results of those changes in the Select Expert. Select each field used in the record selection formula and see how the program translates your formula into Expert selection criteria.

NOTE: Selection formula components that do not fit any of the fixed criteria in the Expert will not be translated. For example, if part of your record selection formula extracts the last four characters in a customer number, the section of the formula code

that does that extraction will not be converted to Expert selection criteria. This is because there is no facility in the Expert to make such an extraction by pointing and clicking. In cases like this, the section of code that does not conform will be displayed in the selection text box.

Group Selection

When you group or summarize data, Crystal Reports, by default, includes all the groups in your report. There may be times, however, when you do not want to include all the groups. You may only want to include those groups that have certain group names, or whose summarized values meet a certain condition. Or you might want to see only the groups with the highest summary values, or the lowest. Crystal Reports enables you to select the groups that appear in your report in several different ways.

NOTE: Some kinds of selection you can do using either a record selection formula or a group selection formula. For example, if you have a mailing list grouped by region and your record selection formula specifies only California customers (`{customer.REGION} = "CA"`), your report will have only a single group: California. If you have a group selection formula that specifies only groups with the group name "CA" (`GroupName({customer.REGION}) = "CA"`) and no record selection formula, you will get an identical report, assuming that the California condition was the only selection test in both situations. When using the group selection method, however, it could conceivably take longer to get the report back, as selection pass through will not occur from group selection.

Select Expert

You can select groups of records using the Select Expert just like you can select individual records.

Instead of basing your selection criteria on standard fields like you do for record selection, however, you base it on group name fields or summary fields when you are setting group selection criteria.

- If you have simply grouped your data but you have not summarized it, you can only set up group selection based on the group name field. For example:

GroupName is equal to MA

- If you have summarized your data, you can set up group selection based either on the group name field or on the summary field. For example:

GroupName is equal to MA

or

```
Sum({Customer.Last Year's Sales},  
{Customer.Region}) > 10000
```

NOTE: You can use the Select Expert to set up record selection and group selection requests. When you select either a GroupName or a summary field, the program knows that the selection criteria you set up is intended for group selection. In all other cases, the program knows that you are setting up record selection.

Formula language

You can select groups using the formula language. To do this you call up the Group Selection Formula Editor using the GROUP command from the Report | Edit Selection Formula menu.

When you are in the Formula Editor you can build your group selection request using group fields, group name fields, and other formulas. As was the case with record selection formulas, your only restriction is that the formula you create must be Boolean, that is, it must return either a true or false value. See Chapter 10, *Formulas 101*, Page 249.

Top N

Many times, you might want to show only the top or bottom groups in a report: the fastest selling product lines, the least productive sales regions, the states that generate the most orders. Because this kind of group selection is so popular, the program includes a facility for setting it up easily.

You set up Top N (or Bottom N) group selection using the Top N/Sort Group Expert that you call up using the Report menu. Using this expert, you specify whether you want to display the Top N or Bottom N groups, and then you specify what number N is. For example:

- if you want to report on the three fastest selling product lines, select top N and set N to be equal to three, or

- if you want to report on the five least productive sales regions, select bottom N and set N to be equal to five.

The program will display those groups to your specifications.

But there is one other consideration with Top N group selection and that is what to do with all the records from other groups that do not fit the Top N or Bottom N criteria you set. You need to decide whether to eliminate those records from your report entirely or to lump them altogether in a single group with the name you specify. The program enables you to do either. See *How to select the Top or Bottom N groups*, Page 333.

Record selection formula templates

Formula templates

The following example formulas can be used as templates to help you create your own selection formulas using the Record Selection Formula Editor. The examples illustrate different kinds of selections that you can do, not necessarily selection that is wise to do from a performance standpoint. See *Performance Tips*, Page 325, to help identify the best way to set up your record selection.

NOTE: All of these formulas are available in online Help so you can copy them directly into the Selection Formula Editor. Search for Record selection formula templates in Crystal Reports online Help.

Record selection templates

FOR SELECTING RECORDS USING CHARACTER STRINGS

"C" in {file.FIELD}[1]

«Selects those records in which the value in the {file.FIELD} field begins with the character "C" (includes values like CyclePath, Corp. and Cyclist's Trail Co., excludes values like Bob's Bikes Ltd., and Feel Great Bikes, Inc.).»

not ("C" in {file.FIELD}[1])

«Selects those records in which the value in the {file.FIELD} field does not begin with the character "C" (includes values like Bob's Bikes Ltd., and Feel Great Bikes, Inc., excludes values like CyclePath, Corp. and Cyclist's Trail Co.).»

"999" in {file.FIELD}[3 to 5]

«Selects those records in which the 3rd through 5th digits of the {file.FIELD} field is equal to "999"(includes values like 10999,70999, and 00999, excludes values like 99901 and 19990).»

"Cycle" in {file.FIELD}

«Selects those records in which the value in the {file.FIELD} field contains the string "Cycle" (includes values such as CyclePath Corp. and CycleSporin, Inc., excludes values like Cyclist's Trail Co. and Feel Great Bikes, Inc.).»

FOR SELECTING RECORDS USING NUMBERS

Single values

{file.FIELD} > 99999

«Selects those records that have a value in the {file.FIELD} field greater than 99999.»

{file.FIELD} < 99999

«Selects those records that have a value in the {file.FIELD} field less than 99999.»

Range of values

{file.FIELD} > 11111 and {file.FIELD} < 99999

«Selects those records that have a value in the {file.FIELD} field greater than 11111 but less than 99999 (neither 11111 or 99999 is included in the range of values).»

{file.FIELD} >= 11111 and {file.FIELD} <= 99999

«Selects those records that have a value in the {file.FIELD} field greater than 11111 but less than 99999 (both 11111 and 99999 are included in the range of values).»

FOR SELECTING RECORDS USING DATES

The Month, Day, and Year functions can all be used in examples like the following:

Year ({file.DATE}) < 1996

«Selects those records where the year found in the {file.DATE} field is earlier than 1996.»

```
Year({file.DATE}) >1992 and Year  
({file.DATE}) < 1996
```

«Selects those records where the year found in the {file.DATE} field falls between 1992 and 1996 (1992 and 1996 not included).»

```
Year({file.DATE}) >=1992 and  
Year({file.DATE}) <= 1996
```

«Selects those records where the year found in the {file.DATE} field falls between 1992 and 1996 (1992 and 1999 are included).»

```
Month({file.DATE}) in 1 to 4
```

«Uses the Make Range/In Range operators to select those records in which the month found in the {file.DATE} field is one of the first four months of the year (includes January, February, March, and April).»

```
Month({file.DATE}) in [1,3]
```

«Uses the Make Array/In Array operators to select those records in which the month found in the {file.DATE} field is the first month or the fourth month of the year (includes January and April, excludes February and March).»

SELECTING RECORDS USING PRESET DATE RANGES

You can use Crystal Reports preset date ranges to create selection formulas similar to these:

```
{file.DATE} in LastFullMonth
```

«Selects those records where the date found in the {file.DATE} field falls within the last full month (If the month is May, this selects all records with an April date.).»

```
not({file.DATE} in LastFullMonth)
```

«Selects all records except those in which the date found in the {file.DATE} field falls within the last full month (If the month is May, this selects all records except those with an April date.).»

```
{file.DATE} < Today
```

«Selects all records in which the date found in the {file.DATE} field falls before today's date.»

SELECTING RECORDS USING DATE/NUMBER/ CHARACTER COMBINATIONS

These formulas simply "mix and match" formulas from the categories above.

```
"C" in {file.FIELD}[1] and Month({file.DATE})  
in [1, 4]
```

«Selects those records in which the value in the {file.FIELD} field begins with "C" and the month is either January or April. For example, if you use this kind of formula with an order database, you could be asking for a report showing all customers whose names begin with "C" and who placed orders in January or in April.»

```
"AOK" in {file.HISTORY}[3 to 5] and  
{file.OPENCRED} >= 5000
```

«Selects those records in which the {file.HISTORY} field shows the characters "AOK" as the 3, 4, and 5 characters and the {file.OPENCRED} field (the amount of available credit) is at least 5000.»

You can use these templates as is (with your own data), or combine them to create complex formulas, or you can use the principles illustrated here plus the online Help topics for functions and operators to create powerful selection formulas for yourself.

Selection Performance Tips

There are a number of performance-related items that you should consider when you are setting up your selection requests:

- Record selection will be faster if it is based on indexed fields instead of non-indexed fields.
- If you have based record selection on indexed fields, make sure *Use Indexes for Speed* is turned on in the Report Options dialog box. Search for *Report Options dialog box* in Crystal Reports online Help.

- Avoid performing record selection based on formula fields if at all possible because it will result in less efficient reporting. For example, assume you have a formula field (@ExtendedPrice) in your report that returns the extended price of a line item (Quantity * Price). If you base your selection criteria on that formula (@ExtendedPrice > 1000, for example), the SQL server will not understand the Crystal Reports formula so the program will not pass the selection criteria down to the server. Instead it will retrieve all of the records from the server, and then it will apply record selection on the client machine. This can tie up network resources and slow processing time considerably.

- Try to avoid doing subscript ranges such as:

```
{file.FIELD}[1 to 5]
```

in the selection formula. There is no SQL equivalent so pass through will not occur. Subscripting of only the first character works, but subscripting multiple characters does not. See *Subscript (I)*, Page 628.

- When using SQL/ODBC data sources, if you are unsure if the record selection is passing through to SQL or not, check it using the SHOW SQL QUERY command on the Database menu. If the SQL query does not have a WHERE statement or if the WHERE statement does not mention all of your fields that you are dealing with in your record selection, then you will need to work through the formula again since the translation did not occur properly.
 - Make sure you have logged on to your datasource before choosing the SHOW SQL QUERY command from the Database menu.
 - SQL syntax will change with different drivers (ODBC or SQL) but the majority follow the Oracle SQL model as a guide. Consider the fact that your driver may use slightly different syntax. This is also the case for non-SQL databases.
- Do not do any data type conversions in the record selection formula (for example, converting a number to a string using the ToText function). Such conversions cannot be translated to SQL so SQL pass through will not occur. See *ToText*, Page 601.

Doing record selection with a group selection formula

If you need to do record selection on indexed and non-indexed fields, you can always set up your record selection in two steps to maximize performance. You do this by creating a record selection formula and a group selection formula, and using them both to select records. A little explanation is called for here.

The Formula Editor that you use for creating group selection formulas has the same functionality as the Editor you use for record selection formulas. While its primary use is for setting up group selection, it can be used to set up record selection as well.

While the two Editors are fundamentally the same and the formulas they produce look the same, they each produce formulas that are evaluated at different times.

- The formulas from the Record Selection Formula Editor get evaluated as the program reads records.
- The formulas from the Group Selection Formula Editor get evaluated as the program is printing records. At this point, the only records that are saved with the report are those that passed record selection criteria.

Using this functionality:

- You set up record selection based exclusively on indexed fields in a record selection formula.
- You set up record selection based exclusively on non-indexed fields in a group selection formula.

Since the program runs record selection when it reads records and it runs group selection when it prints records, the following events occur:

- The record selection formula on the indexed fields quickly returns a subset of data from your database. For this example, lets say that it returns 5000 records out of 100,000 and saves them in a buffer.
- The group selection formula performs record selection but only on the subset of data (5000) records that is saved with the report.

You accomplish the same record selection but do it in a more efficient manner. With really big databases, this technique can save you significant processing time.

Related Topics

See Chapter 10, *Formulas 101*, Page 249

See Chapter 11, *Advanced Formulas*, Page 269

See Chapter 24, *Functions*, Page 569

See Chapter 25, *Operators and Variables*, Page 609

HANDS-ON

How to set up record selection using the Expert

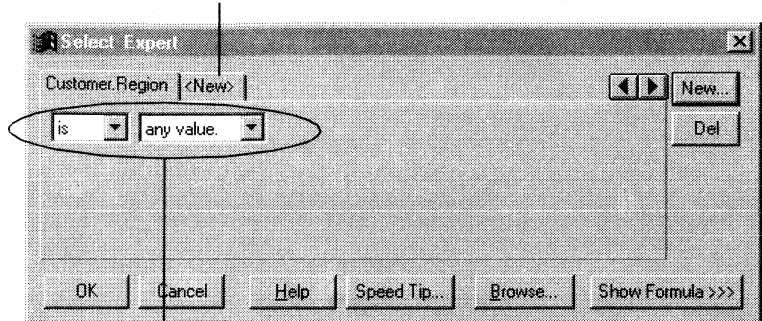
When you work with the Select Expert, you select the field you want to apply selection conditions to and then you specify the conditions.

1. You can begin your work in the Select Expert in one of two ways:



- Highlight the field in your report on which you want to base record selection and then click the **SELECT EXPERT** button. The program opens the Select Expert, ready for you to set the conditions on the highlighted field.
- Without highlighting a report field, click the **SELECT EXPERT** button. The Choose Field dialog box appears with indexed fields identified with colored arrowheads. Select the field you want to base your selection on (preferably an indexed field) and click **OK**. The Select Expert appears. See *Selection Performance Tips*, Page 325.

If you want to base your record selection on more than one field, click the New Tab, choose your next field from the Choose Field dialog box when it appears



Use the drop down boxes to enter your selection criteria for the indicated field.

2. When you are finished, click *OK* in the *Select Expert* to return to the tab you were using when you called up the Expert.

Crystal Reports will generate a selection formula based on your specifications and limit the report to the records you have specified.

NOTE: To view or edit the selection formula generated by Crystal Reports, click the *Show Formula* button. The *Select Expert* expands to show the formula. If you want to use the tools in the *Formula Editor* to modify the formula, click the *Formula Editor* button. The formula will appear in the *Formula Editor*.

How to set up group selection using the Expert

When you work with the *Select Expert*, you select the group name or summary field you want to apply selection conditions to and then you specify the conditions.

1. You can begin your work in the *Select Expert* in one of two ways:



- In your report, highlight the group name or summary field on which you want to base group selection and then click the *SELECT EXPERT* button. The program opens the *Select Expert*, ready for you to set the conditions on the highlighted group field.

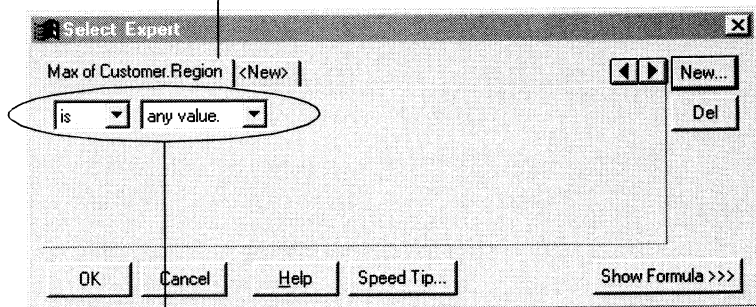
NOTE: Refer to the status bar if you are unsure whether or not a field you have highlighted is a group field.

- Without highlighting a group field in the report, click the SELECT EXPERT button. The Choose Field dialog box appears. Select the group field you want to base your selection on and click OK. The Select Expert appears.

NOTE: Summary fields identify the location of the summary value, the field that triggers a grouping when its value changes, the kind of summary, and the field being summarized, and they look similar to this in the Choose Field dialog box:

Group Footer #1: Customer.Region
Sum of Last Year's Sales

If you want to base your record selection on more than one field, click the New Tab, choose your next field from the Choose Field dialog box when it appears



Use the drop down boxes to enter your selection criteria for the indicated field.

NOTE: If you have not already previewed the report or refreshed the data, there is no data yet saved with the report. Without the data, the program can not calculate group values, thus no values appear when you click the arrow in the right drop down box. In this situation, you will have to type in the values you want. If you want real values to work with, you will need to Preview your report first. This will calculate the actual summary values for you to work with.

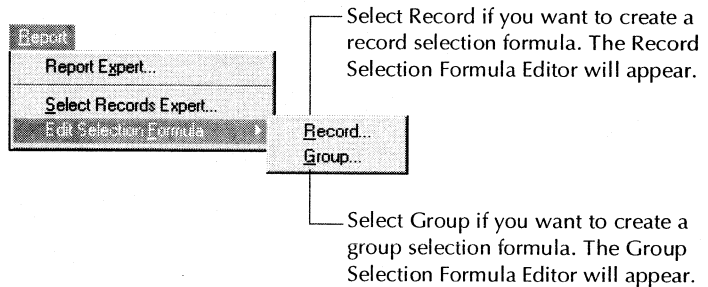
2. When you are finished, click OK in the Select Expert to return to the tab you were using when you called up the Expert.

How to create a record or group selection formula

In order to create a selection formula, you need to have some understanding of the Crystal Reports formula language and the use of the Formula Editors. For a thorough discussion of those tools, see Chapter 10, Formulas 101, Page 249, and Chapter 11, Advanced Formulas, Page 269.

To create a selection formula:

1. Select the EDIT SELECTION FORMULA command from the Report menu. A submenu appears.

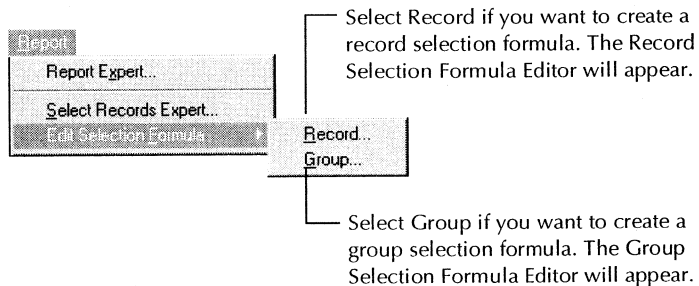


2. Enter your selection formula as you would any other formula. Since this is a selection formula, however, the formula must be Boolean (return either a True or False value).
3. Click OK when finished.
4. If the Formula Editor reports errors, debug the formula as necessary and click OK again. See *How to debug a formula*, Page 287.

Now, when the program runs the report it will include only those records or groups of records that you specify. See *How to set record selection using parameter fields*, Page 309.

How to use record/group selection templates

1. Select the template you want to use. You can do this in either of two ways:
 - Review the list in *Record/Group Selection Formula Templates*, Page 322, and write down the formula of interest, or
 - Find the *Record/Group Selection Templates* topic in Crystal Reports online Help and copy the formula of interest to the Clipboard.
2. Choose the EDIT SELECTION FORMULA command from the Report menu. A submenu appears.



3. In the *Formula Text* box, type in the formula you wrote down in Step 1 or paste it from the Clipboard.
4. Replace the values (fields, text, etc.) in the formula with the values you want. For example, if the example formula is:

```
{file.FIELD} > 99999
```

and you want to limit your report to records that have a value in the {Orders Detail.QUANTITY} field greater than 25, simply replace the existing values with the values you want so your selection formula reads:

```
{Orders Detail.QUANTITY} > 25
```

NOTE: You do not have to type in the old formula first and then modify it. If you are comfortable with doing so, you can modify the formula on paper or in your head and then enter the modified formula. If you do this, make sure to check your final formula against the example formula to make sure you have not left anything out or added anything that should not be there.

5. Click *Accept* when finished to accept the selection formula and return to the tab you were working on when you called up the Selection Formula Editor.

How to select the top or bottom N groups

To select top N or bottom N groups:

- choose whether you want to show the top groups or the bottom,
 - specify how many groups to show, and
 - tell the program how to deal with the records that are not in one of the selected groups.
1. Create your report and summarize the data as you want it. When you summarize the data, the program breaks the data into groups and summarizes each group. With Top N grouping, you are instructing the program to display those groups that have the highest summary values (Top N) or lowest summary values (Bottom N).
 2. Choose the TOP N/SORT GROUP EXPERT command from the Report menu. The Top N/Sort Group Expert appears with a tab for your group active.
 - If you have multiple group sections, the program will display a tab for each of the groups.
 - Select the tab for the group you want to base your selection on.

3. Choose Top N (if you want your report to contain the groups with the highest summaries) or Bottom N (if you want to report on those groups with the lowest summaries).

Top N/Sort Group Expert

Analyse report results by taking the Top N or Sort of totals.

Customer.Customer Name Customer.Region

Top N of Sum of Customer.Last Year's Sales

where N is: 5

include Others, with the name: Others

OK Cancel Help

4. Choose the summary that you want to base your selection on.

NOTE: The summary drop down box (on the right) is for those cases in which you have multiple summaries in a single group section. For example, in an orders report, you may sum the orders for each customer and average the orders too and display both the sum and the average in the same group section. In such a case you would select the sum or the average from this drop down box. If you have only a single summary, the summary will be displayed and you will not have to make a selection.

5. In the *where N is* box, enter the number of groups you want to display. For example, if you want to display the top five groups, type 5 in this box.
6. All that is left is deciding what you want to do with all those records that do not fit into the selected groups.
 - toggle the *include Others, with the name* check box off if you want the other records excluded from the report
 - toggle the check box on and name the group if you want to lump all the other records into a single group.
7. Click *OK* when you are finished. Now, when the program runs the report, it will include only those groups that you specify.

15

Subreports

What you will find in this chapter...

What are subreports? 336

How to insert a subreport 340

How to link a subreport to the data in the primary report 342

How to combine unrelated reports using subreports 344

How to use subreports with "unlinkable" data 347

How to show different views of the same data in one report 349

What are subreports?

A subreport is a report within a report. You create it in much the same way as you create a regular report. A subreport has most of the characteristics of a report, and it can have its own record selection criteria. The only differences between a subreport and a primary report are that a subreport:

- is inserted as an object in a primary report; it cannot stand on its own,
- can be placed in any report section and the entire subreport will print in that section, and
- cannot itself contain a subreport.

There are three typical times that you would want to use a subreport:

- When you want to combine unrelated reports into a single report.
- When you want to coordinate data that can not be otherwise linked.
- When you want to present different views of the same data in a single report.
- When you are doing one to many lookups from a field that is not indexed on the lookup field.

Each of these is discussed in depth in later sections.

Unlinked versus linked

Unlinked subreports are freestanding; their data is not coordinated with the data in the primary report in any way.

- This does not mean that an unlinked subreport has to use the same data as the primary report; it can use the same data source or a different data source entirely.
- It also does not mean that the subreport is limited to reporting on a single table. A linked subreport can be based on single table or on multiple tables.

What it means is that there is no attempt to match up the records in one report with the records in the other. Regardless of the underlying data sources, the reports are treated as unrelated.

Linked subreports are just the opposite. Their data is coordinated. The program matches up the records in the subreport with the records in the primary report. If you create a primary report with customer information and a subreport with order information and link them, the program creates a subreport for each customer and includes in that subreport all the orders for the customer.

HOW SUBREPORT LINKING WORKS

When you link a subreport to a primary report, the program creates the link through the mechanism of a parameter field. See Chapter 13, *Parameter (prompting) Fields*, Page 305.

When you select a subreport link field:

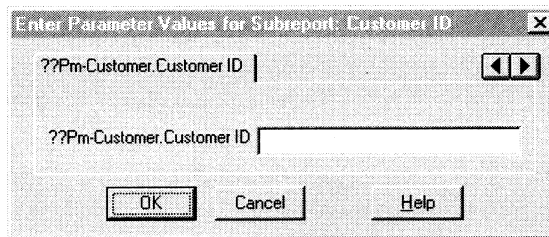
- the program creates a parameter field in the subreport that is used to retrieve values passed to it by the primary report.
- the program also creates a record selection formula for the subreport using the parameter field.
 - the selection formula limits the subreport to those records in which the link field is equal to the parameter field value.

When you run the report, the program finds the first primary field record it needs and passes the value in the link field to the parameter field in the subreport. The program then creates the subreport with record selection based on the parameter field value. Here is an example:

1. You create a report that shows customer data and a subreport that shows order data and you link the two reports using the Customer ID field.
2. When you run the report, the program finds the first customer record it needs and passes the Customer ID value from that record to the subreport parameter field.
3. The program runs the Orders subreport. Since the subreport selection formula selects only those records in which the Customer ID value is equal to the parameter field value, and since that parameter field value is equal to the Customer ID in the first record in the primary report, the subreport contains only those records that have the same customer ID, namely, those records that are orders for the first customer.

4. When the subreport finishes, the program goes to the second record it needs in the primary report, prints the customer data, and then passes this customer's ID number to the parameter field.
5. The program then runs a subreport including only those order records for the second customer.
6. The process continues until the report is finished.
7. All of this parameter field manipulation takes place behind the scenes. You simply pick the fields you want to use to link the primary report with the subreport and the program does the rest. The values are passed without the parameter field prompting you for a value.

NOTE: *If you have a linked subreport and you click the PRINT PREVIEW button from the Subreport Design Tab, the program runs the subreport on its own, without first receiving a parameter field value from the primary report. In this case, the program displays the Enter Parameter Values for Subreport dialog box prompting you for a value.*



Type in a value and the program runs the subreport using that value.

NOTE: *Since the program creates a parameter field, that parameter field is available on the Fields list in the Formula Editors for any other needs you may have for it.*

Database links versus subreports in one-to-many situations

When two tables in your report have a one-to-many relationship, the program retrieves the data in different ways depending on

- the data source,
- the index situation,
- the record selection criteria, and
- whether you are creating a single report based on linked tables or a primary report that contains a subreport.

When you are considering whether to use linked tables or a subreport, you need to understand the ramifications of doing it each way. These issues are discussed fully in the *Performance considerations in one-to-many links* section of Chapter 19, *Working With Databases*, Page 403.

As a general rule,

- if you have indexed tables,
- if you are linking on the indexed fields, and
- if you have range limiting record selection criteria based on the indexed fields,

the program needs to read the same number of records whether you are linking tables in a single report or using subreports. Since each subreport is run as a separate report, however, there may be a performance advantage using linked tables.

HANDS-ON

The first two topics in this section (*How to insert a subreport* and *How to link a subreport to the data in the primary report*) explain in detail how to perform the two basic subreporting tasks. You will do one or both of these tasks every time you set up a subreport.

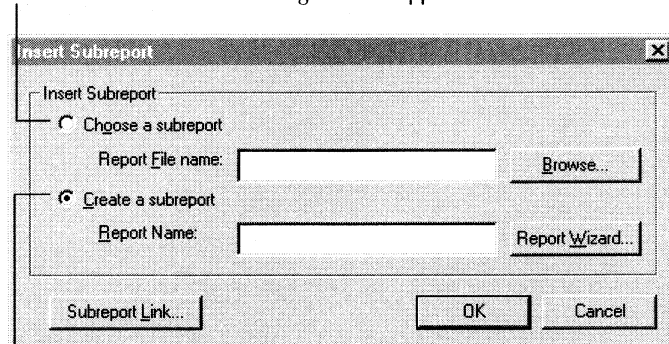
The remaining topics all deal with subreport specifics: how to create specific kinds of reports using subreports. These are explained in broad terms (create a subreport, link these two fields, and so forth). Refer back to the first two topics for the specifics of performing those tasks.

How to insert a subreport



1. Click the SUBREPORT button on the toolbar. The Insert Subreport dialog box appears.

To choose an existing subreport, check this options button and type the name into the text box. If you do not know the name, click the Browse button and locate it in the dialog box that appears.

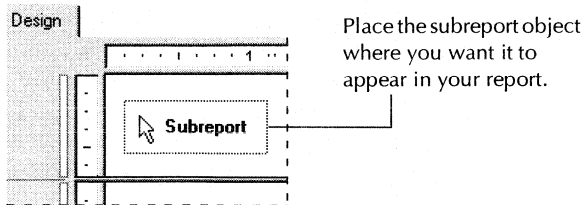


To create a new subreport, check this options button and type a name into the text box. If you would like assistance in creating the subreport, click the Report Wizard button.

- If you want to insert a report that you have already created and saved in a file, click the *Choose a subreport* button.
 - In the Report File name box, enter the name and path of the report, or click *Browse* and choose the report from the Open dialog box when it appears.
 - Click *OK* when finished to return to the Insert Subreport dialog box.
- If you want to create a new report, click the *Create a subreport* button.
 - Enter a name for the report.
 - Click the *Report Expert* button.
 - Follow the instructions on each of the Expert tabs to create the report you are going to use as a subreport.
 - Click *OK* when you are finished to return to the Insert Subreport dialog box.

When you click *OK* in the Insert Subreport dialog box, the program displays a rectangular placement frame.

2. When the object frame appears, move it where you want it to appear in your report and click to place it.

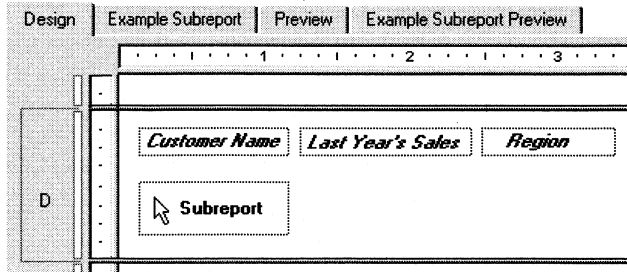


- If you imported your subreport, the program creates a subreport Design Tab labelled with the subreport name.
 - If you do not need to edit your report, you are finished.
 - If you want to edit it, click the Subreport Design Tab and make your modifications.
 - To see the results of your modifications, click the **PRINT PREVIEW** button.
 - If you are creating a new report, the program creates a Subreport Design Tab labeled with the subreport name.
 - If you want to do more to your report than you did in the Expert, click the Subreport Design Tab and finish your subreport as you would any other report.
 - To see your finished subreport, click the **PRINT PREVIEW** button. The program creates a Subreport Preview Tab to display your work.
 - If you do not want to do more to your report, you are finished
3. To see your parent report with the subreport in place, click the Preview Tab for the primary report.



Click the Preview Tab to see your parent report.

Click the Subreport Preview Tab to see your subreport.



How to link a subreport to the data in the primary report

Many times the data in a subreport supplements the data in the primary report. You might, for example, have customer data in a primary report and use subreports to show the orders for each customer.

Primary Report Data	Bike-O-Rama Corporation	12/2/94	\$41.90
	7464 St. Georges Way	12/8/94	\$3,520.30
	Sterling Heights, MI 48358	1/6/95	\$62.33
		2/16/95	\$6,682.97
	Linked Subreport Data	6/14/95	\$0.00
		9/1/95	\$4,078.95
		2/2/96	\$931.05
The Peddallers Inc. 410 Eighth Avenue DeKalb, IL 60148	12/11/94	\$764.85	
	12/24/94	\$42.00	
	1/30/95	\$3,884.25	
	3/1/95	\$1,515.35	
	4/9/96	\$5,549.40	

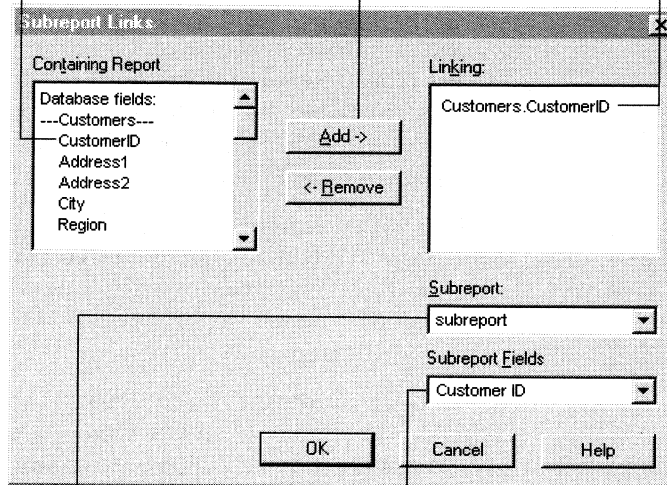
In such a case, you will need to coordinate the data in the primary report and subreport so the orders in each subreport match up with the correct customer.

To do this, you need to specify a field in the subreport and one in the primary report that contain common data. Crystal Reports uses these companion fields to coordinate the data. You coordinate the data in the subreport with the data in the primary report using the Subreport Links dialog box. You get to that dialog box in one of two ways:

- If you are importing a report as a subreport or creating one from scratch, you get to it by clicking the *Subreport Link* button in the Insert Subreport dialog box. Thus you can build or import your subreport and link it to the primary report in one coordinated process.
- If you already have a subreport in your primary report and you did not link it when you were setting it up, you can get to the Subreport Links dialog box by choosing SUBREPORT LINKS on the Edit menu.

Once you are in the Subreport Links dialog box, follow this procedure:

1. Highlight the field you want to use as the link field in the primary (containing) report.
2. Click the Add button.
3. The field will be added to the Linking list box, thus selecting it as a link field.



4. Choose the subreport you want to link from this drop down list (if it is not already selected).
5. Choose the subreport field you want to use to link to the containing report from this drop down list.
6. Repeat Steps 1 and 2 as many times as necessary if you want to set up the link using more than one field.
7. Click OK when finished. Now, when you run the report, the program will coordinate the data in the primary report and the subreport.

How to combine unrelated reports using subreports

There may be times when you want to combine unrelated reports into a single report. For example, you may want to create a single report that presents:

- Sales by sales representative
- Sales by item

While both reports deal with sales data, there is no real linear relationship between the reports.

Sales Summaries

Employee Sales	Employee Name	Employee Sales
	Nancy Davolio	\$58,872.03
	Andrew Fuller	\$24,972.18
	Janet Leverling	\$71,245.08
	Margaret Peacock	\$89,799.32

Product Sales	Product Name	Product Sales
	Lycra Gloves	\$242,362.87
	Craze Mtn Lock	\$44,305.17
	Craze Adult Helmet	\$249,127.99
	Micro Nicros	\$63,682.39
	Rapel	\$441,033.88
	Slick Rock	\$423,605.05

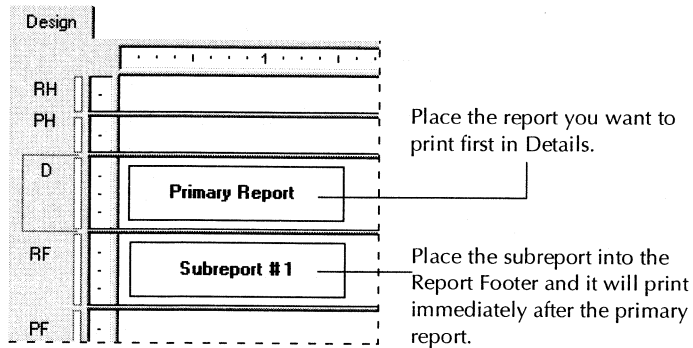
You can combine unrelated reports into a single report like this using subreports. While the reports could be based on the same data set, they do not have to be. They could each be based on entirely different data sets.

Each of these reports is freestanding; the data in any of the reports is not linked in any way to the data in any of the other reports. This is the easiest of the subreport options to work with.

TWO UNRELATED REPORTS

If you want your report to consist entirely of two unrelated reports:

- Create the report you want to print first as the primary report.
- Import an existing report for use as a subreport or create one.
- Place the subreport in the Report Footer section of the primary report. The primary report will print and the subreport will print immediately after the primary report. See *Area printing characteristics*, Page 62.

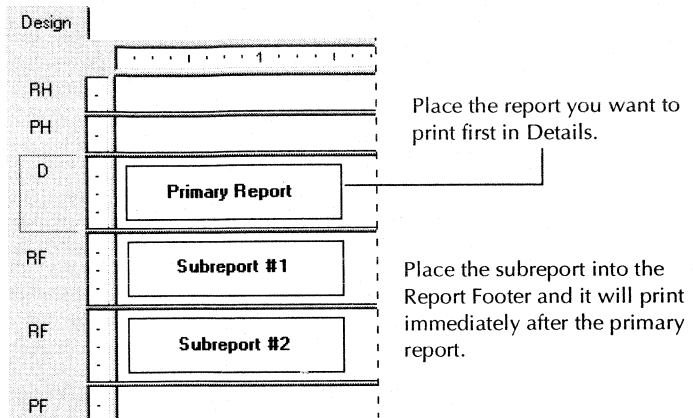


THREE OR MORE UNRELATED REPORTS

If you want your report to consist of three or more unrelated reports:

- Create the report you want to print first as the primary report.
- Import or create each of the other reports you want to use as subreports.
- Using the Section Expert, insert enough Report Footer sections to match the number of subreports that you are using. For example, if you want to use three subreports, insert two new Report Footer sections so you have a total of three Report Footer sections.
- Place the subreport you want to print immediately after the primary report in Report Footer A, the subreport you want to print next in Report Footer B, and so forth. The primary report will print and the subreports will print in the order you placed them on your report.

NOTE: You could also place the subreports side by side in the same Report Footer section.



See *Area printing characteristics*, Page 62, and *How to add, delete, move, and merge sections*, Page 83.

How to use subreports with "unlinkable" data

You can link tables in a report as long as three criteria are met:

- the link fields are both database fields,
- the link fields contain similar data,
- the link fields are the same length, and
- the link field in the *link to* (lookup) table is indexed (PC databases only).

Rarely is linking tables a problem.

There are some circumstances, however, where you cannot coordinate data in different tables because your situation does not fit the linking criteria. If you want to link to or from a formula field, or you want to link two unindexed tables, you can not do it in a single report. You have to use subreports.

Linking to/from a formula field

There are situations where you may need to link to or from a calculated field. For example, an employee ID could be an 11 character value that consists of a two character department code followed by the employee's nine character Social Security number (for example, HR555347487). Using the Crystal Reports formula

language, it is easy to extract the Social Security number from this field:

```
{employee.EMPLOYEE ID} [-9 to -1]
```

or

```
{employee.EMPLOYEE ID} [3 to 12]
```

For the value HR555347487, either formula would return the value 555347487.

While the return value is a valid Social Security number, the fact that it comes from a formula prevents you from using the field to link to a Social Security number field in another table. You can report on and coordinate the values in the two tables, however, using a subreport. (See *Subscript ([])*, Page 628.)

LINKING TO A FORMULA FIELD

1. Create the primary report using a table that includes the Social Security Number field (for our example, {file.SSN}).
2. Create (or import) a subreport using the formula that extracts the Social Security Number from the Employee ID field (for our example, {@EXTRACT}). See *How to insert a subreport*, Page 340.
3. Insert the subreport where you want it to appear in the primary report.
4. Link the subreport to the primary report, linking the Social Security Number field in the primary report ({file.SSN}) with the formula that extracts the number in the subreport (@EXTRACT). See *How to link a subreport to the data in the primary report*, Page 342.

LINKING FROM A FORMULA FIELD

1. Create the primary report using the formula that extracts the Social Security Number from the Employee ID field (for our example, {@EXTRACT}).
2. Create (or import) a subreport using a table that includes the Social Security Number field (for our example, {file.SSN}). See *How to insert a subreport*, Page 340.
3. Insert the subreport where you want it to appear in the primary report.

4. Link the subreport to the primary report, linking the formula that extracts the Social Security Number in the primary report ({@EXTRACT}) with the Social Security Number field in the subreport ({file.SSN}). See *How to link a subreport to the data in the primary report*, Page 342.

LINKING UNINDEXED TABLES

When using PC databases (not SQL or ODBC), the link field in the lookup database needs to be indexed to create a valid link in Crystal Reports. When two tables contain related data yet neither is indexed on the field you want to use as a link field, or if the primary table is indexed but the lookup table is not, you can not link the tables in a single report. You must use subreports if you want to coordinate the data in both tables.

NOTE: It is important to note that linking unindexed tables or linking from an indexed primary table to an unindexed lookup table makes for very inefficient reporting. If your data set is large, expect this kind of report to take a considerable time to run. Use this kind of subreporting only if you do not have any other options.

1. Create your primary report.
2. Create (or import) the subreport and insert it into the primary report. See *How to insert a subreport*, Page 340.
3. Link the subreport to the primary report using the unindexed fields (or the indexed field in the primary table and the unindexed field in the lookup table). See *How to link a subreport to the data in the primary report*, Page 342.

How to show different views of the same data in one report

You can use subreports to provide a different view of the same data as the primary report. For example, assume you want to show summary values at the top of your report and details at the bottom, like this:

Sales Summaries		
Employee Sales		
Employee Name		Employee Sales
Nancy Davolio		\$58,872.03
Andrew Fuller		\$24,972.18
Janet Leverling		\$71,245.08
Margaret Peacock		\$89,799.32
Product Sales		
Product Name		Product Sales
Lycra Gloves		\$242,362.87
Craze Mtn Lock		\$44,305.17
Craze Adult Helmet		\$249,127.99
Micro Nicros		\$63,682.39
Rapel		\$441,033.88
Slick Rock		\$423,605.05

You can do this in different ways. The two easiest ways are:

- You can create the summary report as the primary report and the details report as the subreport. If you do this, you can place the details subreport in the Report Footer section.
- You can create the summary report as the subreport and the details report as the primary report. If you do this, you can place the summary report in the Report Header section.

Coordinate the data in the two reports by linking the report using the appropriate link fields.

16

Cross-Tab Objects

What you will find in this chapter...

Cross-Tab Overview 352

Cross-Tab capabilities in Crystal Reports 357

Cross-Tab Overview

A Cross-Tab is an object that summarizes data and then presents the summaries in a compact row and column format that makes it easy to make comparisons and identify trends.

The following examples are provided to demonstrate the power of a Cross-Tab in those situations in which the Cross-Tab is an option. Those are often situations in which the word *by* is included in your report description:

- sales *by* state,
- products sold *by* color and size,
- orders *by* customer,

While there are certainly many ways to create these kinds of reports, Cross-Tabs generally present more data in a more compact, easier to understand form than other reporting methods.

NOTE: With this version of Crystal Reports, you can insert as many Cross-Tab objects in your report as you need. You can even place them in subreports. Thus, you can use Cross-Tabs to show summarized data in a report that presents the details in another form.

In the following examples, the goal is to analyze the unit sales of five different bicycle locks in four different regions (a unit sales of locks *by* region report). For clarity we have included only the most essential information in these reports, that is, the region the order came from, the name of the lock, and the quantity ordered.

The first way of looking at the data is in the most basic of all reports, a columnar report with no grouping or sorting.

**REPORT OF
ORDER DATA -
NO SORTING/
GROUPING**

Region	Product Name	Quantity
AL	Guardian Chain Lock	1
AL	Guardian ATB Lock	3
CA	Guardian "U" Lock	2
CA	Guardian ATB Lock	2
CA	Guardian Chain Lock	1
CA	Guardian Chain Lock	1
CA	Guardian XL "U" Lock	3
FL	Guardian Chain Lock	2
FL	Guardian Mini Lock	1
BC	Guardian Mini Lock	3
AL	Guardian Mini Lock	3
AL	Guardian Chain Lock	2
CA	Guardian XL "U" Lock	2
CA	Guardian Chain Lock	2
CA	Guardian Chain Lock	3
CA	Guardian "U" Lock	2
BC	Guardian "U" Lock	2

This report presents nothing but details. Each row represents an individual order. There are many orders from each of the regions for different locks. But because there is no summary information, it is nearly impossible to get any useful information out of a report like this.

The logical next step is to group the data in some way. You could group it by region, or by product line. We will take a look at both of those options.

This report uses the data seen in the first report, but here the data is grouped by region. All the orders for each region are grouped together, but each regional group contains orders for different kinds of locks. And because the groups contain different kinds of data, summarizing the quantity field would determine the total number of locks sold per region, but not the total of each kind.

**REPORT OF
ORDER DATA -
GROUPED BY
REGION**

Each region group contains orders for different kinds of locks.

Region	Product Name	Quantity
AL	Guardian Chain Lock	1
AL	Guardian ATB Lock	3
AL	Guardian Mini Lock	3
AL	Guardian Chain Lock	2
AL	Guardian "U" Lock	2
AL	Guardian ATB Lock	2
AL	Guardian Chain Lock	1
BC	Guardian Mini Lock	3
BC	Guardian "U" Lock	2
BC	Guardian "U" Lock	2
BC	Guardian Mini Lock	3
BC	Guardian Chain Lock	1

This report groups the data by product. Each group displays all the orders for a specific product. At first it appears that this might be useful, but then it becomes clear that each product group includes orders from several different regions. The information is helpful, and it brings us closer to our goal, but we are still a long way from having the information we need.

**REPORT OF
ORDER DATA -
GROUPED BY
PRODUCT**

Each product group contains orders for many different regions.

Region	Product Name	Quantity
CA	Guardian "U" Lock	2
CA	Guardian "U" Lock	2
BC	Guardian "U" Lock	2
BC	Guardian "U" Lock	2
AL	Guardian "U" Lock	2
AL	Guardian ATB Lock	3
CA	Guardian ATB Lock	2
AL	Guardian ATB Lock	2
AL	Guardian Chain Lock	1
CA	Guardian Chain Lock	1
CA	Guardian Chain Lock	1

This report is the logical next step. If the by region report contained multiple products in each region group, and the by product report contained multiple regions in each product group, then it seems to make sense to combine the two. Doing that, we group first by region and then by product.

**REPORT OF
ORDER DATA -
GROUPED BY
REGION AND
PRODUCT**

Each group contains orders for one product for one region.

<u>Region</u>	<u>Product Name</u>	<u>Quantity</u>
AL	Guardian ATB Lock	3
AL	Guardian ATB Lock	2
AL	Guardian Chain Lock	1
AL	Guardian Chain Lock	2
AL	Guardian Chain Lock	1
AL	Guardian Mini Lock	3
BC	Guardian "U" Lock	2
BC	Guardian "U" Lock	2

Now each group contains unit sales information for one product in one region. But the data is all spread out and remains difficult to analyze. The information is useful, and with a little work you can use a report like this to get the comparison information you need. But a Cross-Tab offers a better solution.

ORDER DATA IN A CROSS-TAB OBJECT

		Regions				
		AL	BC	CA	FL	Total
Product names	Guardian "U" Lock	2	4	4	0	10
	Guardian ATB Lock	5	0	2	0	7
	Guardian Chain Lock	4	1	7	2	14
	Guardian Mini Lock	3	6	0	1	10
	Guardian XL "U" Lock	0	0	5	0	5
Total - one product in one region.	Total	14	11	18	3	46
		Total - all products in one region.	Total - one product in all regions.	Total - all products in all regions.		

Here is all the information you need provided in a compact format. The report shows what products were sold in which regions and what the unit sales were. It is easy to see, for example, that Guardian Mini Locks are not popular at all in California but they are the biggest seller in BC. Or that Florida is being outsold by Alabama in every lock category even though it has roughly two and one half times the population of Alabama.

In this Cross-Tab:

- Product names make up the row headings,
- Regions make up the column headings,
- The value at each row / column intersection is the sum of all the orders for a particular product from a particular region, for example, the total number of Guardian Mini Locks ordered in BC.
- The total at the end of each row is the total of all of the purchases for one product in all regions, for example, the total number of Guardian ATB Locks ordered in Alabama, British Columbia, California, and Florida combined.

- The total at the bottom of each column is the total number of all kinds of locks ordered in one region, for example, the number of locks of all kinds purchased in California.
- The total in the bottom right corner is the grand total showing the total unit sales of all five locks in all four regions.

The report is compact, and you can compare your customers' purchasing habits in a hurry. Clearly this is a worthwhile report in situations such as this.

Cross-Tab capabilities in Crystal Reports

Crystal Reports has powerful Cross-Tab capabilities.

- You can create Cross-Tabs using the Cross-Tab expert or as custom Cross-Tab objects.
- You can include multiple summary fields or calculations in your Cross-Tabs. For example, for a single item you can summarize both quantity and price information, or you could show the sum of the orders and the size of the average order.
- You can insert as many Cross-Tab objects as you need, and you can even insert Cross-Tab objects in subreports.
- You can use first pass formulas that are defined elsewhere on the report in Cross-Tabs.
- You can print Cross-Tabs that extend beyond the width of a page.
- You can independently format rows and columns with background colors, borders, and fonts for better looking and easier-to-understand Cross-Tabs.
- You can create Cross-Tabs that are longer than a single page and the program will automatically repeat the column headings at the top of each page.
- You can design and edit Cross-Tabs easily with the ability to call up the Cross-Tab dialog box from both the Design and Preview Tabs.

Using these capabilities, you can create sophisticated Cross-Tabs like this with ease:

Design Preview

<div style="display: flex; flex-direction: column; align-items: flex-start;"> <div style="display: flex; align-items: center; margin-bottom: 5px;"> Units </div> <div style="display: flex; align-items: center;"> Amount </div> </div>		Craze Adult Helmet			
		small/medium			
		white	green	red	Totals
Western Region	CA	3628	2713	2035	8376
		\$122,989.20	\$91,970.70	\$68,986.50	\$263,946.40
	WA	1436	934	1123	3493
	\$48,680.40	\$31,662.60	\$38,069.70	\$118,412.70	
Totals	5064	3647	3158	11869	
	\$171,669.60	\$123,633.30	\$107,056.20	\$402,359.10	

For complete details on creating and formatting Cross-Tabs, search for *Cross-Tab topics* in Crystal Reports online Help.

17

Queries

What you will find in this chapter...

- The Crystal Query Designer 360
- How to use an SQL query that you designed elsewhere 362
- How to create a new query 364
- How to add tables and fields 365
- How to link tables and specify a join type 367
- How to summarize data with aggregate functions 370
- How to sort records according to field values 372
- How to specify records to be included 373
- How to identify unique values 369
- How to select groups to be included 374
- How to create an SQL expression 377
- How to select a Crystal Query for a report 381

The Crystal Query Designer

A query is simply a request for specific information from a database. If you are requesting that information from an SQL database (or from a database that you access via ODBC), your query must be written using the Structured Query Language (SQL). The SQL language is not difficult to learn, but mastering the fine points of creating and retrieving data using SQL can take years. Since the Crystal Query Designer eliminates the need to understand SQL, it can get you building effective queries right away.

The Crystal Query Designer has been designed to meet the needs of individuals with little or no query background as well as the needs of experienced SQL professionals.

- If you are new to querying, you will enjoy the way the Query Designer helps you create queries, even if you have no knowledge of SQL whatsoever. By answering a few questions on a set of sequential tabs, you give the program all the information to generate a query that fits your needs.
- If you are an SQL professional, you will appreciate the facility that enables you to fine tune the queries that the Query Designer generates. If you are more comfortable writing your own SQL queries, you will find it easy to enter your queries directly or even paste them in from another source.

The Crystal Query Designer can be a powerful tool for many of your information gathering needs.

NOTE: The Query Designer can only access data stored in an ODBC data source. Any SQL or other type of database mentioned in this chapter must be accessed through an ODBC data source.

Why Use A Query?

Crystal Query files provide a means of off-loading much of the data generating tasks normally performed by Crystal Reports on to a SQL or other database server. The set of data returned represents a subset of the actual data in the database, but it is the data that you most need.

If you are an experienced SQL programmer, Crystal Query allows you to reuse your existing queries, quickly and easily. All of your

existing work becomes convenient query files that can be used to design reports with Crystal Reports.

In addition, a Crystal Query file provides full ANSI SQL compatibility. Although Crystal Reports supports powerful SQL pass-through reporting, it does not support the full SQL language or allow you to edit every part of a SQL statement. By using the Query Designer to create data sets to base your reports on, you get all of the power of SQL, including complex joins, sorts, and aggregate functions.

NOTE: Many of the sections in this chapter refer to specific SQL clauses. If you are experienced with SQL, use this information to better understand how the Query Expert generates a SQL statement. If you are new to SQL, you can ignore this information and continue to use the Query Expert without losing any of its power.

Using the Query Designer

The Query Designer has two primary uses:

1. Designing and developing data sets for building reports in Crystal Reports, and
2. Retrieving and analyzing current information on an "as needed" basis to facilitate informed decision making.

CRYSTAL REPORTS AND QUERY DATA SETS

Crystal Reports allows you to design a report based on a query data set rather than tables and fields. Since the query contains a predefined set of data, the tables and fields necessary for the report are already included.

The set of data produced by the SQL query works just like a database table when you design your report. The name of the query, along with fields it accesses, appears in the Insert Field dialog box. Aggregate functions and SQL expressions act like fields, providing data values corresponding to each record in the SQL query.

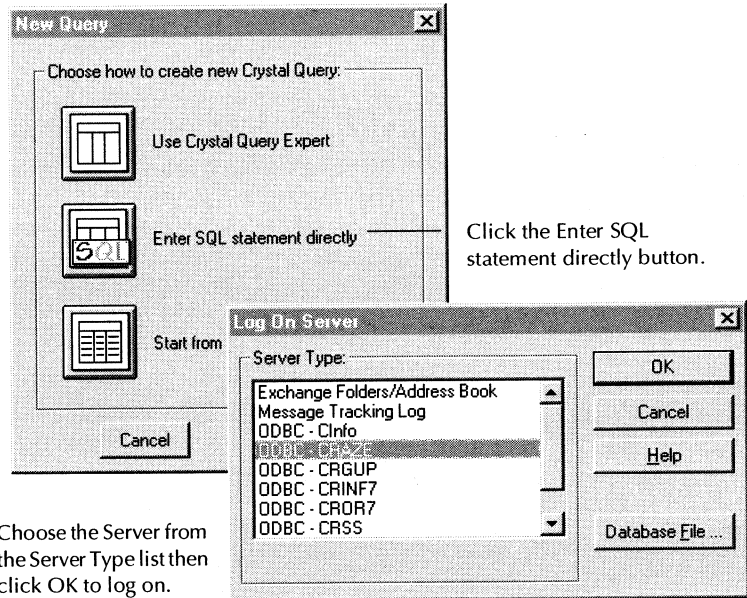
DATA ANALYSIS AND DECISION MAKING

You may not always need to produce finished reports based on your data; sometimes you just need numbers in a hurry (to prepare for a meeting, to help you make projections, etc.). The Query Designer makes it easy for you to get the information you need.

Sometimes you may need to retrieve the same data on a recurring basis (weekly, every month end, etc.). By setting up and saving a single query, you can retrieve updated data quickly, whenever you need it, and with minimal effort.

How to use an SQL query that you designed elsewhere

1. Copy your SQL statement to the Windows Clipboard. Many SQL editors will let you copy the SQL statement to the Clipboard using the Ctrl+C key combination.
2. Click the NEW button on the toolbar in the Crystal Query Designer. The New Query dialog box appears.

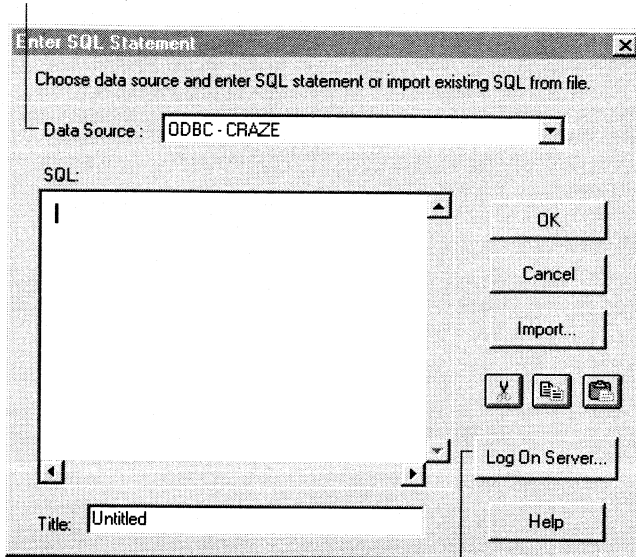


Choose the Server from the Server Type list then click OK to log on.

3. Click *Enter SQL statement directly*. The Log On Server dialog box appears.
4. Select the desired SQL server for your SQL statement, and click OK. The SQL Server Login dialog box appears.
5. Type in your user ID and password to log on to the SQL server, and click OK.

6. After a message indicating the success of your log on, assuming you logged on correctly, the Enter SQL Statement dialog box appears.

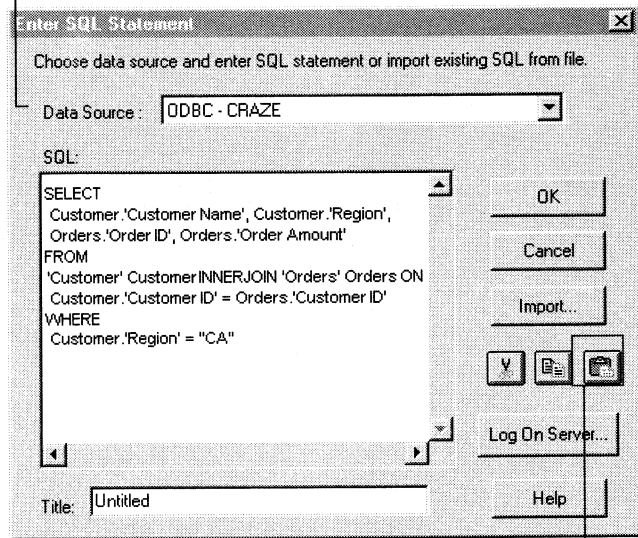
Specify the ODBC data source from this drop down list.



If you are not already logged on to an SQL Server, click the Log On Server button and do so using the Log On Server dialog box.

7. If you need to log on to another server or ODBC data source, click the *Log On Server* button. All data sources that you are logged on to will appear in the Data Source drop down list box.
8. Use the *Title* text box to give a title to your query that describes its purpose.
9. You enter your SQL statement in the large *SQL* edit box. Click anywhere inside the box to place an insertion point.
10. Press Ctrl-V or click the *Paste* button in the dialog box to paste your SQL statement into the *SQL* edit box.

Edit your SQL query in this box.



Paste button

11. The *SQL* edit box allows you to make any changes you wish to your *SQL* statement. You can even enter an entirely new *SQL* statement if you wish. Make any changes you need and click the *OK* button when you are finished.

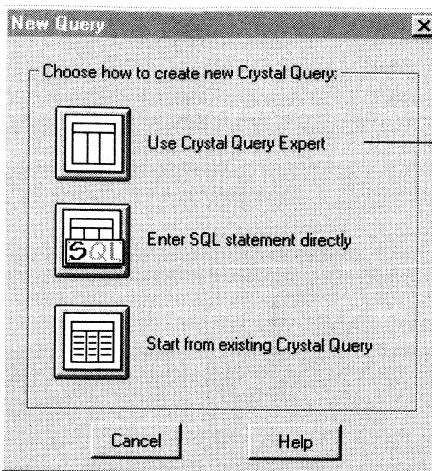
Your old *SQL* statement becomes a new *SQL* query that can be used with Crystal Reports just like any other *SQL* query file.

Alternatively, you can import an *SQL* statement saved in an ASCII text file. To do so, disregard step 1 above, follow steps 2 through 6 to open the Enter *SQL* Statement dialog box and log on to the ODBC data source, then click *Import* to import the *SQL* statement from the text file.

How to create a new query



1. Click the **NEW** button on the toolbar in the Crystal Query Designer. The New Query dialog box appears.



Click the Use Crystal Query Expert button.

2. Click the *Use Crystal Query Expert* button to open the Create Query Expert.

The Create Query Expert dialog box consists of several tabs. The tabs are numbered to lead you step-by-step through the query creation process.

Step 1: Tables | 2: Links | 3: Fields | 4: Sort | 5: Select | 6: SQL

NOTE: *The Links Tab only appears if you select two or more database tables on the Tables Tab.*

The remaining sections in this chapter explain how to perform specific tasks with the Crystal Query Expert.

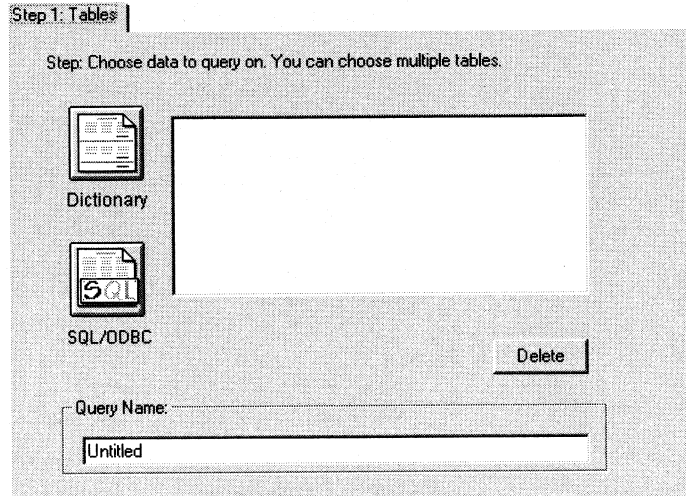
NOTE: *Once you select at least one field for the query on the Fields Tab, you can view the data retrieved by your query at any time, by clicking the Preview Query button at the bottom of the dialog box. After you are done viewing the query data, use the EDIT button on the toolbar to return to the Query Expert.*

How to add tables and fields

Generates FROM clause

The FROM clause specifies the sources (tables) of the database fields indicated in the SELECT statement.

1. Click the Tables Tab in the Crystal Query Expert.



2. Click *SQL/ODBC* or *Dictionary*.
3. If you clicked *SQL/ODBC*:
 - Select an SQL or ODBC data source in the Log On Server dialog box.
 - Choose one or more database tables in the Choose SQL Table dialog box. Click *Add* to add each table to your SQL query. Click *Done* when finished.
4. If you clicked *Dictionary*:
 - Select the Crystal Dictionary (.DC5) in the File Open dialog box, and click *OK*.
 - All ODBC database tables accessible from the Dictionary will be added to your SQL query.

Related Topics

See *FROM clause*, Page 441.

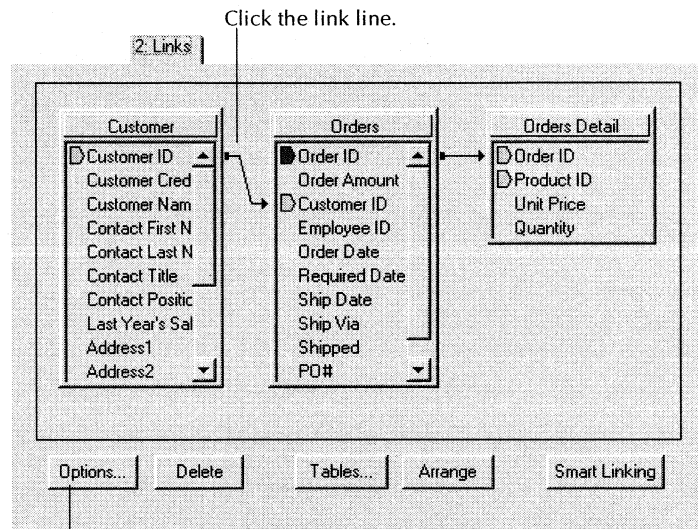
How to link tables and specify a join type

Generates WHERE clause

The WHERE clause indicates how two database tables are joined.

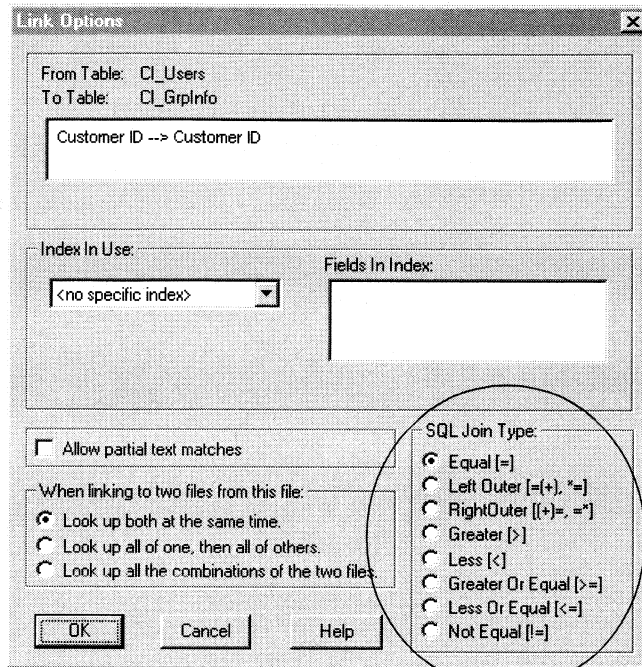
NOTE: Some links can be generated in the FROM clause.

1. Click the Links Tab and click the Smart Linking button. The Create Query Expert will make any possible links it can detect between tables. These links are represented by an arrow between fields in two tables, called a link line.



Then click the Options button to view the description of the link in the Link Options dialog box.

2. Click one of the link lines between tables. The link line becomes highlighted, along with the fields in each table that it is linking.
3. Click *Options* to open the Link Options dialog box. This dialog box describes the link between the tables in more detail.



4. Select an SQL join

Related Topics

See *SQL join types*, Page 426.

See *WHERE clause*, Page 442.

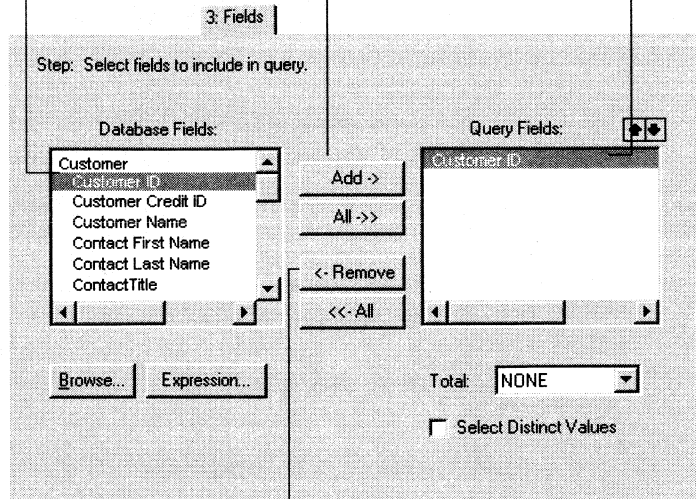
How to add fields

Generates SELECT clause

The **SELECT** clause selects specific data items to retrieve from the database tables indicated by the **FROM** clause.

1. Click the Fields Tab in the Create Query Expert.

2. Highlight the fields you want to appear in your query from this list.
3. Click the Add button.
4. The field(s) will be added to this list.



5. Click Remove to remove any fields from the Query Fields list box.

Related Topics

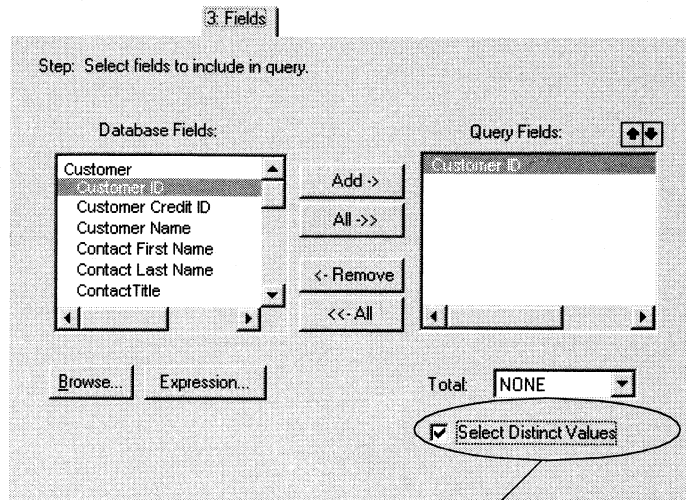
See *SELECT clause*, Page 441.

How to identify unique values

Generates DISTINCT clause

DISTINCT forces the query to retrieve only unique (distinct) sets of data.

1. Click the Fields Tab of the Create Query Expert.



Click the Select Distinct Values option to activate it.

2. Click *Select Distinct Values* to put a check in the check box.

Related Topics

See *DISTINCT clause*, Page 441.

How to summarize data with aggregate functions

Generates GROUP BY clause

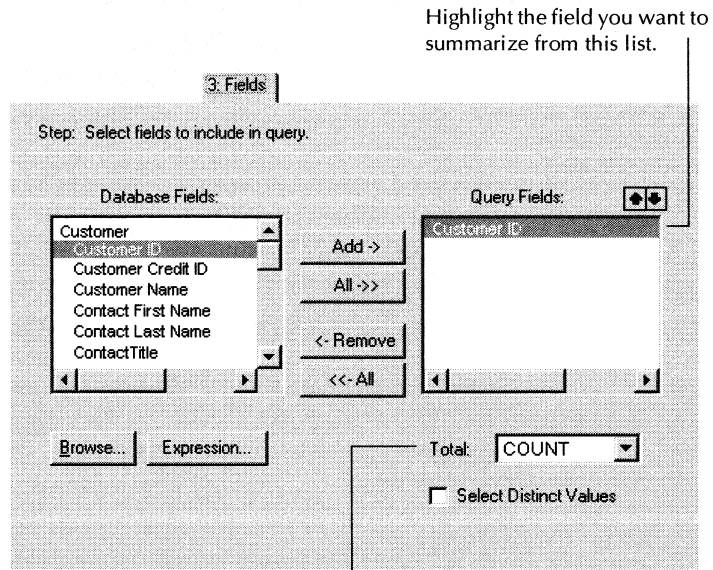
The GROUP BY clause retrieves a set of summary data.

Aggregate functions

Use aggregate functions to obtain summary information on all records or on groups of records. Aggregate functions are most useful when you do not need the detail information and only want to examine totals.

For example, you might need to know the total number of orders made and the average amount of each order. For this kind of query, you apply the COUNT function to the Order ID field, and the AVG (average) function to the Amount field. The query calculates the summary information and provides the results you need.

1. Click the Fields Tab of the Create Query Expert.
2. Select the field in the *Query Fields* list box that you want summarized.



Choose an aggregate function to apply to the field highlighted in the Database Fields list.

3. In the *Total* drop down box, choose an aggregate function to apply to the highlighted field.
 - COUNT() counts the number of values within a group.
 - SUM() adds together the values within a group for a total.
 - AVG() finds the average value of all values within a group.
 - MIN() finds the minimum value within a group.
 - MAX() finds the maximum value within a group.
4. The query summarizes the field that the aggregate function is applied to. Any other fields that appear in the *Query Fields* list box are used to sort the data.

Related Topics

See *GROUP BY clause*, Page 443.

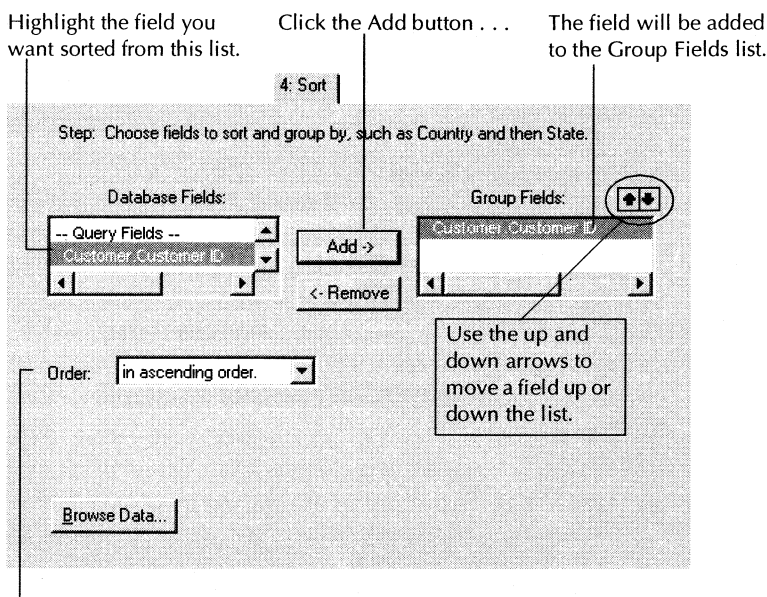
How to sort records according to field values

Generates ORDER BY clause

The ORDER BY clause indicates that the database records retrieved be sorted according to the values in a specific field.

Query data can be grouped either by sorting data, so that records with like data appear grouped in the sorted list, or by summarizing data with aggregate functions, so that summary data appears in your query for each group of records. For information on using aggregate functions in your query, see Page 370. This section shows you how to group data by sorting.

1. Click the Sort Tab in the Create Query Expert.



2. Select the field(s) in the *Database Fields* list box that you want the sort to be based on.
3. Click *Add* to add the selected field to the *Group Fields* list box. The field(s) listed in that box will be sorted in your query in the order they appear in the list.

4. If you want to move a field up or down the list, highlight the field and use the *up* and *down* arrows above the list to move the field where you want it.
5. For each sort field, use the *Order* drop down box to select the order in which you want data in the specified field sorted (ascending or descending).

Related Topics

See *ORDER BY clause*, Page 442.

How to specify records to be included

Generates WHERE clause

The WHERE clause can specify record selection criteria.

1. Click the Select Tab of the Create Query Expert.

Highlight the field you which contains the data you want to base selection criteria on from this list.

Click the Add button . . .

The highlighted field will be added to the Select Fields list.

Step: Choose fields to select (or filter) records in query. Important for speed.

Database Fields:

- Customer
- Customer ID
- Customer Credit ID

Select Fields:

- Customer Credit ID

is any value.

Use these controls to specify the selection criteria to be applied to the field highlighted in the Select Fields dialog box.

Browse Data...

2. In the *Database Fields* list box, select the field containing the data that you want to base selection criteria on.

3. Click the *Add* button to add the field to the *Select Fields* list box. Add as many fields as you want to base selection on.
4. Select one of the fields in the *Select Fields* list box. Several controls appear below the list boxes.
5. Use the controls that appear to specify the selection criteria to be applied to the field highlighted in the *Select Fields* list box. These choices work much like the *Select Records Expert* in Crystal Reports. Search for *Select Records Expert* in Crystal Reports online Help.
6. Repeat steps 4 and 5 for every field that appears in the *Select Fields* list box.

Related Topics

See *WHERE clause*, Page 442.

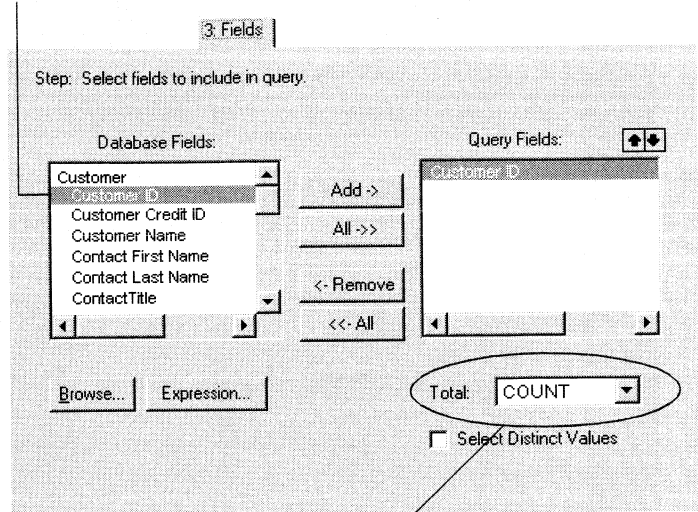
How to select groups to be included

Generates GROUP BY and HAVING clauses

The HAVING clause creates selection criteria for the summary information produced by the GROUP BY clause.

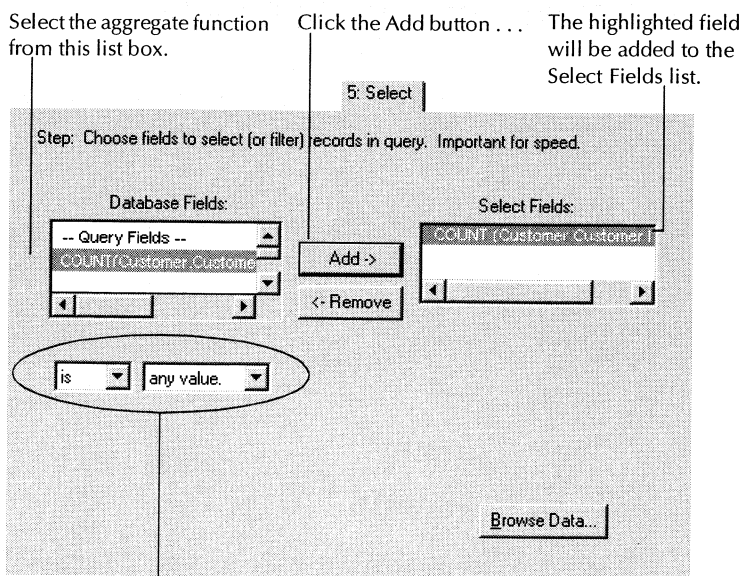
1. Click the Fields Tab of the Create Query Expert.

Highlight the field you want to summarize in this list.



Select the aggregate function from this drop down list that you want to apply to the highlighted field.

2. Select the field in the *Query Fields* list box that you intend to summarize.
3. Select the aggregate function from the *Total* drop down box that you want to apply to the selected field.
4. Click the *Select* Tab.



Use the selection criteria controls to specify which group summary values based on the aggregate function should appear in the result.

5. Select the aggregate function in the *Database Fields* list box that you created.
6. Click *Add* to add the aggregate function to the *Select Fields* list box. Several selection criteria controls appear below the list boxes.
7. Use the selection criteria controls to specify which group summary values, based on the aggregate function, should appear in the query results.

Related Topics

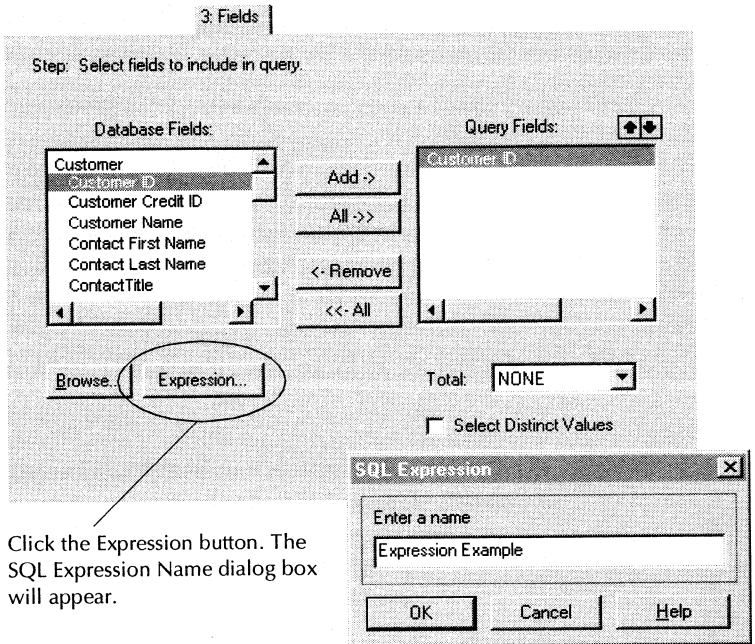
See *GROUP BY clause*, Page 443.

See *HAVING clause*, Page 443.

How to create an SQL expression

NOTE: You must have some familiarity with the SQL language and SQL expressions before trying to add an expression to your Crystal query file.

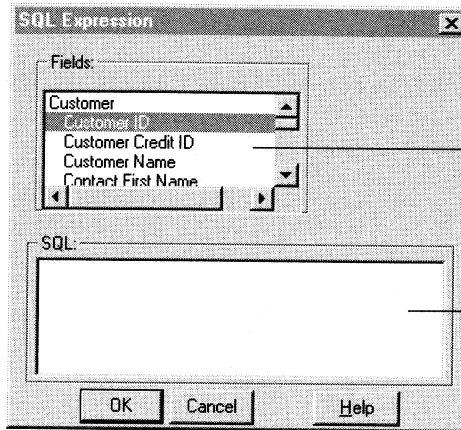
1. Click the Fields Tab in the Create Query Expert.



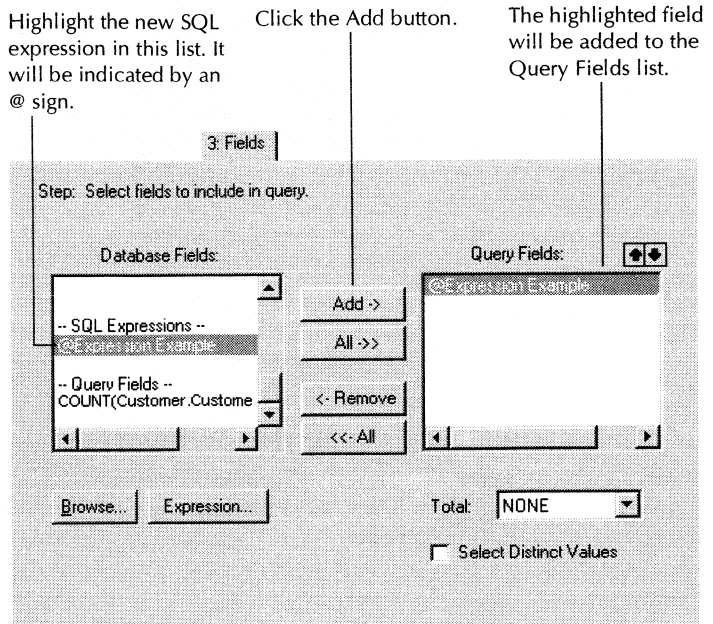
Click the Expression button. The SQL Expression Name dialog box will appear.

Type a name for your new expression and then click OK.

2. Click the *Expression* button. The SQL Expression Name dialog box appears.
3. Type a name for your new expression, and click OK. The SQL Expression dialog box appears.



4. Create an SQL expression for your query.
 - To add a field to the expression, double-click on the field in the *Fields* list box at the top of the dialog box.
 - To edit the expression directly, click the mouse pointer to place an insertion point in the SQL Expression text box. Type any changes to the expression you want to make.
5. When you are finished designing your SQL expression, click *OK*, you are returned to the Fields Tab.

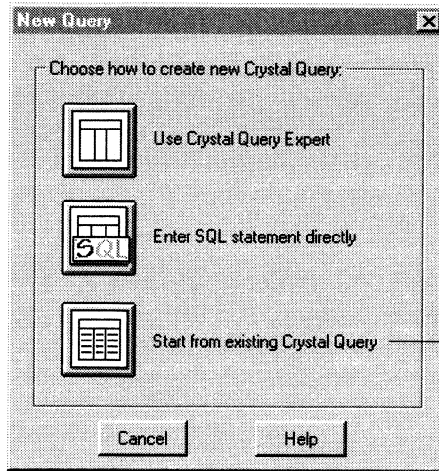


6. Scroll the *Database Fields* list box to find the new SQL expression. An @ symbol appears before the name of the expression, indicating it is an expression and not a field.
7. Select the expression in the *Database Fields* list box.
8. Click *Add* to add the new expression to the *Query Fields* list box.

How to create a query from another Crystal Query



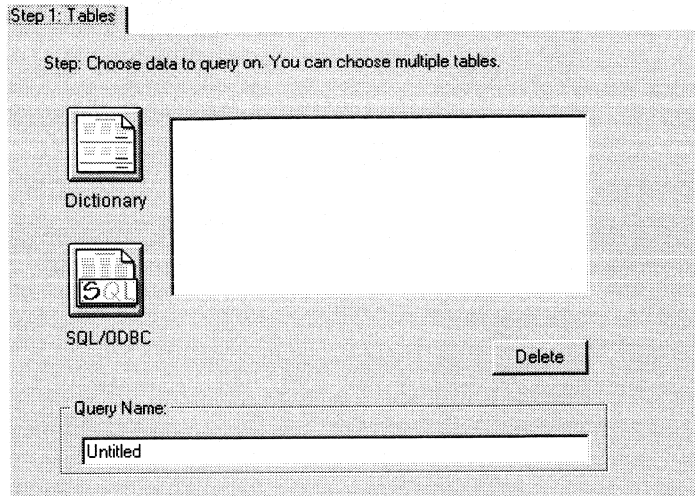
1. Click the NEW button on the toolbar in the Crystal Query Designer. The New Query dialog box appears.



Click the Start from existing Crystal Query button.

2. Click *Start from existing Crystal Query*. A File Open dialog box appears.
3. Use the File Open dialog box to select the query (*.QRY) file that you want to use to base a new query on, and click OK. The program runs the existing SQL query and displays the data set.
4. Click the EDIT button on the toolbar. The Create Query Expert dialog box appears with the specifications for the selected query already in place.





5. Use the Create Query Expert to make any changes necessary to the query. Click *Preview Query* when finished to display the new query results.
6. Choose the SAVE AS command from the File menu to save the new query under a different name. The new query is saved in a separate file; your source query remains unchanged.

How to select a Crystal query for a report



1. In Crystal Reports, Click the NEW button on the toolbar. The Report Gallery appears.
2. Select a Report Expert from the Report Gallery, and click the button for that Expert. The Create Report Expert appears.
3. On the Tables Tab, click the *Query* button, and a File Open dialog box appears.
4. Use the controls in the File Open dialog box to locate and select the query (.QRY) file you want to use to create a new report.
5. Click *OK*, and the query is opened for your report. Use the tools in the Create Report Expert to design a new report. Your query fields will appear in each section of the Expert just as

any other database fields would. However, the name of the query file will appear above the fields in list boxes where you would normally see the name of a database table.

NOTE: You can not use more than one query file in a report.

18

Dictionaries

What you will find in this chapter...

Crystal Dictionaries Overview 384

How to create a new Dictionary 385

How to open a data file 386

How to open an SQL or ODBC data source 388

How to link multiple tables 389

How to select tables and fields for users 390

How to add formulas 392

How to rename field headings, fields, and formulas 393

How to update the location of a database table 394

How to move fields/field headings within the list 395

How to add a new field heading 396

How to add Help text 396

How to add a graphic 397

How to create sample data for users to browse 399

How to edit an existing Dictionary 400

How to convert a Crystal Reports 3.x or 4.x Dictionary file 400

How to select a Crystal Dictionary for a report 401

Crystal Dictionaries Overview

A Dictionary is a structured and simplified view of data that you can create for some or all of the individuals in your organization that are using Crystal Reports.

Unlike some systems that force users to access data through a data distribution metalayer, Dictionaries are an optional component of Crystal Reports. Data can still be accessed directly by the user. Dictionaries simply provide all of the convenience without the restrictions.

Crystal Dictionaries let you:

- design a single, dynamic view of all the data that is necessary to create organizational reports and queries,
- organize the data and rename tables and fields, to make it easier for users to understand the content and purpose of the data, and
- create complex data-manipulation formulas that users can access without the need to understand formula concepts.

Dictionaries reduce support cost and time, increase user productivity, and reduce data misuse, loss, and damage. It is a powerful component of Crystal Reports.

Why use a Crystal Dictionary?

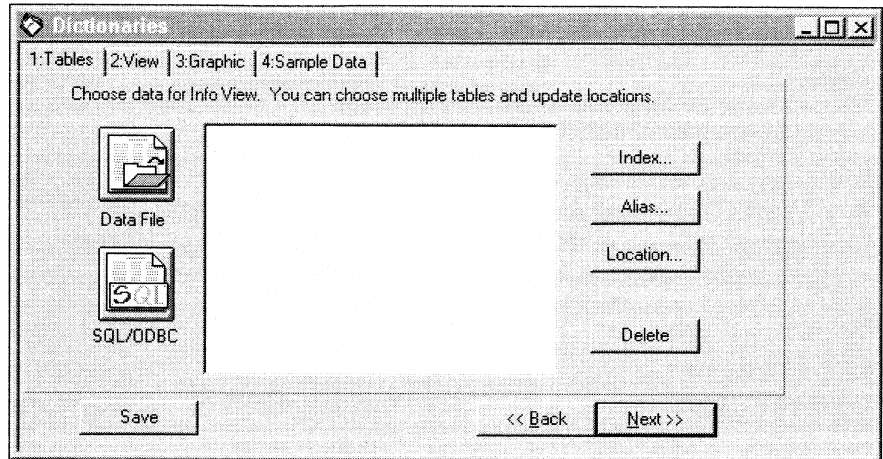
Crystal Dictionaries are often designed and distributed by Information System (IS) Managers or Network Administrators who control and manage a company's databases. These databases are often complex collections of data spread throughout several tables with hundreds or even thousands of fields. A user, trying to locate and use a small set of data for a report, can easily get lost among database, table, and field names.

By creating a customized dictionary that contains a small amount of data specific to the work performed by a small group of users. You provide those users with clear and easy access to all of the data they will ever need. For example, the Accounting department's dictionary can be different than the Sales department's dictionary, or the Personnel department's dictionary. Some data may overlap between dictionaries, but it can be named and organized in a fashion that best suits the users accessing it.

How to create a new Dictionary



1. In Crystal Dictionaries, click the NEW button on the toolbar. The Dictionaries Expert appears:



This Expert contains four tabs. Each tab is numbered to lead you through the Dictionary development process step-by-step.

NOTE: If you add more than one table, the Links Tab will appear next to the Tables Tab.

2. To access the information and controls on each tab, simply click the tab. Information and controls needed for the selected step will be displayed in the dialog box. You may also use the *Next>>* and *<<Back* buttons to go to the next (right) and previous (left) tab, respectively.

NOTE: Certain steps must be performed before others in the Dictionary creation process. For example, you must select tables for the Dictionary before you can select fields from those tables. For that reason, some tabs may not be available until you perform the required steps prior to selecting those tabs.

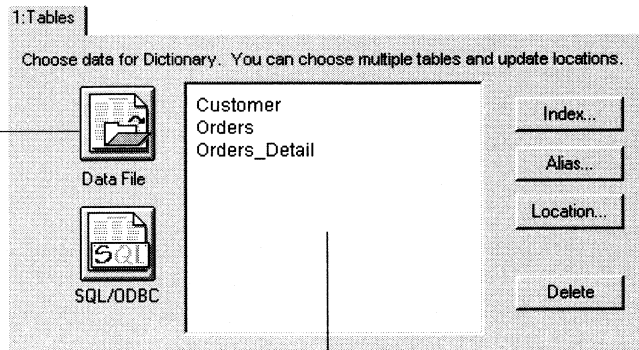
3. Click *Save* at the bottom of the Expert to save your Dictionary at any time.

NOTE: Since a Dictionary must contain some database data, the Save button will be disabled until you add at least one field to the Headings & fields in View list box on the View Tab.

How to open a data file

1. While in the Dictionaries Expert, click the Tables Tab to activate it.

2. Click the Data File button.



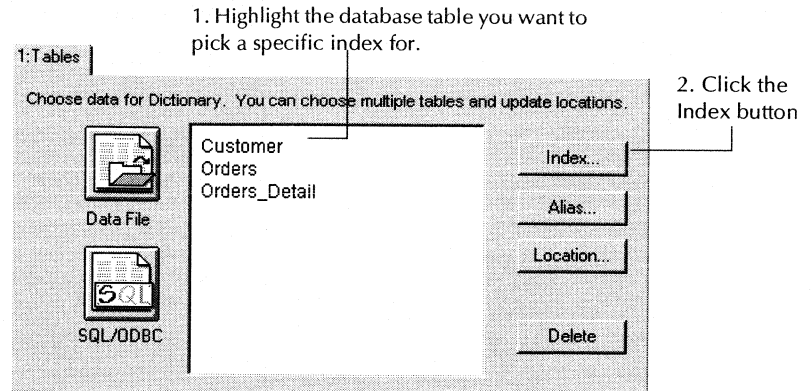
All tables that you add in the Choose Database File dialog box will appear in this list box.

The Choose Database File dialog box appears.

3. Use the controls in the Choose Database File dialog box to select a database file.
4. Click *Add* to add the file to the Dictionary. All tables from that file will appear in the list box on the Tables Tab.
5. Repeat steps 3 and 4 for each database file you want to add to the Dictionary.
6. Click *Done* when you are finished adding database files.

NOTE: When you add a database file to a Dictionary from the Tables Tab, the tables and fields from that file will not necessarily appear to the user who opens the Dictionary from Crystal Reports. The tables that appear on the Tables Tab are only the tables available to you, as an administrator, for adding to the Dictionary. To add specific fields to the Dictionary, see How to select tables and fields for users, Page 390.

You may also need to pick a specific index file to be used by one of the database tables. This is done in the Tables Tab:



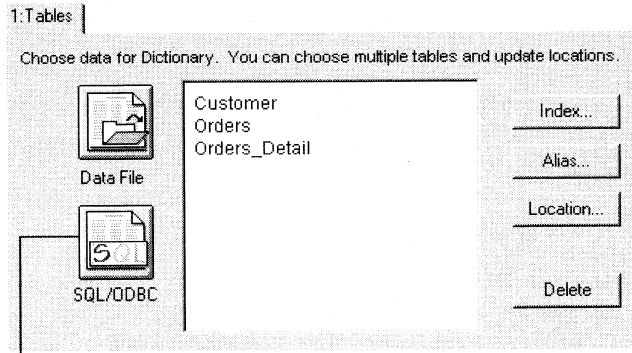
3. The Choose New Location dialog box appears. Use the controls in this dialog box to select a new index file, and click OK when finished.

NOTE: By default, Crystal Dictionaries will use any index file it finds with the same name as the database file. You only need to select an index file if you want to use an index with a different name than the database file. For more information on indexes, see *Indexed Tables, Page 409*.

NOTE: You can mix data files and ODBC data sources in the same Dictionary file. However, if you wish to link data files and ODBC data sources together, you can only use string fields to perform the links.

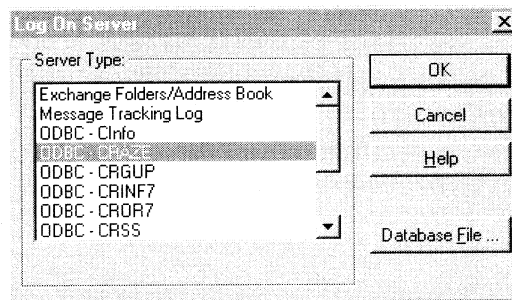
How to open an SQL or ODBC data source

1. While in the Dictionaries Expert, click the Tables Tab to activate it.



2. Click the SQL/ODBC button.

3. The Log On Server dialog box appears:



4. Select an SQL or ODBC data source and click OK when finished.

5. If the data source requires any log on information, such as user name and password, the SQL Server Login dialog box will appear. Use this dialog box to log on to the ODBC data source just as you normally do from your Database Management System application.
6. Click *OK*, and the Choose SQL Table dialog box appears.
7. Select a database table in the Choose SQL Table dialog box, and click *Add* to add it to your Dictionary.
8. Repeat Step 7 for each table you want to add to the Dictionary.

9. Click *Done* when finished.

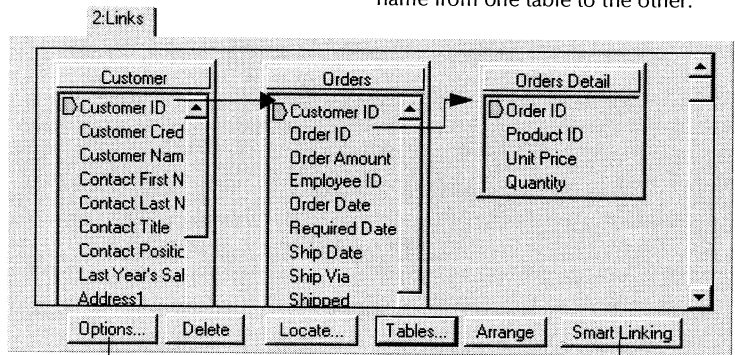
NOTE: When you add an SQL or ODBC data source to a Dictionary from the Tables Tab, the tables and fields from that data source will not necessarily appear to the user who opens the Dictionary from Crystal Reports. The tables that appear on the Tables Tab are only the tables available to you, as an administrator, for adding to the Dictionary. To add specific fields to the Dictionary, see *How to select tables and fields for users*, Page 390.

NOTE: You can mix data files and ODBC data sources in the same Dictionary file. However, if you wish to link data files and ODBC data sources together, you can only use string fields to perform the links.

How to link multiple tables

1. If you have added more than one database table to your Dictionary, click the Links Tab in the Dictionaries Expert.

To create a link manually, drag a field name from one table to the other.



Click the Options button for a detailed description of a highlighted link.

Click Smart Linking to let Crystal Dictionaries create logical links between tables in your Dictionary.

- If a link is possible between two tables, you can create a new link by dragging a field name from one table to the other. Crystal Dictionaries will draw a new link arrow between the tables. See *Linking Tables*, Page 411.

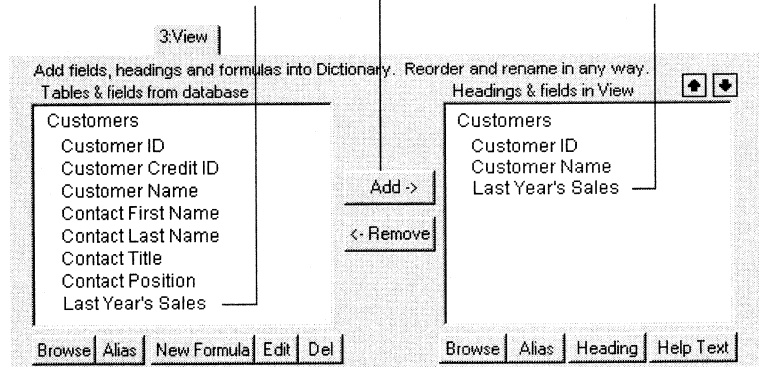
- If you select a link and click the *Options* button, the Link Options dialog box appears. You can use the controls in this dialog box to make any necessary changes to the highlighted link. Search for *Link Options dialog box* in Crystal Reports online Help.

How to select tables and fields for users

After you add tables to your dictionary using the Tables Tab, those tables, and the fields in them, are not automatically available to your users. You must expose the fields using the View Tab. The View Tab lets you design the actual view of the data that your users will see.

1. While in the Dictionaries Expert, click the View Tab to activate it.

2. Highlight the field you want to make available to users from this list.
3. Click the Add button.
4. The field will be added to this list.



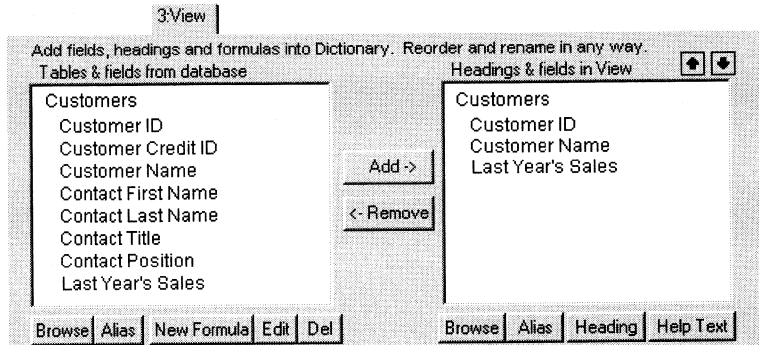
- To make a table and all its fields available to the users of your Dictionary, select the table name from the *Tables & fields from database (Database)* list box, and click *Add*. The table name becomes a field heading in the *Headings & fields in View (View)* list box. All fields from the table appear as fields under the new field heading.
- To make a single field available to the users of your Dictionary, select the field in the *Database* list box and click *Add*. The field is added to the end of the list in the *View* list box.

When exposing and organizing tables and fields in the View Tab, keep in mind the following points:

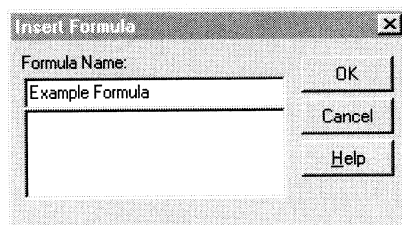
- Tables are not exposed as database tables. Table names become field headings. These headings appear just like tables when the user designs a report based on your Dictionary. However, field headings do not necessarily represent database tables that actually exist.
- Field headings can be added anywhere in the *View* list box to provide clearer organization of data for your users. See *How to add a new field heading*, Page 396.
- Fields can be added to the *View* list box as many times as necessary. A field does not have to appear under a field heading that matches the table the field exists in. Organize fields in any order and under any field headings that will work best for your users.
- Fields must be grouped under field headings. If you add a single field to the *View* list box and no field heading appears in the list box (the list box is empty), Crystal Dictionaries will provide a default field heading for you and will add the field you selected underneath that heading.
- Field names and field headings that appear in the *View* list box can be renamed to anything you want. See *How to rename field headings, fields, and formulas*, Page 393.

How to add formulas

1. While in the Dictionaries Expert, click the View Tab to activate it.



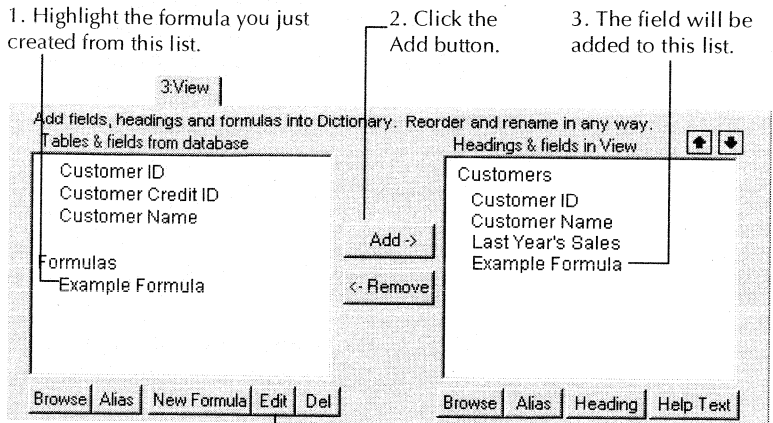
2. To create a formula, click the New Formula button.
3. The Insert Formula dialog box appears:



4. Type a name for your formula and click OK when finished.

5. The Crystal Reports Formula Editor appears.
6. Use the Formula Editor to create a formula for the Dictionary, just as you would create a formula for a report. See Chapter 10, *Formulas 101*, Page 249, or search for *Formula Editor* in Crystal Reports online Help.
7. Click *Accept* when finished. The new formula is added to the *Tables & fields from database* list box.

Next, you will need to add the formula to the view for your dictionary.



1. Highlight the formula you just created from this list.

2. Click the Add button.

3. The field will be added to this list.

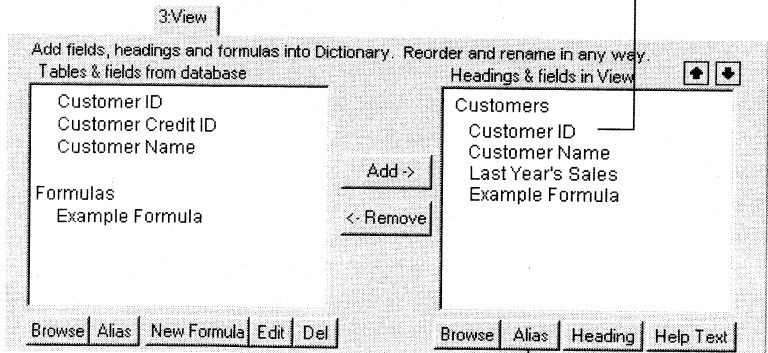
4. Highlight the formula above and click the Edit button to modify your formula.

NOTE: The formula name appears in the Headings & fields in View list box without an @ symbol. By removing the @ symbol, the Dictionary hides the fact that this is a formula. The user can use it just like another field.

How to rename field headings, fields, and formulas

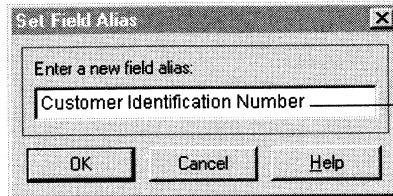
1. While in the Dictionaries Expert, click the View Tab to activate it.

2. Highlight the heading or field you want to rename from this list.



3. Next, click the Alias button.

4. A Set Alias dialog box appears. If you chose a heading, the Set Table Alias dialog box appears. If you chose a field, the Set Field Alias dialog box appears. See *Aliases*, Page 406.

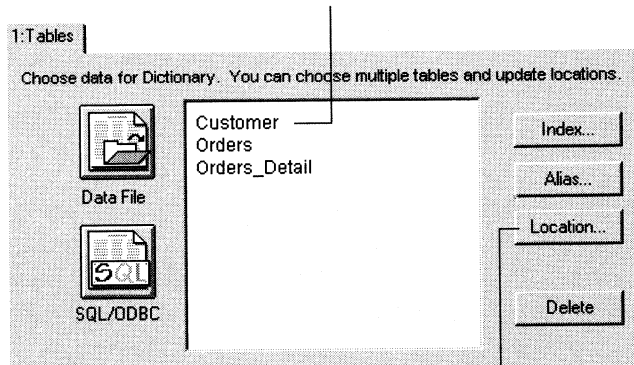


5. Type the new name into this text box and click OK when finished.

NOTE: This process does not actually rename tables or fields. Only the alias name that appears to the user opening the Dictionary from Crystal Reports is changed. The original database file is not affected.

How to update the location of a database table

1. While in the Dictionaries Expert, click the Tables Tab to activate it.
2. Highlight the table or field that has changed name or location from this list.



3. Then, click the Location button.

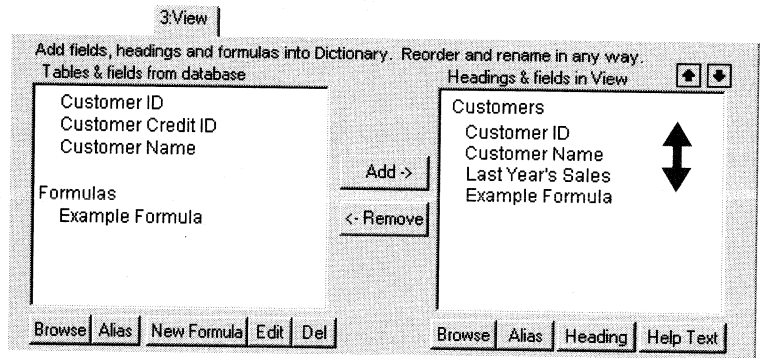
NOTE: Remember, the table names that appear here are aliases rather than the actual database table names. See *Aliases*, Page 406.

- If the table is from a data file, the Choose New Location dialog box appears.

- If the table is from an SQL server or other ODBC data source, the Choose SQL Table dialog box appears.
4. Select the new name or location of the database table from the dialog box that appears, and click *OK*.

How to move fields/field headings within the list

1. While in the Dictionaries Expert, click the View Tab to activate it.
2. Highlight the heading or field that you want to move from this list and drag it up or down to the desired position.

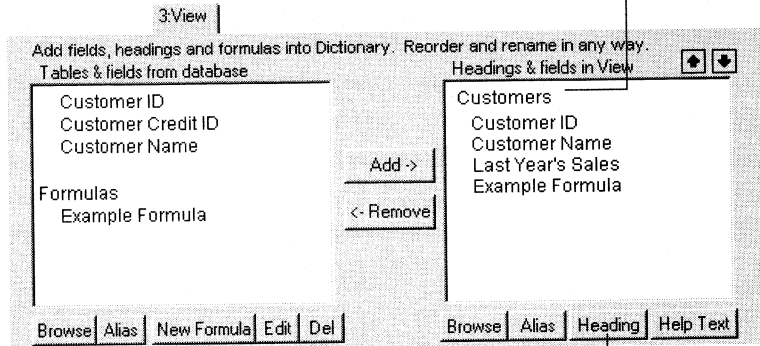


The order that the fields and field headings appear in this list box is the order that they will appear to users that open the Dictionary from Crystal Reports.

NOTE: When organizing fields and field headings, remember that field headings take the place of database tables when the user opens the Dictionary in Crystal Reports. Because of this organization, the first item in the list box must be a field heading.

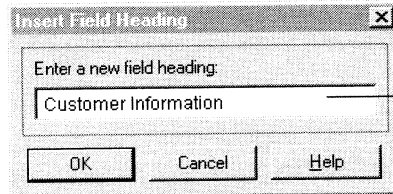
How to add a new field heading

1. While in the Dictionaries Expert, click the View Tab to activate it.
2. Highlight the field that will be the first to appear under the new field heading from this list.



3. Then, click the Heading button.

4. The Insert Field Heading dialog box appears.



5. Type the new name into this text box and click OK when finished.

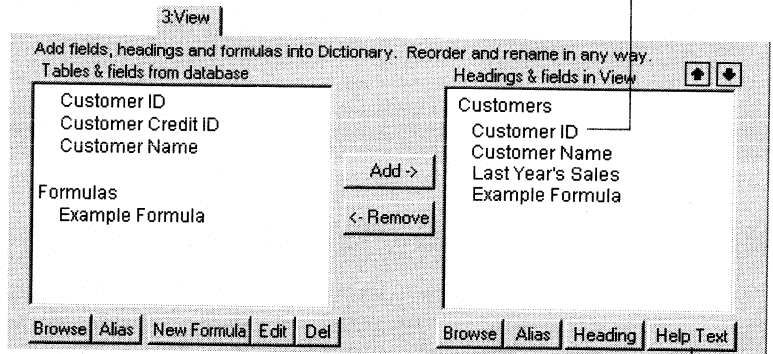
The field appears in the *Headings & fields in View* list box with the new name.

How to add Help text

To maximize the efficiency of your Dictionary, Crystal Reports allows you to add Help text. Whenever anyone needs clarification on what the elements of your Dictionary are, they can point to the item in question and Help text will appear in a pop-up window to assist them.

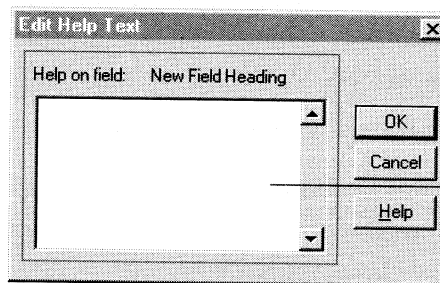
1. While in the Dictionaries Expert, click the View Tab to activate it.

2. Highlight the field that you want to add help text to from this list.



3. Then, click the Help Text button.

4. The Edit Help Text dialog box appears.

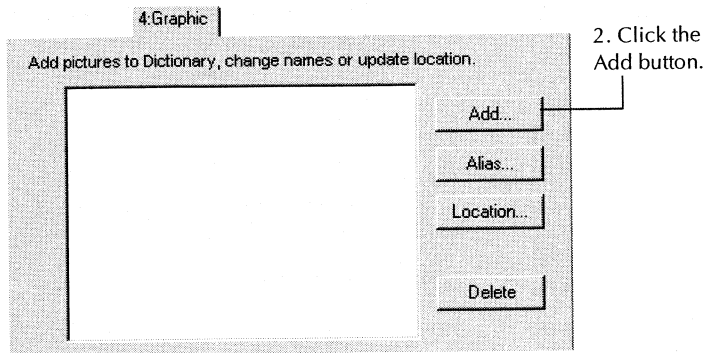


5. Enter the Help Text into this text box and click OK when finished.

When the user selects the field or field heading from Crystal Reports and holds the cursor over the selected item, the help text will appear in a pop-up window.

How to add a graphic

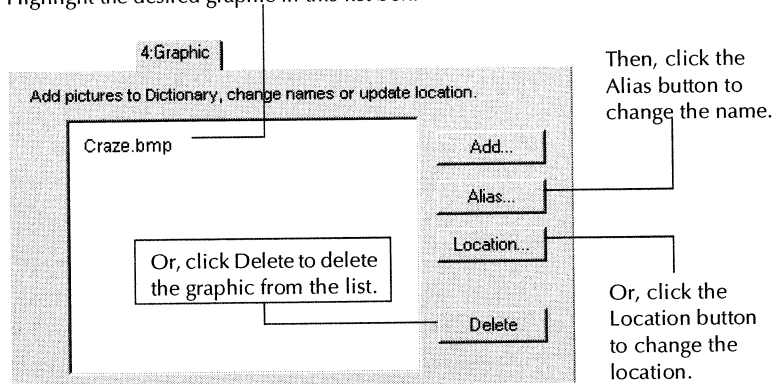
1. While in the Dictionaries Expert, click the Graphic Tab to activate it.



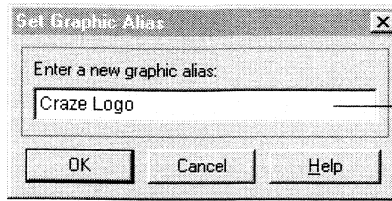
3. The File Open dialog box appears.
4. Use the controls in the File Open dialog box to select a graphic image file, and click *OK*. Crystal Dictionaries support the Windows bitmap (.bmp), CompuServe (.gif), PC Paintbrush (.pcx), Tiff (.tif), and TARGA (.tga) graphic formats. See *How to insert a graphic object*, Page 397. The image file appears in the list box of the Graphic Tab.

You may now want to change the alias name, update the location, or delete a graphic. Simply highlight the desired graphic and click the appropriate button for your needs.

Highlight the desired graphic in this list box.



- If you clicked the *Alias* button, the Set Graphic Alias dialog box appears.



Type a new name for the graphic in this text box and click OK when finished.

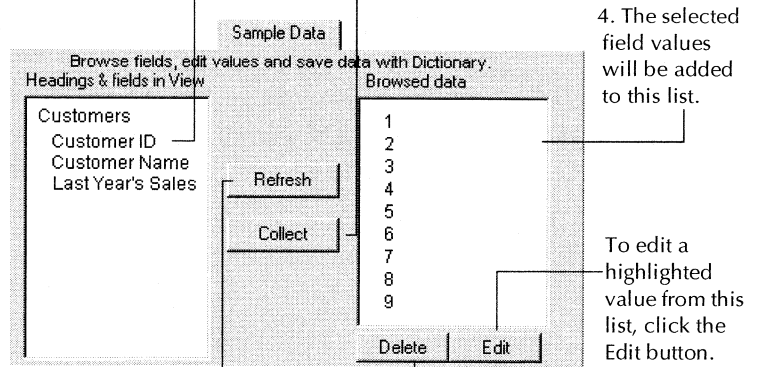
- If you clicked the *Location* button to set the new location of the image, the Choose New Location dialog box appears. Use the Choose New Location dialog box to find the new name and/or location of the graphic file.

How to create sample data for users to browse

The Sample Data Tab in the Dictionaries Expert lets you create a set of custom sample data that appears to users when they browse field data. The data they see may or may not reflect actual data in the database, depending on the sample data you design.

1. While in the Dictionaries Expert, click the Sample Data Tab to activate it.

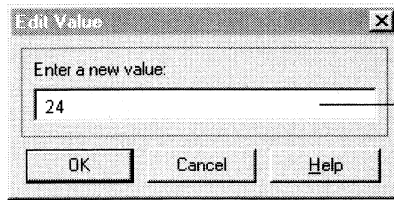
2. Highlight a field that you want to make browseable from this list.
3. Then click the Collect button.



If the data has changed, click the Refresh button to retrieve a list of the new data.

To delete a highlighted field value from the list above, click the Delete button.

- If you clicked the *Edit* button, the Edit Value dialog box appears.



1. Enter the new Value in this text box and click OK when finished.

2. The new value will appear in the *Browsed data* list box.

NOTE: Deleting and editing values in the *Browsed data* list box of the *Sample Data* Tab does not change the database file. It only changes the list of values that appears to a user whenever the *Browse and Paste* dialog box is opened in *Crystal Reports*. This allows you to customize the look of the data when users browse data, without actually changing the data that is reported on.

How to edit an existing Dictionary



1. In Crystal Dictionaries, click the OPEN button on the toolbar. The File Open dialog box appears.
2. Use the *Drives*, *Directories*, and *File Name* controls to select the existing Dictionary (.DC5) file and click OK. The Dictionaries Expert will appear.
3. Use the Dictionaries Expert to make changes to your Dictionary file.

How to convert a Crystal Reports 3.x or 4.x Dictionary file



1. In Crystal Dictionaries, click the OPEN button on the toolbar. The File Open dialog box appears.
2. Use the *Drives*, *Directories*, and *File Name* controls to select the old Crystal Dictionary (.DCT) file and click OK. The Select View dialog box appears, listing the names of all views from the old Dictionary file.
3. Select the view you want to convert to a new Dictionary file, and click OK. The Dictionaries Expert appears with the data from the view you selected.

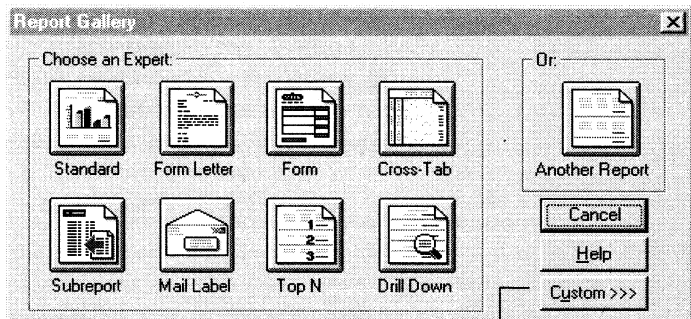
NOTE: Crystal Dictionaries 5.0 provides an easier and more powerful method for controlling data access than older versions of Crystal Dictionaries. However, each view from an older Dictionary file is handled as a separate Dictionary in Crystal Dictionaries 5.0. An older Crystal Dictionary file, on the other hand, held several views. To convert an entire Crystal Dictionary 3.x or 4.x file, you must open each view in the file separately and save it as a new Crystal Dictionaries 5.0 file.

4. Use the Dictionaries Expert to make changes to your new Dictionary file.
5. When you save your Dictionary file, it will be saved in the new 5.0 Crystal Dictionary format (.DC5).

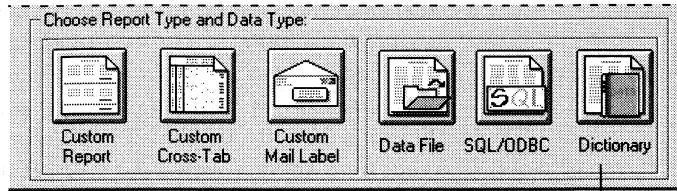
How to select a Crystal Dictionary for a report



1. Click the NEW button on the toolbar. The Report Gallery appears.



2. Click the Custom button.
3. An additional section will be added to the dialog box, revealing custom report options.



4. Click the Custom Report, Custom Cross-Tab or Custom Mail label button.

5. Then, click the Dictionary button.

6. Use the *Drives*, *Directories*, and *File Name* controls from the dialog box that appears to select the Dictionary (.DC5) file that you want to use, and click *OK*.



7. To add items from the Dictionary to your report, click the **INSERT FIELDS** button on the toolbar, and use the Insert Fields dialog box to select fields and images for your report. Field headings from the Dictionary file appear as tables in the Insert Fields dialog box, while fields from the Dictionary appear as fields for each table.

NOTE: You can not use more than one Dictionary file in a report at a time. Also, you can not link fields from Dictionary files to fields in other database tables.

19

Working With Databases

What you will find in this chapter...

For additional information 444

How to open Access queries through DAO 444

How to open Access queries through ODBC 446

How to open Access parameter queries 448

How to set up an ODBC data source 453

How to check settings for an ODBC data source 455

How to log on to an ODBC data source 457

How to add an ODBC database table to a report 457

How to log on to MS SQL Server via ODBC 459

How to log off an ODBC data source 461

How to change the ODBC data source accessed by a report 461

How to select a stored procedure from an SQL database and change stored procedure parameters 465

How to set up an A to B, A to C link 469

How to edit an SQL query 470

How to use an ACT! database 471

How to open the NT Event Log 473

Though there are hundreds of Database Management Systems available, Crystal Reports eliminates many of the differences once it connects to the actual database files. The process of working with database files, tables, fields, and records is much the same regardless of the actual type of data being accessed.

This chapter discusses several concepts and tasks common to working with database files. Using database aliases, locating moved or renamed database files, working with indexed tables, and linking tables are subjects common to anyone who designs reports in Crystal Reports. The last topic in this section, Using SQL and SQL databases, is especially important for anyone who accesses data in SQL databases and other database formats that are accessed through ODBC.

Relational Database basics

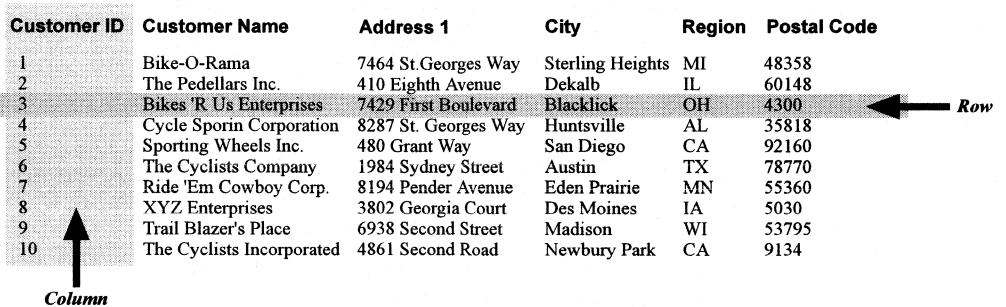
The most popular architecture for database files used in the corporate world is based on the relational model. Applications that allow you to create databases with the relational model are, therefore, often referred to as Relational Database Management Systems (RDBMS).

In a relational database, data is organized in a system of rows and columns. The rows are called records, and the columns are called fields. Each record contains a collection of related data, all information relating to a specific customer, for example. Each field refers to a common type of data that exists in all records, the names of the customers, for example. Records and fields are stored in a database table. The following diagram illustrates the basic relational database model:

Relational database table model

Customer Table

Customer ID	Customer Name	Address 1	City	Region	Postal Code
1	Bike-O-Rama	7464 St. Georges Way	Sterling Heights	MI	48358
2	The Pedellars Inc.	410 Eighth Avenue	Dekalb	IL	60148
3	Bikes 'R Us Enterprises	7429 First Boulevard	Blacklick	OH	4300
4	Cycle Sporin Corporation	8287 St. Georges Way	Huntsville	AL	35818
5	Sporting Wheels Inc.	480 Grant Way	San Diego	CA	92160
6	The Cyclists Company	1984 Sydney Street	Austin	TX	78770
7	Ride 'Em Cowboy Corp.	8194 Pender Avenue	Eden Prairie	MN	55360
8	XYZ Enterprises	3802 Georgia Court	Des Moines	IA	5030
9	Trail Blazer's Place	6938 Second Street	Madison	WI	53795
10	The Cyclists Incorporated	4861 Second Road	Newbury Park	CA	9134



Often, data in two different tables can be related by a common field. For example, a Customers table will have a Customer ID for each customer, and an Orders table will have the Customer ID of each customer who placed an order. This demonstrates a relationship between tables. The two tables can be linked (Page 411) by their common field. Examine the following diagram to understand how two tables can have a relationship:

Relationships between database tables

Customer Table

Customer ID	Customer Name	Address 1	City	Region	Postal Code
1	Bike-O-Rama	7464 St. Georges Way	Sterling Heights	MI	48358
2	The Pedellars Inc.	410 Eighth Avenue	Dekalb	IL	60148
3	Bikes 'R Us Enterprises	7429 First Boulevard	Blacklick	OH	4300
4	Cycle Sporn Corporation	8287 St. Georges Way	Huntsville	AL	35818
5	Sporting Wheels Inc.	480 Grant Way	San Diego	CA	92160
6	The Cyclists Company	1984 Sydney Street	Austin	TX	78770
7	Ride 'Em Cowboy Corp.	8194 Pender Avenue	Eden Prairie	MN	55360
8	XYZ Enterprises	3802 Georgia Court	Des Moines	IA	5030
9	Trail Blazer's Place	6938 Second Street	Madison	WI	53795
10	The Cyclists Incorporated	4861 Second Road	Newbury Park	CA	9134

Order Table

Customer ID	Order ID	Order Amount	Order Date	Ship Via
1	1	\$41.90	94/12/02	UPS
41	1002	\$5,060.27	94/12/02	Pickup
77	1003	\$186.86	94/12/02	UPS
18	1004	\$823.05	94/12/02	Pickup
64	1005	\$0.00	94/12/03	Loomis
7	1006	\$64.90	94/12/03	Purolator
32	1007	\$49.50	94/12/03	Parcel Post
11	1008	\$2,214.93	94/12/03	Purolator
25	1009	\$29.00	94/12/03	Loomis
54	1010	\$14,872.28	94/12/03	UPS
71	1011	\$0.00	94/12/03	Purolator
66	1012	\$10,259.10	94/12/03	Loomis
28	1013	\$1,142.12	94/12/03	Parcel Post
8	1014	\$0.00	94/12/04	Purolator
72	1015	\$0.00	94/12/04	UPS
64	1016	\$563.70	94/12/04	FedEx
38	1017	\$72.00	94/12/05	Purolator
37	1018	\$115.50	94/12/05	Loomis
30	1019	\$0.00	94/12/05	Parcel Post
25	1020	\$67.80	94/12/05	FedEx

Relationships between tables based on a common field.

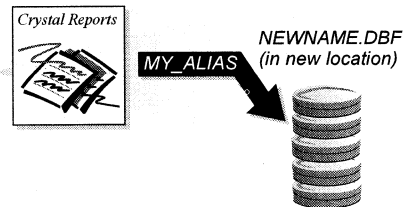
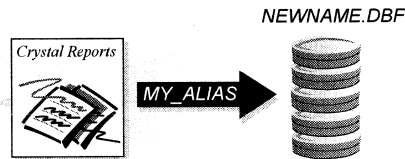
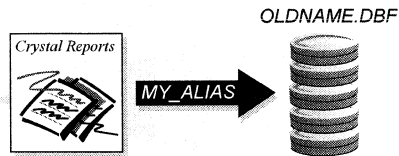
Aliases

For a variety of reasons, database names and locations get changed. If you create a report, then change the name or location of a table or file, Crystal Reports must be able to find the new name or location. This is especially important when you create formulas in your report that access a table that has been renamed or moved. To fix the reference for a single field in the Report Designer would not be difficult, but to find every formula that uses that field could be a difficult and time consuming task.

To solve this problem, Crystal Reports uses aliases to refer to database tables and files. Aliases are pointers, internal devices that tell Crystal Reports where it should look for a database field.

Now, if you change the name or location of the database, you simply reset the pointer. See *Locating files*, Page 408. The name of the alias does not change, so your formulas are not affected. Crystal Reports looks to the alias for the location and name, goes to the new location for the database field, and executes the formula without a problem.

Aliases



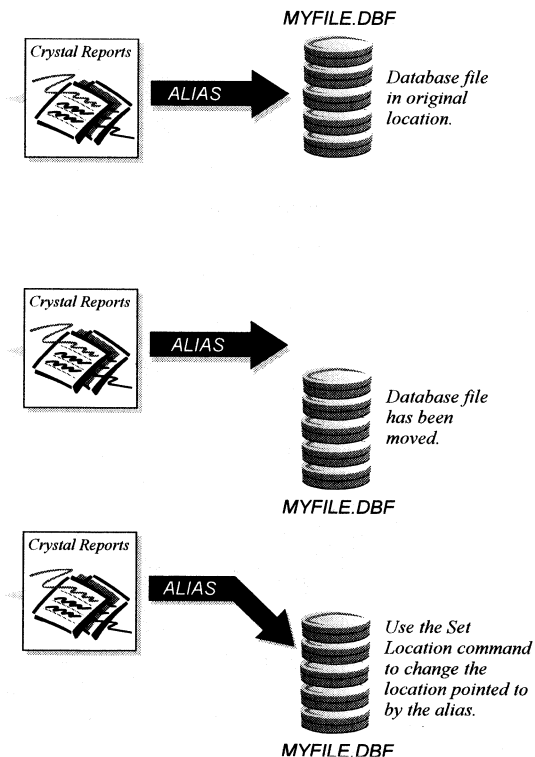
Crystal Reports automatically assigns default alias names to database tables when you first select the table or file. By default, an alias matches the original name of the table. In databases where the database table is a separate file (for instance, dBASE), the name of the database file is used without the file name extension. For example, if you use the dBASE database file COMPANY.DBF, Crystal Reports will assign a default alias name of COMPANY to the file. You can accept the default alias or assign a new one to the database table.

You can change an alias at any time using the SET ALIAS command on the Database menu. However, if you have already created formulas in your report using the original alias name, you will need to edit the formulas to use the new alias.

Locating files

When a database file is moved or renamed, Crystal Reports will not be able to find the data the next time the report is printed. On other occasions, a report may be created on one machine where all of the database data is stored in a certain directory, then the report is copied or moved to another machine that stores the same data in a different directory. In any of these events, you need to verify the location of the database files accessed by the report and reset the alias pointers to the new database location or name.

Locating Files



The VERIFY DATABASE command on the Database menu checks the alias pointers stored in a report file to verify that the database files expected are located in the indicated directories. If the databases are not found in the specified location, Crystal Reports notifies you of the discrepancies.

Use the SET LOCATION command on the Database menu to change the alias pointers stored by Crystal Reports. The SET LOCATION command provides a simple way to indicate the new name or location of database files. In addition, the SET LOCATION command can be used to change the ODBC data source used by a report. See *How to change the ODBC data source accessed by a report*, Page 461.

Indexed tables

Creating indexes for database tables can increase the speed of data access and reduce the time it takes for Crystal Reports to evaluate data. Some DBMS applications automatically index your database tables, while others require that you create an index yourself. For the best report generation performance, make sure each of your database tables has a corresponding index.

NOTE: Some DBMS applications do not support indexed tables. Refer to the documentation for your DBMS to find out if it supports indexes and how to create them. If your DBMS documentation does not mention indexed tables, it may not support them, and you should link tables based on common fields. The Crystal Reports Visual Linking Expert can also help you determine if your tables include indexes. Search for Visual Linking Expert in Crystal Reports online Help.

Indexes organize the records in a relational database table so that data can be located easier. For example, assume you have a table with the following data:

Order#	Customer	Amount
10444	Allez Distribution	25141.50
10470	BG Mountain Inc.	19164.30
10485	Sierra Mountain	8233.50
10488	Mountain Toad	24580.50
10495	SFB Inc.	7911.80
10501	La Bomba de Bicicleta	1956.20
10511	BG Mountain Inc.	1683.60

Order#	Customer	Amount
10544	Sierra Bicycle Group	19766.20
10568	Mountain Tops Inc.	29759.55
10579	Sierra Bicycle Group	12763.95

The information in this table is organized according to the Order# field. This is fine anytime you want to look up information in the table based on order numbers. However, what if you wanted to look up information specific to a certain customer?

Let's say you want to look up all orders made by Sierra Bicycle Group. The database engine must begin by looking at the first order number in the list and checking to see if the customer name matches the request. If not, it goes to the second order number, and checks that customer name. When an order number is reached that contains the correct customer name, the database engine retrieves the information, then continues to the next order number. Using this technique, both the Order# field and the Customer field must be read for every single record in the table. This takes a long time and a large amount of computer processing effort for examining extensive database tables with thousands, or even millions of records.

Instead, you can create an index for the table based on the Customer field. Such an index might look like this:

Customer	Pointer to Order#
Allez Distribution	10444
BG Mountain Inc.	10470
BG Mountain Inc.	10511
La Bomba de Bicicleta	10501
Mountain Toad	10488
Mountain Tops Inc.	10568
SFB Inc.	10495
Sierra Bicycle Group	10544
Sierra Bicycle Group	10579
Sierra Mountain	10485

In this index, information is organized by customers, not order numbers. Also, notice that the second column actually contains pointers to specific order numbers in the original table. By using

this index, the database engine can search just the information in the Customer column until it finds the customer you're interested in, Sierra Bicycle Group.

For each correct customer entry the database engine finds in the index, it looks up the matching order in the table according to the pointer in the second column of the index. Only the orders for the correct customer are read. Finally, since information in the index is organized according to the customer names, the database engine does not need to continue searching through the index or the table as soon as it finds an index entry that does not match the requested customer.

The result of this highly organized search through a database table according to an index is speed. Using indexes speeds up data retrieval and report generation, important factors when reporting on large database files.

Linking tables

You link tables so records from one table will match related records from another. For example, if you activate an Orders table and a Customers table, you link the tables so that each order (from the Orders table) can be matched up with the customer (from the Customer table) that made the order.

When we link, we use a field that is common to both tables. Crystal Reports uses the link to match up records from one table with those from the other. In our example, the link assures that the data in each row of the report refers to the same order.

Link from and link to

When you link two tables, you link *from* one table *to* another table. When linking tables, you must understand this concept. The *from* table is used as a primary table, while the *to* table acts as a lookup table where records are looked up by the primary table. In a simple link, Crystal Reports examines the first record in the primary table and finds all matching records in the lookup table. Once all matches have been found in the lookup table for the first record in the primary table, all matches in the lookup table for the next record in the primary table are found.

NOTE: Crystal Reports can link two records based on a partial match of string data. This is called a partial link. To enable partial linking in

Crystal Reports, check the Allow partial text matches check box in the Link Options dialog box. As an example of a partial link, a record with a field value of "Chris" can link to a record with a field value of "Christopher". However, partial linking only works when the value in the lookup table is longer than the value in the primary table. In other words, the value "Chris" can link to the value "Christopher", but the value "Christopher" can not link to the value "Chris".

Link relationships

When you link records from one table to another table, the records will typically fall under one of two relationship types: one-to-one or one-to-many.

One-to-one relationships

In a one-to-one relationship between records in two linked tables, for every record in the primary table, there is only one matching record in the lookup table, based on the linked fields. For example, in the CRAZE.MDB database, the sample database shipped with Crystal Reports, the Employee table can be linked to the Employee Addresses table based on the Employee ID field in each table. The Employee table contains information about employees at the company, the positions they hold, their salaries, and hiring information. The Employee Addresses table contains each employee's home address. There is only one record for each employee in each of these tables. Therefore, if the Employee table is linked to the Employee Addresses table, only one record will be found in the Employee Addresses table for each record in the Employee table. This is a one-to-one relationship.

One-to-many relationships

In a one-to-many relationship between records in two linked tables, for every record in the primary table, there may be more than one matching record in the lookup table, based on the linked fields. In the CRAZE.MDB database, the Customer table can be linked to the Orders table based on the Customer ID field in each table. The Customer table contains information about each customer that has placed an order with the company. The Orders table contains information about orders that customers have placed. Since customers can place more than one order, there may be more than one record in the Orders table for each customer record in the Customers table. This is a one-to-many relationship.

Performance considerations in one-to-many links

The information provided in this section is intended to help you maximize processing time and minimize network traffic when you're running your reports. You will learn about the best ways to use selection formulas and indexes in one-to-many situations to make your reporting more efficient. If you do not use the information in this section, your reports may end up processing dozens or even hundreds of times more records than necessary.

When a one-to-many situation exists between two database tables and the program matches up records from the tables, there are a number of factors that determine how many records the program reads and evaluates. Understanding this information will help you avoid those situations that require excessive processing time or that generate unnecessary network traffic.

The charts that follow show the effects of the different factors on the number of records the program ultimately has to read. The charts are based on these assumptions.

- Table A contains 26 records (one for each letter in the alphabet).
- Table B contains 2600 records (100 matching records for every record in Table A).
- The scenario is to produce a report that finds two specific records in Table A and the 200 records (100+100) in Table B that match those two records in Table A.
 - In a best case scenario, the program would only have to read about 200 records to accomplish the task.
 - In a worst case scenario the program would have to read about 67,600 records to accomplish the same task.

NOTE: The performance considerations for data files are different from the considerations for SQL databases. A data file is any non-SQL database that is accessed directly from Crystal Reports. For the purpose of this discussion, a SQL database is any SQL database accessed directly from Crystal Reports or through ODBC as well as any other database types that are accessed through ODBC. For a better understanding of the difference between direct access databases and ODBC data sources, see Chapter 20, Data Sources, Page 475.

Extended descriptions of chart columns

The performance charts use the following columns:

Linking or Subreport

Are you creating a report from linked databases or are you inserting a subreport and binding it to the data in your primary report?

Selection Formula

Does your primary report include a record selection formula that sets range limits on the key (indexed) field in Table A?

Index A

Is table A indexed on the field you are going to use to match up the records?

Index B

Is table B indexed on the field you are going to use to match up the records?

Reads A

How many records does the program have to read out of Table A to find the two records it is looking for?

For each A reads in B

How many records does the program have to read in Table B to find the 200 records it is looking for?

Total Records Read

What is the total number of records the program has to process to complete the task?

<i>PC Data</i>						
Linking/ Subreport	Selection Formula	Index A	Index B	Reads A	For each A reads in B	Total Records Read
Linking	No	Yes or No	Yes	26	100 (26*100)	2600
Linking	Yes	No	Yes	26	100 (26*100)	2600
Linking	Yes	Yes	Yes	2	100 (2*100)	200

PC Data						
Subreport	No	No	No	26	2600 (26*2600)	67,600
Subreport	No	Yes	No	2	2600 (26*2600)	67,600
Subreport	No	Yes	Yes	26	100 (26*100)	2600
Subreport	Yes	No	No	2	2600 (2*2600)	5200
Subreport	Yes	No	Yes	26	100 (26*100)	2600
Subreport	Yes	Yes	Yes	2	100 (2*100)	200

SQL Data				
Linking/ Subreport	Selection Formula	Reads A	For each A reads in B	Total Records Read
Linking	No	26	100 (26*100)	2600
Linking	Yes	2	100 (2*100)	200
Subreport	No	26	100 (26*100)	2600
Subreport	Yes	2	100 (2*100)	200

Data File Considerations

When working with data files, one-to-many links can occur when linking tables in a single report or when adding a subreport to your report.

Linking Data Files

Here is the process the program follows in retrieving data from linked data files in one-to-many situations.

- If there is a selection formula, the program parses the selection formula and passes what it can down to the database DLL. This is generally range limit information. For example, consider the following record selection formula:

```
{Customer.Region} in "CA" to "IL" AND  
Remainder ({Customer.Customer ID}, 2) = 0
```

In this formula, the part before the AND

```
{Customer.Region} in "CA" to "IL"
```

contains range selection criteria on the Region field. The region must fall alphabetically between "CA" and "IL". The program passes this kind of condition down to the database DLL (for PC data) or the server (for SQL data). See Chapter 14, *Record and Group Selection*, Page 315.

The second half of the selection formula, however,

```
Remainder ({Customer.Customer ID}, 2) = 0
```

requires processing that must be done in the Report Engine. It uses a Crystal Reports function to manipulate and evaluate a field value and it can not be done in the database DLL or the server. The program does not pass this condition to the database DLL.

- If there is an index on Table A, and the range limit selection condition is based on the indexed field {Customer.Region} in our example, the program goes directly to the record it is seeking in Table A (the first CA record) and reads it.
 - For that record, it locates the first matching record in Table B, using the Table B index.
 - It passes this merged record (A+B) back to Crystal Reports which tests it against the entire selection formula.
 - It then reads the second matching record and passes the merged record on, then the third matching record, and so on until it has read all of the matching records.
 - Then it returns to Table A and reads the next record. There is no need to test the record to see if it meets the CA condition; it does because the field is indexed and the records are in alphabetical order. But it tests the record to see if it goes beyond the "IL" condition (for example, could the next record be from Mississippi or Tennessee?). If the record is still within the specified range, it begins the matching process again for that record.
 - It continues the process until it has located all targeted Table A records and the matching Table B records.

To find two records in Table A and the 100 records in Table B

that match the Table A records, the program reads 200 records.

NOTE: The lookup table in a link (Table B) must always be indexed or you will not be able to link the tables.

- If there is no index on Table A, or if there is an index but the range limit selection condition is not based on the indexed field, the program reads the very first record that it finds.
 - For that record, it locates the first matching record in Table B, using the Table B index.
 - It passes this merged record (A+B) back to the print engine which tests it against the entire selection formula.
 - It then locates the second matching record in Table B and passes that merged record back, then the third record and so on until it has located, merged, and passed back all the records in Table B that match the first record in Table A.
 - It then returns to the next record in Table A and begins the matching and merging process all over again.

To find two records in Table A and the 100 records in Table B that match the Table A records, the program reads 2600 records.

Subreports and Data Files

If your Primary report is based on Table A and the subreport is based on Table B and the records are linked, your primary considerations are as follow:

- The number of subreports that get run is determined by the index and selection formula situation in the primary report.
 - If Table A is indexed and if the primary report has a selection formula that passes down range limit conditions for the indexed field, the program runs 2 subreports. (See PC Databases - Linking for a discussion of these selection formulas.)
 - If Table A is not indexed, or if Table A is indexed but the selection formula does not pass down range limit conditions for the indexed field, the program runs 26 subreports.
- The number of records that get read for each subreport is

determined by the index situation on table B.

- If you have an index on table B, the program will read only the matching records (100) each time it runs a subreport.
- If you don't have an index on Table B, the program will always read every record in Table B (2600) every time it runs a subreport.

SQL Database Considerations

Since indexes aren't critical with SQL data, your primary concern with both linked tables and with subreports is whether or not there is a selection formula in the primary report that puts range limits on Table A. (See *Linking Data Files*, Page 415, for a discussion of these selection formulas.)

Linked SQL Tables

If there are range limit conditions in the selection formula, the program passes those conditions down to the server.

- If there is a selection formula that puts range limits on Table A, the server locates the records in Table A that satisfy the selection criteria (2), matches them up with the appropriate records in Table B (100), and returns 200 merged records to the Report Engine.
- If there is no selection formula or if there is a selection formula that doesn't put range limits on Table A, the server matches up each record in Table A (26) with the appropriate records in Table B (100), and returns 2600 merged records to the Report Engine.

In both cases, the Report Engine then applies the entire selection formula to the merged records.

Subreports and SQL Databases

If you are creating a primary report from Table A and a subreport from Table B

- The number of subreports that get run is determined by the selection formula situation in the primary report.
 - If there is a selection formula and it passes down range limits on Table A, the program runs a subreport only for

- those records that satisfy range limit conditions (2).
- If there is no selection formula, or if the selection formula does not pass down range limits on Table A, the program runs a subreport for every record in table A (26).
 - The number of records read by each subreport is the same whether or not there was range limit selection on Table A. Each subreport will read only those records in Table B that match each record read in the primary report (Table A) (100).

Performance Considerations For All Reports

1. With both data files and SQL databases, the program parses the entire selection formula and passes down whatever it can, wherever it finds it in the formula. Thus, if it finds criteria it can pass, then criteria that it can not, then criteria that it can, it passes down the first part, skips the second, and then passes down the third.
 - In the case of data files, it passes the criteria that it can down to the database translation layer.
 - In the case of SQL databases, it passes the criteria that it can down to the server in the form of a WHERE clause.

While there are exceptions, as a general rule it can pass down any part of the record selection formula that compares a field with a constant. Typically, this means that it can pass down any kind of record selection criteria that can be set up in the Select Records Expert (*field* equal to, one of, less than, greater than, less than or equal, greater than or equal, between, starting with, or like *constant*).

There are two special selection formula situations that need to be considered. In these situations, there are multiple conditions in the record selection formula and some can be passed down while others can not.

- AND situations

```
{customer.REGION} = "CA" AND  
{customer.CUSTOMER ID}[3 to 5] = "777"
```

- In this situation, the program sees that it can pass down the condition before the AND but not the condition after. Since the only records that will meet the second

condition will have to meet the first as well, it passes down the first condition, retrieves the data set that satisfies the condition, and then applies the second condition to only the retrieved data. The rule for AND situations is that the program passes down whatever conditions it can.

NOTE: If all of the conditions in an AND situation can be satisfied on the server or in the database DLL, the program passes them all down.

- OR situations

```
{customer.REGION} = "CA" OR  
{customer.CUSTOMER ID}[3 to 5] = "777")
```

- In this situation, the program also sees that it can pass down the condition before the OR but not the condition after. Since there are records that can satisfy the second condition without satisfying the first, passing the first condition down doesn't make any sense because it will retrieve an incomplete data set. In other words, even if it retrieves all the data that satisfies the first condition, it will still have to retrieve all the data in the table(s) so it can apply the second condition in Crystal Reports. Thus, instead of duplicating parts of the data retrieval, the program passes nothing down. It retrieves all the data and then runs both tests in Crystal Reports. The rule for OR situations is that the program passes down all the tests or none of the tests.

NOTE: If all of the tests in an OR situation can be performed on the server or in the database DLL, the program passes them all down.

2. To make certain the program can use the index on table A to enhance performance, make certain that:
 - there is a selection formula,
 - there are range limits in the selection formula on the key (indexed) field in Table A, and
 - the *Use Indexes* option is on in the File Options dialog box.

3. If the fields you are using from Table A are not indexed, but there is an indexed field that you can use in your record selection request, use it. For example, assume that you have three products (Product 1, Product 2, and Product 3) and you want to identify all sales of Product 2 in the US. There is no index on the Product field but there is an index on the Order Date field. Since you know that Product 2 did not begin shipping until July of 1995, you can get some speed improvement by limiting your report to orders in and after July 1995 using the selection formula. In such a case, the program uses the Order Date index to retrieve only those orders from July 1995 on (a small subset of the entire database) and then searches for the occurrences of Product 2 in that subset, not in the entire database.

The Visual Linking Expert

In Crystal Reports, the Visual Linking Expert lets you easily link two or more tables together. When you select the ADD DATABASE TO REPORT command from the Database menu and select an additional database table, the Visual Linking Expert appears and displays the additional database table.

The easiest way to link database tables is to click the Smart Linking button in the Visual Linking dialog box. Smart Linking automatically chooses links for your tables based on common fields in tables or indexed fields (if your database supports indexed fields).

Linking indexed tables

When you are linking direct access database tables, you must link from the primary table to an indexed field in the lookup table. The link field in the primary table can be indexed, but does not have to be. The link field in the lookup table, however, must be indexed.

In addition, the fields you use to link two tables must have the same data type. For example, you link a string field in one table to a string field in another table, or a numeric field in one table to a numeric field in another table, but you can not link a numeric field in one table to a string field in another table.

NOTE: Some DBMS applications allow you to convert the value in a field to another data type in the index. For instance, the field in

the table can be numeric, while the index converts the field value to a string. However, if you use that field to link to another table, you must link to a field of the original data type. You can not link a string value to the numeric field that has been converted to a string in the index.

NOTE: If you are linking tables from two different ODBC data sources, MS SQL Server and Oracle, for example, you can only use string fields to link the tables. String fields are stored in databases in the same manner, regardless of the data source. Other types of values, however, may not be stored the same way in different data sources, so you can not link different data sources in Crystal Reports using anything other than string values.

Changing index used in linking

When you use the Smart Linking feature to link tables using a field that is a component of multiple indexes (two or more), Crystal Reports selects one of the indexes for the link. That index may or may not be the one you want to use. To determine the index in use and to change it if you wish, you use the Index section of the Link Options dialog box. To call up the dialog box, either:

- double-click the link line of interest,
- select the link line of interest and click the *Options* button at the bottom of the Visual Linking Expert, or
- select the link line of interest, right-click, and select **OPTIONS** from the shortcut menu that appears.

The Index section of the Link Options dialog box has two parts:

- An *Index In Use* drop down box displays the index that is currently in use. If you click the arrow, it also lists the other indexes that are available for the link as well as the option <no specific index>. If you are using an indexed database table and you do not see a particular index that you expect to see on the list, use the *Add Index* button.
- A *Fields in Index* list box lists the fields that are included in the index that is currently selected in the *Index In Use* box.

If you select the <no specific index> option, the program will select an index for you the next time you print the report to the Preview Tab.

NOTE: Not all DBMS applications support indexed tables. Verify that your database uses indexes before trying to select an index for linking. Refer to your DBMS documentation to find out if your DBMS can use indexes and how to create them.

Methods of looking up tables (direct access databases)

When a single table is linked to two or more tables, Crystal Reports needs to know in what order it should look up and link data from the primary table to data in the second, third, etc. lookup table. Crystal Reports offers three different options for looking up records in two or more lookup tables from a single primary table:

- Look up both files at the same time.
- Look up all of one file, then all of the other.
- Look up all combinations of the two files.

NOTE: These options are not available if you are using data from an ODBC data source.

These options are only available when you have a single table, a primary table, that is linked to two or more lookup tables. The primary table must be the *link from* table in each of the links. For instance, if you link from the Customer table to the Orders table and from the Customer table to the Credit table, these look-up options are available. However, if you link from the Customer table to the Orders table and from the Credit table to the Customer table, these options are not available.

In the sections that follow, for each method demonstrated, three fields from three different linked tables are shown. In each case, the Customer table is linked to the Credit table and the Orders table. The fields displayed are the Customer Name field from the Customer table, the Amount field from the Credit table, and the Order Amount field from the Orders table. These are not necessarily link fields for the tables, but the data in these fields illustrates how data is retrieved using each of the three lookup methods.

Look up both at the same time

For each record in the Customer table, this option looks for a matching record in the Credit table and a matching record in the

Orders table. Then it looks for the next matching record in the Credit table and the next matching record in the Orders table, etc. Once it finds all the matching records, it repeats the process with the next record in the Customer table.

<i>Customer Table</i>	<i>Credit Table</i>	<i>Orders Table</i>
<i>Customer Name</i>	<i>Amount</i>	<i>Order Amount</i>
Cyclists Incorporated	(\$1088.56)	\$1529.70
Cyclists Incorporated	(\$1260.12)	\$23.50
CyclePath Corp.	(\$1958.03)	\$49.50
CyclePath Corp.	(\$1076.43)	\$1702.60
CyclePath Corp.	(\$75.04)	
The Great Bike Shop	(\$138.98)	\$3269.70
The Great Bike Shop		\$5219.55
The Great Bike Shop		\$1538.20

Notice that for each line in the report, the value in the Amount field of the Credit table does not necessarily have any connection to the value in the Order Amount field of the Orders table. However, for each record in the Customer table, one record is selected from the Credit table, and one record is selected from the Orders table at the same time.

Also notice that when the Credit table runs out of records for a given record in the Customer table, blanks are left in place of values until all related records from the Orders table are found. The same holds true when the Orders table runs out of records, but the Credit table still has some.

Look up all of one, then all of others (A to B, A to C)

For each record in the Customer table, this option looks for all the matching records in the Credit table (table B) and then all the matching records in the Orders table (table C). Then it repeats the process with the next record in the Customer table, then the next, etc.

<i>Customer Table</i>	<i>Credit Table</i>	<i>Orders Table</i>
<i>Customer Name</i>	<i>Amount</i>	<i>Order Amount</i>
Cyclists Incorporated	(\$1088.56)	
Cyclists Incorporated	(\$1260.12)	
Cyclists Incorporated		\$1529.70
Cyclists Incorporated		\$23.50
CyclePath Corp.	(\$1958.03)	
CyclePath Corp.	(\$1076.43)	
CyclePath Corp.	(\$75.04)	
CyclePath Corp.		\$49.50
CyclePath Corp.		\$1702.60
The Great Bike Shop	(\$138.98)	
The Great Bike Shop		\$3269.70
The Great Bike Shop		\$5219.55
The Great Bike Shop		\$1538.20

NOTE: *If you want table C data (the Orders table in this example) to appear in your report before table B data (Credit table), you will need to change your links so the A to C link comes first, then the A to B link. You do this via the Visual Linking Expert dialog box. To change the order of the links, delete the existing links and set up new links in the order you want.*

Look up all the combinations of the two files

For each record in the Customer table, this option looks for a matching record in the Credit table, then it finds all the matching records in the Orders table. Once it finds all the matching records in the Orders table, it repeats the process with the next record in the Credit table, then the next, etc. When it finds matching Orders records for all the Credit records that match the first Customer record, it moves to the next Customer record and repeats the process.

<i>Customer Table</i>	<i>Credit Table</i>	<i>Orders Table</i>
<i>Customer Name</i>	<i>Amount</i>	<i>Order Amount</i>
Cyclists Incorporated	(\$1088.56)	\$1529.70
Cyclists Incorporated	(\$1088.56)	\$23.50
Cyclists Incorporated	(\$1260.12)	\$1529.70
Cyclists Incorporated	(\$1260.12)	\$23.50
CyclePath Corp	(\$1958.03)	\$49.50
CyclePath Corp	(\$1958.03)	\$1702.60
CyclePath Corp	(\$1076.43)	\$49.50
CyclePath Corp	(\$1076.43)	\$1702.60
CyclePath Corp	(\$75.04)	\$49.50
CyclePath Corp	(\$75.04)	\$1702.60
The Great Bike Shop	(\$138.98)	\$3269.70
The Great Bike Shop	(\$138.98)	\$5219.55
The Great Bike Shop	(\$138.98)	\$1538.20

NOTE: If you want the program to look up the first matching record in table C (Orders table in this example), then find all matching records in table B (Credit table), the reverse of the current process, you will need to change your links so the A to C link comes first, then the A to B link. You do this via the Visual Linking Expert dialog box. To change the order of the links, delete the existing links and set up new links in the order you want.

SQL join types (ODBC data sources)

Crystal Reports enables you to specify the type of join you want to use when SQL tables are linked. An SQL join indicates how linked fields in two SQL tables are compared when records are read. The SQL Join Type options are specified in Crystal Reports using the Link Options dialog box.

NOTE: When linking using SQL joins, no indexed fields are required.

The join types are:

- Equal join [=]
- Left outer join [= (+), *=]
- Right outer join [(+) =, =*]
- Greater join [>]
- Less join [<]
- Greater or Equal join [>=]
- Less or Equal join [<=]
- Not Equal join [!=]

Each of these joins is discussed below, complete with SQL syntax and example result sets. If you are not familiar with SQL syntax, see *The SQL language*, Page 441.

NOTE: *In the sample SQL statements that appear in the following sections, each join is specified in the WHERE clause of the SQL statement.*

Equal to [=] join

The result set from an Equal join includes all the records where the linked field value in both tables is an exact match. In the following example, the Customer table is linked to the Orders table by the Customer ID field. When a Customer ID is found in the Orders table that matches a Customer ID in the Customer table, information is displayed for the corresponding records in both tables.

SQL uses the following syntax to describe an equal join:

```
SELECT Customer.'Customer ID',  
       Customer.'Customer Name',  
       Orders.'Order Amount'  
FROM 'Customer' Customer,  
     'Orders' Orders  
WHERE Customer.Customer ID =  
       Orders.Customer ID
```

This statement produces the following data:

<i>Customer Table</i>	<i>Customer Table</i>	<i>Orders Table</i>
<i>Customer ID</i>	<i>Customer Name</i>	<i>Order Amount</i>
52	Allez Distribution	25141.50
53	BG Mountain Inc.	19164.30
53	BG Mountain Inc.	1683.60
57	Hansen MTB Inc.	15716.40
58	La Bomba de Bicicleta	1956.20
60	Mountain Toad	24580.50
62	SFB Inc.	7911.80
63	Sierra Bicycle Group	19766.20
63	Sierra Bicycle Group	12763.95
64	Sierra Mountain	8233.50

Left Outer [= (+), * =] join

The result set from a Left Outer join includes all the records where the linked field value in both tables is an exact match. It also includes a row for every record in the primary (left) table whose linked field value has no match in the lookup table. For instance, we can use a Left Outer join to view all customers and the orders they have placed, but we also get a row for every customer that has not placed any orders. These customers appear at the end of the list with blanks in the fields that would hold order information:

<i>Customer Table</i>	<i>Customer Table</i>	<i>Orders Table</i>
<i>Customer ID</i>	<i>Customer Name</i>	<i>Order Amount</i>
52	Allez Distribution	25141.50
53	BG Mountain Inc.	19164.30
53	BG Mountain Inc.	1683.60
57	Hansen MTB Inc.	15716.40
58	La Bomba de Bicicleta	1956.20
60	Mountain Toad	24580.50
62	SFB Inc.	7911.80
63	Sierra Bicycle Group	19766.20

<i>Customer Table</i>	<i>Customer Table</i>	<i>Orders Table</i>
<i>Customer ID</i>	<i>Customer Name</i>	<i>Order Amount</i>
63	Sierra Bicycle Group	12763.95
64	Sierra Mountain	8233.50
54	Bicicletas Aztecas	
55	Deely MTB Inc.	

NOTE: *Left Outer and Right Outer joins are handled differently in the SQL language from other join types. If the database is accessed through ODBC, Crystal Reports will use ODBC syntax in the SQL statement. If you are connecting to a SQL database directly (not through ODBC), Crystal Reports will use a syntax native to the database. For complete information on what an Outer join looks like in a SQL statement, refer to Microsoft ODBC documentation or the documentation for your SQL database.*

Right Outer [(+)=, =*] join

The result set from a Right Outer join includes all the records where the linked field value in both tables is an exact match. It also includes a row for every record in the lookup (right) table whose linked field value has no match in the primary table. If we link the Customer table to the Orders table, we get a row in our table for every order a customer has placed, just like an Equal join. We also get a row for every order found that can not be linked to a customer. Theoretically, this shouldn't happen, but if an inexperienced sales person forgot to assign a customer ID to an order, you can quickly locate that order with a Right Outer join. The resulting table leaves a blank in any Customer fields for the order without a customer:

<i>Customer Table</i>	<i>Orders Table</i>	<i>Orders Table</i>
<i>Customer ID</i>	<i>Order ID</i>	<i>Order Amount</i>
52	6	25141.50
53	11	19164.30
53	21	1683.60
57	4	15716.40
58	20	1956.20
60	16	24580.50

<i>Customer Table</i>	<i>Orders Table</i>	<i>Orders Table</i>
<i>Customer ID</i>	<i>Order ID</i>	<i>Order Amount</i>
62	19	7911.80
63	28	19766.20
63	32	12763.95
64	14	8233.50
	25	10320.87

NOTE: *Left Outer and Right Outer joins are handled differently in the SQL language from other join types. If the database is accessed through ODBC, Crystal Reports will use ODBC syntax in the SQL statement. If you are connecting to a SQL database directly (not through ODBC), Crystal Reports will use a syntax native to the database. For complete information on what an Outer join looks like in a SQL statement, refer to Microsoft ODBC documentation or the documentation for your SQL database.*

Greater Than [>] join

The result set from a Greater Than join includes all records in which the linked field value from the primary table is greater than the linked field value in the lookup table. As an example, a company may want to compare the salaries made by all of their sales representatives to the salaries made by all of their sales managers. The company executives want to make sure no sales representative is making more money than any manager. With this in mind, you can link the SalesRep table to the Manager table, by the Salary field in each table, using a Greater Than join:

```
SELECT SalesRep.'Last Name',
       SalesRep.'Salary',
       Manager.'Last Name',
       Manager.'Salary'
FROM 'SalesRep' SalesRep,
     'Manager' Manager
WHERE SalesRep.'Salary' >
      Manager.'Salary'
```

This SQL statement might produce data similar to this:

<i>SalesRep Table</i> <i>Last Name</i>	<i>SalesRep Table</i> <i>Salary</i>	<i>Manager Table</i> <i>Last Name</i>	<i>Manager Table</i> <i>Salary</i>
Davolio	\$35,000.00	Fuller	\$32,000.00
Davolio	\$35,000.00	Brid	\$30,000.00
Davolio	\$35,000.00	Buchanan	\$29,500.00
Dodsworth	\$48,300.00	Hellstern	\$45,000.00
Dodsworth	\$48,300.00	Fuller	\$32,000.00
Dodsworth	\$48,300.00	Brid	\$30,000.00
Dodsworth	\$48,300.00	Buchanan	\$29,500.00
Dodsworth	\$48,300.00	Martin	\$35,000.00
Patterson	\$30,000.00	Buchanan	\$29,500.00

In this table, there is no relationship between sales representatives and sales managers established. Since all managers have seniority over all sales representatives, a company might find it necessary to check if any representatives make more money than any managers, evidence of a salary problem that needs to be remedied.

Less Than [<] join

The result set from a Less Than join includes all records in which the linked field value from the primary table is less than the linked field value in the lookup table. Using the Less Than join, we can compare sales representative and manager salaries in a different direction. Once again, the Salary field in each table is used as the link field. This time, though, we link from the Manager table to the SalesRep table using a Less Than join on the linked Salary fields:

```
SELECT Manager.'Last Name',
       Manager.'Salary',
       SalesRep.'Last Name',
       SalesRep.'Salary'
FROM 'Manager' Manager',
     SalesRep' SalesRep
WHERE Manager.'Salary' <
       SalesRep.'Salary'
```

This SQL statement produces a slightly different table than the Greater Than join:

<i>Manager Table</i>	<i>Manager Table</i>	<i>SalesRep Table</i>	<i>SalesRep Table</i>
<i>Last Name</i>	<i>Salary</i>	<i>Last Name</i>	<i>Salary</i>
Fuller	\$32,000.00	Davolio	\$35,000.00
Fuller	\$32,000.00	Dodsworth	\$48,300.00
Brid	\$30,000.00	Davolio	\$35,000.00
Brid	\$30,000.00	Dodsworth	\$48,300.00
Buchanan	\$29,500.00	Davolio	\$35,000.00
Buchanan	\$29,500.00	Dodsworth	\$48,300.00
Buchanan	\$29,500.00	Patterson	\$30,000.00
Martin	\$35,000.00	Dodsworth	\$48,300.00
Hellstern	\$45,000.00	Dodsworth	\$48,300.00

Greater Than Or Equal [>=] join

The result set from a Greater Than Or Equal join includes all records in which the linked field value from the primary table is greater than or equal to the linked field value in the lookup table. The example here is identical to the example for the Greater Than join, but it uses the Greater Than Or Equal join:

```
SELECT SalesRep.'Last Name',
       SalesRep.'Salary',
       Manager.'Last Name',
       Manager.'Salary'
FROM 'SalesRep' SalesRep,
     'Manager' Manager
WHERE SalesRep.'Salary' >=
      Manager.'Salary'
```

This statement might produce data such as this:

<i>SalesRep Table</i>	<i>SalesRep Table</i>	<i>Manager Table</i>	<i>Manager Table</i>
<i>Last Name</i>	<i>Salary</i>	<i>Last Name</i>	<i>Salary</i>
Davolio	\$35,000.00	Fuller	\$32,000.00
Davolio	\$35,000.00	Brid	\$30,000.00
Davolio	\$35,000.00	Buchanan	\$29,500.00
Davolio	\$35,000.00	Martin	\$35,000.00
Dodsworth	\$48,300.00	Hellstern	\$45,000.00

<i>SalesRep Table</i>	<i>SalesRep Table</i>	<i>Manager Table</i>	<i>Manager Table</i>
<i>Last Name</i>	<i>Salary</i>	<i>Last Name</i>	<i>Salary</i>
Dodsworth	\$48,300.00	Fuller	\$32,000.00
Dodsworth	\$48,300.00	Brid	\$30,000.00
Dodsworth	\$48,300.00	Buchanan	\$29,500.00
Dodsworth	\$48,300.00	Martin	\$35,000.00
Patterson	\$30,000.00	Brid	\$30,000.00
Patterson	\$30,000.00	Buchanan	\$29,500.00

Less Than Or Equal [<=] join

The result set from a Less Than Or Equal join includes all records in which the linked field value from the primary table is less than or equal to the linked field value in the lookup table. The example here is identical to the example for the Less Than join, but it uses the Less Than Or Equal join:

```
SELECT Manager.'Last Name',
       Manager.'Salary',
       SalesRep.'Last Name',
       SalesRep.'Salary'
FROM 'Manager' Manager',
     SalesRep' SalesRep
WHERE Manager.'Salary' <=
       SalesRep.'Salary'
```

This SQL statement produces data like the following:

<i>Manager Table</i>	<i>Manager Table</i>	<i>SalesRep Table</i>	<i>SalesRep Table</i>
<i>Last Name</i>	<i>Salary</i>	<i>Last Name</i>	<i>Salary</i>
Fuller	\$32,000.00	Davolio	\$35,000.00
Fuller	\$32,000.00	Dodsworth	\$48,300.00
Brid	\$30,000.00	Davolio	\$35,000.00
Brid	\$30,000.00	Dodsworth	\$48,300.00
Brid	\$30,000.00	Patterson	\$30,000.00
Buchanan	\$29,500.00	Davolio	\$35,000.00
Buchanan	\$29,500.00	Dodsworth	\$48,300.00
Buchanan	\$29,500.00	Patterson	\$30,000.00

<i>Manager Table</i>	<i>Manager Table</i>	<i>SalesRep Table</i>	<i>SalesRep Table</i>
<i>Last Name</i>	<i>Salary</i>	<i>Last Name</i>	<i>Salary</i>
Martin	\$35,000.00	Davolio	\$35,000.00
Martin	\$35,000.00	Dodsworth	\$48,300.00
Hellstern	\$45,000.00	Dodsworth	\$48,300.00

Not Equal [!=, <>] join

The result set from a Not Equal join includes all records in which the linked field value from the primary table is not equal to the linked field value in the lookup table. This type of join can be used to find possible combinations of items when a table is joined to itself (a self-join). For example, a company can have a table listing all products they sell. When they decide to hold a sale where their customers buy one item and get the second item half price, they may need a list of all possible two item combinations:

```
SELECT Product1.'Product Name',
       Product2.'Product Name',
FROM 'Product' Product1
     'Product' Product2
WHERE Product1.'Product Name' !=
       Product2.'Product Name'
```

In this SQL statement, the Product table is opened twice. The first time, it is given the alias name Product1. The second time, it is given the alias name Product2. Then, the Product Name field is used to link from the Product1 table to the Product2 table. This is the same table, but since it has been opened twice using different aliases, Crystal Reports will consider it as two separate tables. A Not Equal join is used to link the tables by the Product Name field. As a result, each product is paired with every other product offered, but is not paired with itself:

<i>Product1</i>	<i>Product2</i>
<i>Product Name</i>	<i>Product Name</i>
Craze Adult Helmet	Craze Mtn Lock
Craze Adult Helmet	InFlux Lycra Glove
Craze Adult Helmet	Roadster Micro Mtn Saddle
Craze Mtn Lock	Craze Adult Helmet

Product1	Product2
Product Name	Product Name
Craze Mtn Lock	InFlux Lycra Glove
Craze Mtn Lock	Roadster Micro Mtn Saddle
InFlux Lycra Glove	Craze Adult Helmet
InFlux Lycra Glove	Craze Mtn Lock
InFlux Lycra Glove	Roadster Micro Mtn Saddle
Roadster Micro Mtn Saddle	Craze Adult Helmet
Roadster Micro Mtn Saddle	Craze Mtn Lock
Roadster Micro Mtn Saddle	InFlux Lycra Glove

NOTE: The symbol != is used to represent a not equal to join if the ODBC data source driver for the data being accessed supports this symbol. If not, the default symbol <> is used to represent a not equal to join.

NOTE: For more information on self-joins, search for Self join in Crystal Reports Help.

Using SQL and SQL databases

Perhaps the most popular and most powerful database formats are DBMS applications based on the Structured Query Language (SQL). SQL databases usually work over a client/server network architecture, providing an SQL Server to create, store, and manipulate database files, tables, fields and records, and an SQL Client interface allowing workstation users to not only design and work with database files, but to also retrieve useful and meaningful data that will help them in their everyday work.

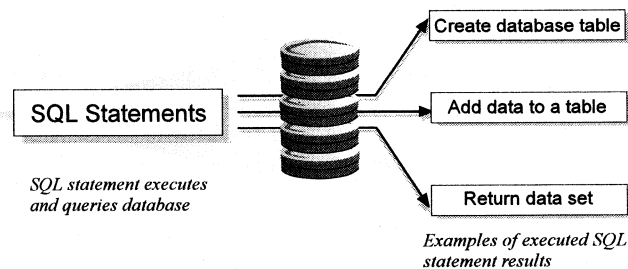
What is SQL?

SQL is a query language designed for organizing, managing, developing and querying large relational databases over computer networks. SQL is a common language among the Information Science (IS) and Information Management industry. The language has been standardized by the American National Standards Institute (ANSI) and the International Standards Organization (ISO), meaning that there are specific features that must be present in any version of SQL produced by a software company for that version to be officially called SQL. Many software vendors add more advanced features to their version of

SQL in an effort to improve the language and attract customers, but it must retain the original standards established by ANSI and ISO.

You should realize that SQL is not a true computer language. It can not be used to create stand-alone computer application or operating systems. SQL is often referred to as a sublanguage that can be used from within other languages or applications. Most importantly, the purpose of the SQL language is specific to working with relational databases.

The syntax of the SQL language is built on a system of sending SQL statements to the SQL database server. Each statement is a request to perform a database operation such as create a database file, add tables and fields to the database, add records to tables, or retrieve data from databases. The SQL server analyzes the SQL statement and performs the required operation. If the statement is a request for data, the server gathers the data and returns it to the client workstation for the user to view.

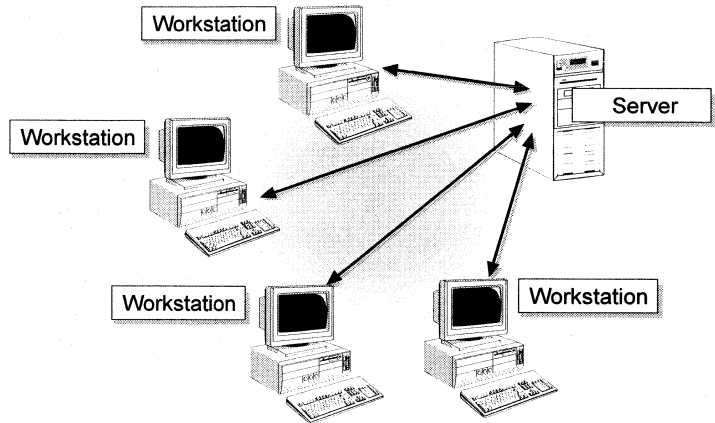


A SQL query is a SQL statement designed specifically to request data from one or more SQL databases. Some SQL applications require that you type in a SQL query directly using a text editor, while others provide graphical user interfaces that lead you through the process of querying a SQL database. In the latter case, the application must create a SQL statement based on the information you provide. This statement is the actual SQL query, and it is the SQL query that is actually used to request the data. Crystal Reports falls into this latter category of SQL-compliant applications.

Client/Server architecture

One of the most powerful features of SQL DBMS applications is their ability to efficiently use the client/server architecture of a network.

Client/Server Network

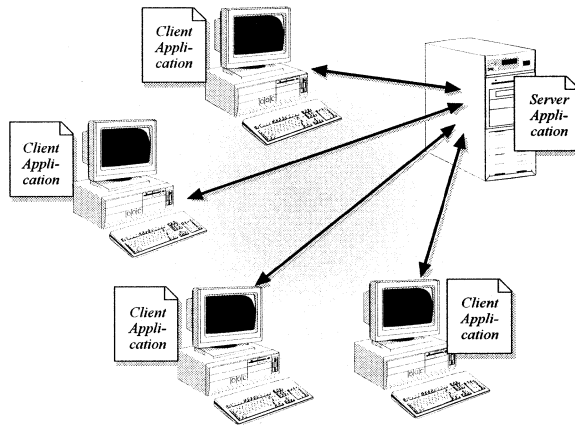


A simple network structure consists of one or more network servers that provide a common location for all users on the network to obtain data and applications. Many network servers also provide network security, automated services such as backing up data, and network resource monitoring to provide the best service possible to all workstations on the network. Because of the high processing demands required by a network server, the computer used as the server is often a high powered, fast machine that may contain multiple processors, multiple hard drives, and multiple CD-ROM drives.

A network client is a single computer workstation that is used regularly by one or more company employees. A user works on the client and accesses data and applications from the server over the network. Large processing jobs that require a lot of time and resources are handled by the server, and the finished results are sent back to the client. This provides more efficient time management for users because the local workstation has less processing time and more "up" time available to the user.

Many modern computer applications are based on this client/server architecture. A simple client/server application has two parts, a server based application that is located on a network server machine, and a client based application that is located on a user's workstation. The server application handles complex, time consuming, or power demanding processes, taking advantage of the network server computer's power and resources, while the client application provides an easy-to-use user interface designed to help get work done faster and better than it could be done otherwise.

Client/Server Application



Often, a client/server application will be made available with a certain number of seats, depending on how much is paid for the application. Each seat is a single client workstation, or a single client user, depending on what the software vendors chooses, that can be connected to and use the client/server software. Software vendors often sell additional seats for their applications, each seat coming with a complete set of client application software.

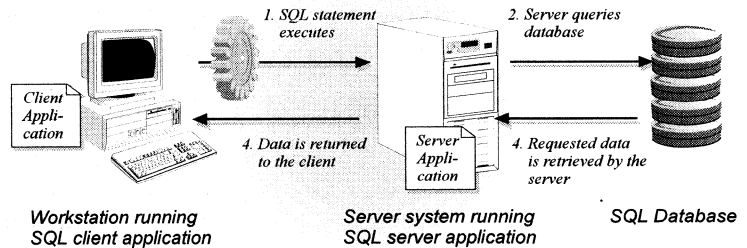
Do not confuse server applications with a network server computer. Both are often referred to as servers. However, a server application resides on a network server, taking advantage of the hardware and operating system capabilities of the server machine. A network server is a physical machine that network clients are connected to by cables or some other connection device.

The SQL DBMS

An SQL Database Management Systems is a common example of a client/server software package. A standard SQL DBMS will include an SQL server application that handles all of the actual work of building and working with databases and database data. The DBMS will also include at least one set of SQL client software (one seat) that can connect to the SQL server over your network. SQL client software usually consists of, at the very least, an SQL statement editor that you can use to write and execute SQL statements and an underlying communications layer that works with the SQL server application over the network.

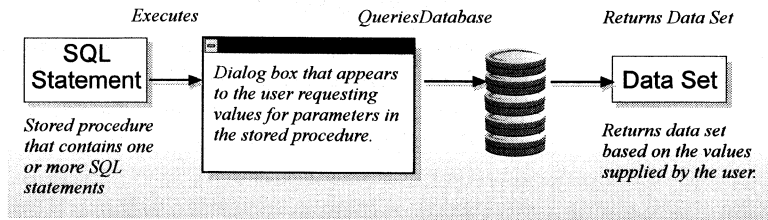
When you execute an SQL statement, the client software passes the statement to the communications layer, which sends the statement over the network to the server software. The SQL server analyzes the statement, performs the requested operation, and returns any data requested to the client software. If the client software receives back any data, it displays the data to the user.

SQL Client communicating with SQL server



Stored procedures

In addition to the common relational database attributes, tables, fields, records, etc., many SQL DBMS systems support stored procedures. A stored procedure is a compiled SQL program, consisting of one or more SQL statements. A stored procedure can be used to define a SQL query that you will use over and over again. Furthermore, variables, conditional expressions, and variable arguments can be defined in the stored procedure so that you are prompted to provide information before the procedure is executed.



Since stored procedures can return a result set, they can provide a specific set of data when executed. In fact, Crystal Reports allows you to execute a stored procedure on a SQL database and use the data returned to design a report. If the stored procedure is designed to prompt a user for information to base its query on, Crystal Reports will prompt you for that information when you select the stored procedure for your report. See *How to select a stored procedure from an SQL database and change stored procedure parameters*, Page 465.

How does Crystal use SQL?

When you connect to a SQL database, or any ODBC database, Crystal Reports acts as a SQL client application, connecting to your SQL server through your network.

When you design a report that accesses SQL data, Crystal Reports builds a SQL query. This query can be edited if know SQL and you feel that the query can be further optimized. If you choose the SHOW SQL QUERY command from the Database menu, the Show SQL Query dialog box displays the SQL query that Crystal Reports has designed.

This SQL query is a representation of the SQL statement that Crystal Reports sends to the SQL server. By interpreting as much as possible from your report design into a SQL query, Crystal Reports can offload much of the report processing onto the server machine. Rather than having to sift through and entire database to find the specific data you requested, Crystal Reports lets the server do the sifting and gets back a much smaller set of data, thus reducing the time and resources your workstation must use to finish the report.

The SQL language

Since Crystal Reports uses the SQL language to access client\server databases through ODBC, you can better understand the report generating process by understanding some of the SQL clauses (commands) used:

SELECT

The SELECT clause indicates specific data items to retrieve from database tables. The item retrieved may be the values in a database field (column) or it may be the result of a calculation performed while gathering the data. For example:

```
SELECT
    TABLEA. 'CUSTNAME' ,
    TABLEA. 'STATE'
```

DISTINCT

The DISTINCT clause can be added to a SQL statement just after the SELECT clause. DISTINCT forces the query to retrieve only unique (distinct) sets of data. A row of results will only be retrieved once. The previous SELECT statement can be changed to use the DISTINCT clause:

```
SELECT DISTINCT
    TABLEA. 'CUSTNAME' ,
    TABLEA. 'STATE'
```

FROM

The FROM clause specifies the sources of the database fields indicated in the SELECT clause. FROM lists actual database tables that contain the fields and records containing the requested data. The FROM clause generated by Crystal Reports precedes the name of each table with the alias it uses to identify the table in your report. The following illustrates the FROM clause with the SELECT clause:

```
SELECT
    TABLEA. 'CUSTNAME' ,
    TABLEA. 'STATE'
FROM
    'TABLEA' TABLEA
```

WHERE

The WHERE clause has two purposes:

- WHERE can specify record selection criteria.
- WHERE can specify how two database tables are joined.

When WHERE is used to specify record selection criteria, it specifies a search condition to use to determine which records (rows of data) are to be retrieved. For example:

```
SELECT
    MYTABLE. ' SALESPERSON' ,
    MYTABLE. ' SALESTOTAL'
FROM
    'MYTABLE' MYTABLE
WHERE
    MYTABLE. ' SALESTOTAL' < 10000.00
```

If WHERE is used to specify how two tables are linked, a SQL join operator sits between the two table names. See *SQL join types*, Page 426. The following is an example of the WHERE clause joining two tables:

```
SELECT
    CUSTOMER. ' CUST_ID' ,
    CUSTOMER. ' CUST_NAME' ,
    ORDERS. ' AMOUNT'
FROM
    'CUSTOMER' CUSTOMER,
    'ORDERS' ORDERS
WHERE
    CUSTOMER. ' CUST_ID' = ORDERS. ' CUST_ID'
```

ORDER BY

The ORDER BY clause indicates that the database records retrieved be sorted according to the values in a specific field. If the ORDER BY clause is not used, records are retrieved in the order that they appear in the original database. If more than one field is specified after the ORDER BY clause, the records are sorted according to the values in the first field specified, then, within that sort, they are sorted by the values in the second field, specified, and so on. The following SQL statement uses the ORDER BY clause:

```

SELECT
    MYTABLE.'COMPANY' ,
    MYTABLE.'CITY' ,
    MYTABLE.'STATE'
FROM
    'MYTABLE' MYTABLE
ORDER BY
    MYTABLE.'STATE' ASC,
    MYTABLE.'CITY' ASC

```

NOTE: *ASC indicates that the values in the field are sorted in ascending order rather than descending order (DESC). Ascending order sorts letters from A to Z and numbers from 0 to 9.*

GROUP BY

The GROUP BY clause retrieves a set of summary data. Instead of retrieving the data itself, GROUP BY groups the data and summarizes each group according to a SQL aggregate function. Only the summarization information for each group is returned to Crystal Reports. For example:

```

SELECT
    MYTABLE.'STATE' ,
    MYTABLE.'ZIPCODE' ,
    SUM (MYTABLE.'SALES')
FROM
    'MYTABLE' MYTABLE
GROUP BY
    MYTABLE.'STATE' ,
    MYTABLE.'ZIPCODE'

```

HAVING

The HAVING clause creates selection criteria for the summary information produced by the GROUP BY clause. The function of the HAVING clause is similar to the record selection function of the WHERE clause, but HAVING applies to summary data produced by the GROUP BY clause. For this reason, you will always see a GROUP BY clause just before the HAVING clause in any Crystal Reports SQL statement that produces summary results. This can be seen in the following example:

```
SELECT
    TABLE.'CUSTOMER',
    SUM (TABLE.'ORDERAMOUNT')
FROM
    'TABLE' TABLE
GROUP BY
    TABLE.'CUSTOMER'
HAVING
    SUM (TABLE.'ORDERAMOUNT') > 20000.00
```

For additional information

This chapter has only touched on some of the more important aspects of database access, relational databases, and SQL. If you are interested in learning more about any of these topics, refer to the documentation provided with your DBMS application.

In addition, there are hundreds of books available on the market that discuss database theory and design in depth. Look for the computer related section at your local bookstore.

HANDS-ON

This section provides step-by-step instructions for performing several common procedures related to accessing database files from within Crystal Reports. Use these procedures to get started working with Crystal Reports fast.

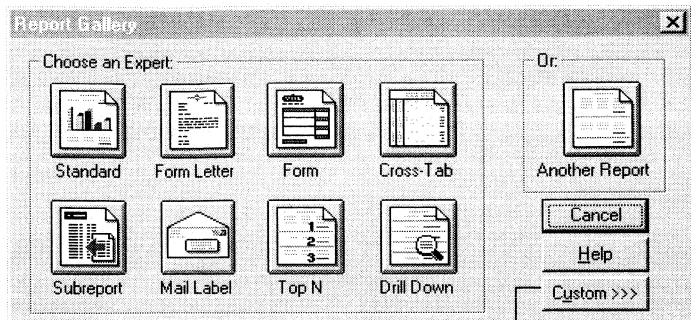
How to open Access queries through DAO

Microsoft Access queries can be used in Crystal Reports as separate data sets, just like Access tables. When opening an Access database through the DAO engine (Page 504), any queries in the database can be automatically read.

NOTE: *Only the 32-bit version of Crystal Reports ships with the DAO engine. This section assumes the 32-bit DAO engine is being used.*

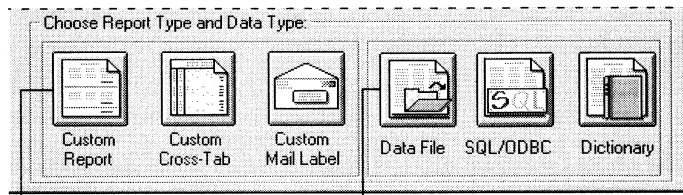


1. Click the NEW button on the toolbar in Crystal Reports. The Report Gallery appears.



2. Click the Custom button.

The Report Gallery will expand.



3. Click a Custom Report button.

4. Click the Data File button.

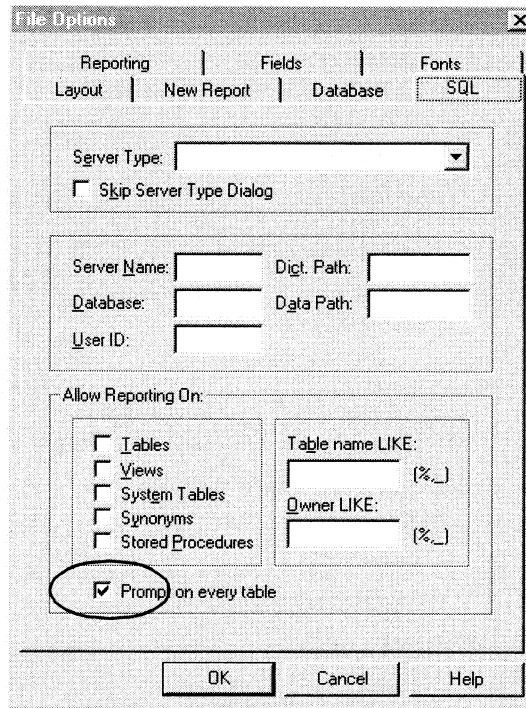
5. The Choose Database File dialog box appears.
6. Locate and select the Microsoft Access database (.MDB) file that contains the query you want to use in your report.
7. Click OK. The Design Tab appears in the Crystal Reports window with the Insert Fields dialog box.
8. All of the tables and queries from your Access database appear in the list box on the Database Tab of the Insert Fields dialog box. Queries appear at the end of the list, after tables. Locate your query on the Database Tab, and double-click on the name of the query in the list. The query expands to display all fields it contains.
9. Select any fields you want to use in your report, and add them to the Design Tab of Crystal Reports.

NOTE: You can not use Access action queries or update queries in Crystal Reports. You can use Access select queries and Cross-Tab queries.

How to open Access queries through ODBC

ODBC gives you more control over what parts of a database you want to use. For this reason, using an Access query through ODBC requires a few extra steps.

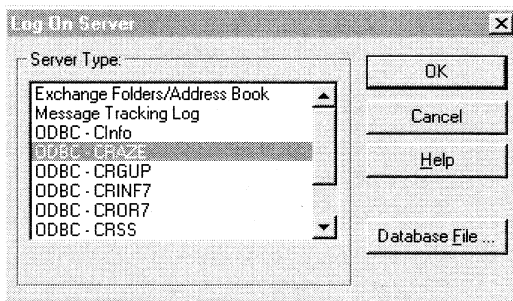
1. Choose the **OPTIONS** command from the File menu in Crystal Reports. The File Options dialog box appears. Click the **SQL** Tab to display SQL and ODBC options.



2. Make sure either the **Views** or the **Prompt on every table** check box is checked.

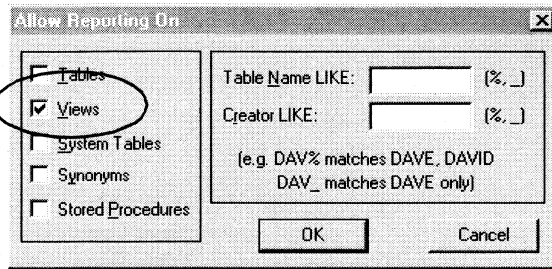
- The *Views* option automatically displays any available queries in your Access database.
- The *Prompt on every table* option displays the Allow Reporting On dialog box every time you log on to an ODBC data source. This dialog box provides the same options as the Allow Reporting On section of the File Options dialog box, but provides those options for every ODBC data source you log on to.

3. In addition, you can specify *Table name LIKE* and *Owner LIKE* options if you wish.
 - *Table name LIKE* is based on the SQL LIKE clause. This option allows you to specify the kinds of table names you want to appear in the Choose SQL Table dialog box. Use the underscore character (_) or the percent sign character (%) as wildcards with this function. The underscore character specifies any single character, while the percent sign signifies any character string. For example, DAV_ matches DAVE only, while DAV% matches DAVE and DAVID. *Table name LIKE C%* would display only those tables that have a table name beginning with the letter C.
 - *Owner LIKE* is also based on the SQL LIKE clause. The *Owner LIKE* option allows you to select the Owner (or Creator or Alias) of the table, not the table name itself. For example *Owner LIKE C%* would display only those tables that had an owner beginning with the letter C.
4. Click **OK** to exit the File Options dialog box.
5. Create a new report. When you choose SQL/ODBC as the source of your data, the Log On Server dialog box appears.



6. Choose the ODBC data source for your Access database, then click **OK**.

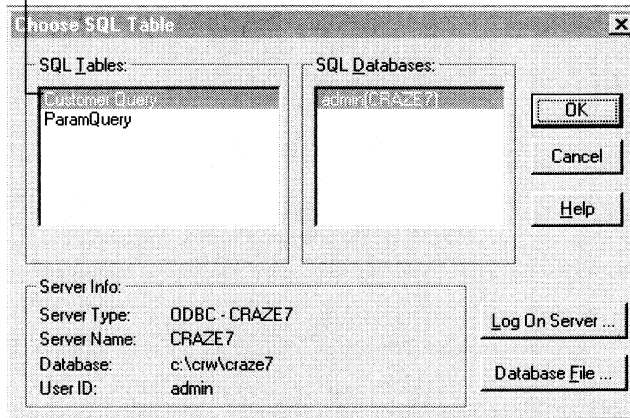
7. If you did not specify a particular Access database file with your Access ODBC data source, the Select Database dialog box will appear. Locate and select the database containing the Access query you want to use, and click **OK**.
8. If you selected the *Prompt on every table* check box in the File Options dialog box, the Allow Reporting On dialog box now appears. Otherwise, skip to Step 11.



9. Make sure the Views check box is selected and click OK when finished.

10. The Choose SQL Table dialog box appears.

11. Highlight your query in this list box, then click OK.



12. The Design Tab appears with the Insert Fields dialog box. Your Access query, and all fields associated with that query, appear on the Database Tab of the Insert Fields dialog box.

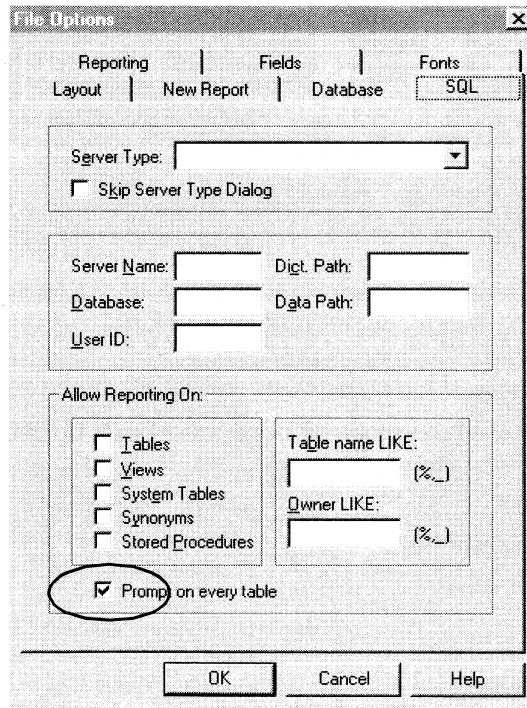
NOTE: You can not use Access action queries or update queries in Crystal Reports. You can use Access select queries and Cross-Tab queries.

How to open Access parameter queries

Access parameter queries can only be opened when an Access database is opened via ODBC. Make sure you have an ODBC data source set up for your Access database before attempting this procedure. See *How to set up an ODBC data source*, Page 453.

NOTE: When you design a parameter query in Access, you must provide a prompt for the query and specify a data type for the parameter. First, with your query open in Design View in Microsoft Access, enter a prompt in the Criteria cell for the field that will act as a parameter. Then, choose the PARAMETERS command from the Query menu in Access, and specify a data type for the parameter you just created. Make sure the prompt appears exactly as it does in the Criteria cell. For complete instructions, refer to your Access documentation. If you do not set up your parameter query correctly, Crystal Reports will not be able to use it.

1. In Crystal Reports, choose the OPTIONS command from the File menu. The File Options dialog box appears. Click the SQL Tab to activate it.

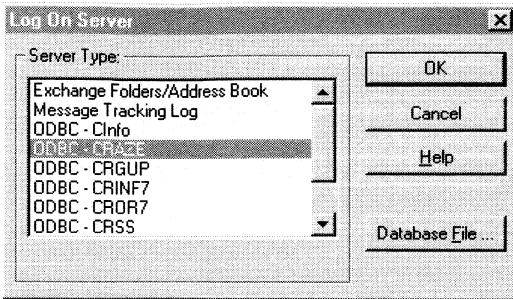


2. Make sure either the Stored Procedures or the Prompt on every table check box is selected.

- The *Stored Procedures* option automatically displays any available stored procedures when you log on to a ODBC data source. Crystal Reports treats Access parameter

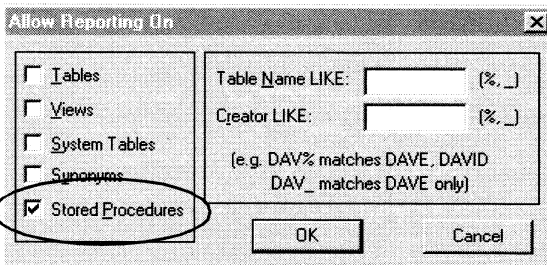
queries much like it treats SQL stored procedures. So, to use a parameter query, the *Stored Procedures* option must be checked.

- The *Prompt on every table* option displays the Allow Reporting On dialog box every time you log on to an ODBC data source. This dialog box provides the same options as the Allow Reporting On section of the File Options dialog box, but provides those options for every ODBC data source you log on to.
3. In addition, you can specify *Table name LIKE* and *Owner LIKE* options if you wish.
 - *Table name LIKE* is based on the SQL LIKE clause. This option allows you to specify the kinds of table names you want to appear in the Choose SQL Table dialog box. Use the underscore character (_) or the percent sign character (%) as wildcards with this function. The underscore character specifies any single character, while the percent sign signifies any character string. For example, DAV_ matches DAVE only, while DAV% matches DAVE and DAVID. *Table name LIKE C%* would display only those tables that have a table name beginning with the letter C.
 - *Owner LIKE* is also based on the SQL LIKE clause. The *Owner LIKE* option allows you to select the Owner (or Creator or Alias) of the table, not the table name itself. For example *Owner LIKE C%* would display only those tables that had an owner beginning with the letter C.
 4. Click *OK* to exit the File Options dialog box.
 5. Create a new report. When you choose SQL/ODBC as the source of your data, the Log On Server dialog box appears.



6. Choose the ODBC data source for your Access database, then click OK.

7. If you did not specify a particular Access database file with your Access ODBC data source, the Select Database dialog box will appear. Locate and select the database containing the Access parameter query you want to use, and click OK.
8. If you selected the *Prompt on every table* check box in the File Options dialog box, the Allow Reporting On dialog box now appears. Otherwise, skip to Step 11.

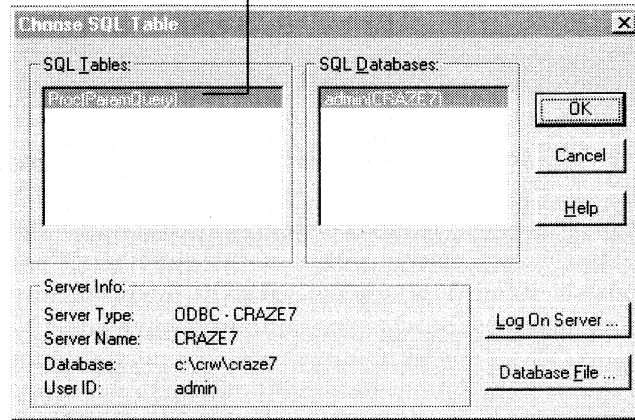


8. Make sure the Stored Procedure check box is selected and click OK when finished

9. The Choose SQL Table dialog box appears. Access parameter queries appear in the SQL Tables list box as:

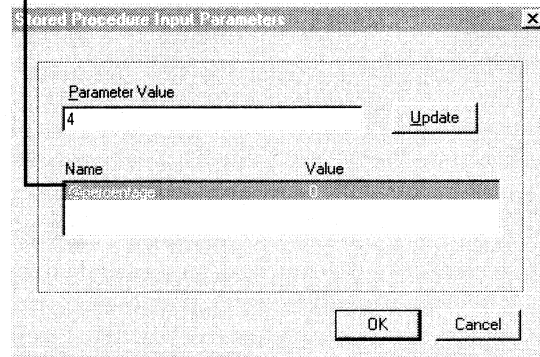
Proc (QueryName)

10. Highlight your query in this list box, then click OK.



The Stored Procedure Input Parameters dialog box appears.

11. Highlight a parameter from this list box.



12. Assign a value by typing into the Parameter Value text box and click the Update button.

13. The value displayed in the *Name* list box will be updated.
14. Repeat steps 12 and 13 for each parameter in your Access parameter query.
15. Click *OK* when finished. You can change parameter values at any time by choosing the **STORED PROCEDURE PARAMETERS** command on the Database menu.
16. Create your report using the fields in the parameter query. Only the records that satisfy the parameter values you

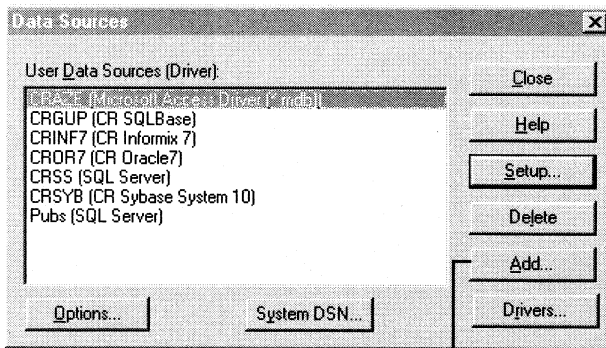
specified in the Stored Procedure Input Parameters dialog box are used in your report.

NOTE: You can not use Access action queries or update queries in Crystal Reports. You can use Access select queries and Cross-Tab queries.

How to set up an ODBC data source

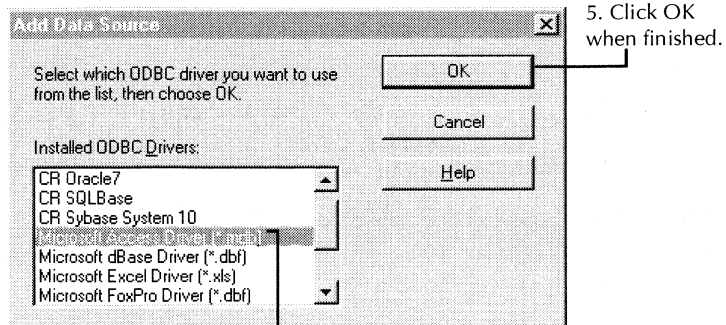
To set up an ODBC data source, you must have an ODBC driver installed for the type of data you want to use. Many DBMS applications automatically install and set up ODBC drivers. If you are not sure whether ODBC drivers have been installed for your data, refer to the documentation that came with your DBMS application.

1. From Program Manager, double-click the ODBC Administrator icon in your Crystal Reports 5.0 program group. The Data Sources dialog box appears.



2. Click the Add button to add a new ODBC data source.

3. The Add Data Source dialog box appears.



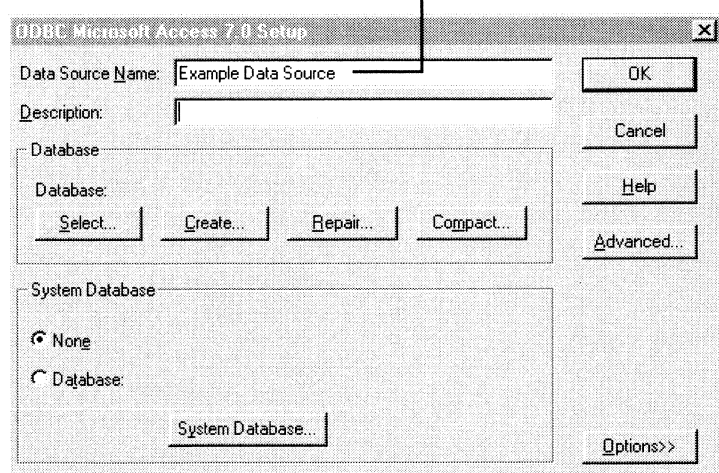
4. Choose the ODBC driver appropriate for your data type from this list.

If a driver does not appear for your data type, an ODBC driver has not been correctly installed. Refer to the documentation for your DBMS application.

6. When you click *OK*. An ODBC Data Source Setup dialog box appears that is specific to the ODBC driver you selected.

NOTE: If an error message appears instead of the Setup dialog box, you do not have the correct ODBC drivers installed on your system for the type of data you selected.

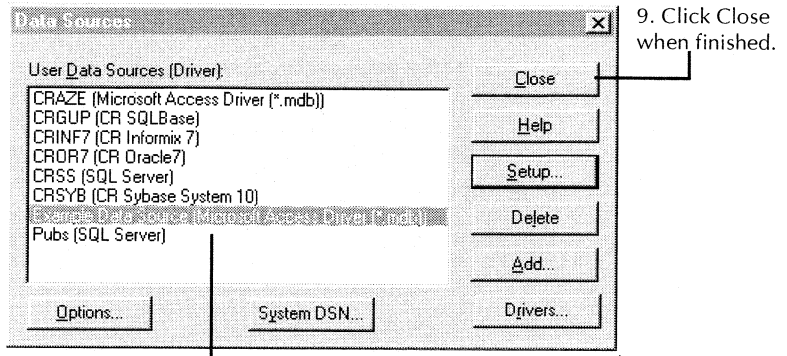
7. Type the name for your new ODBC data source here.



This is the name that you select when you log on to the data source from Crystal Reports.

NOTE: The dialog box that appears may look different than the one shown here, depending on the type of data you are using. This dialog box is specific to the Access 7.0 ODBC driver. For complete information on using the dialog box that appears for your data, click the *Help* button.

8. Click *OK* in the ODBC Data Source Setup dialog box when you are finished setting up your data source. The new ODBC data source will appear in the Data Sources dialog box.

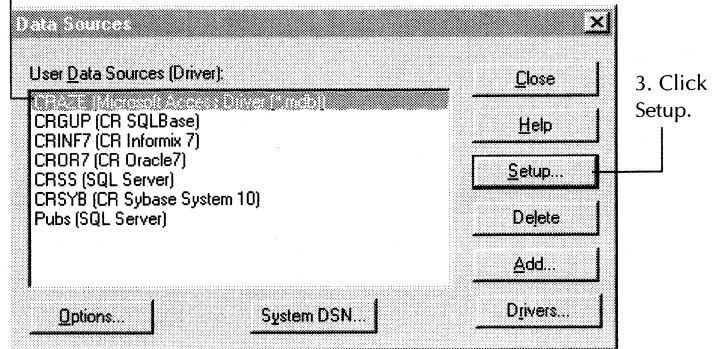


Your new data source will now appear in this list.

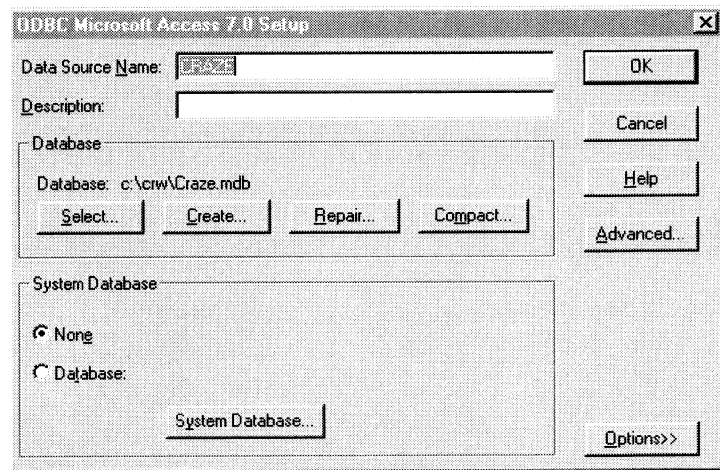
How to check settings for an ODBC data source

1. From the Crystal Reports program group, or the Crystal Reports folder in Windows 95, run the ODBC Administrator application. The Data Sources dialog box appears.

2. Highlight the appropriate data source from this list.



The ODBC Data Source Setup dialog box appears.



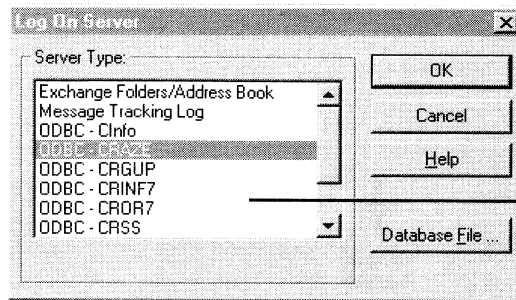
4. The ODBC Data Source Setup dialog box is specific to the data source you selected, and contains controls and information for setting up your data source. Check the settings in this dialog box to make sure the information matches your system and database.

NOTE: The dialog box that appears may look different than the one shown here, depending on the type of data you are using. This dialog box is specific to the Access 7.0 ODBC driver. For complete information on using the dialog box that appears for your data, click the *Help* button.

5. Make any changes that are necessary, and click *OK*.
6. Close the ODBC Administrator Data Sources dialog box.

How to log on to an ODBC data source

1. Choose the **LOG ON SERVER** command from the Database menu. The Log On Server dialog box appears.

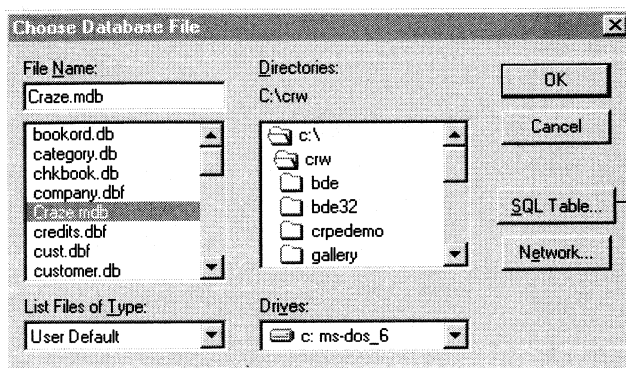


2. Choose the ODBC data source from this list that is appropriate for the database file you want to open and click *OK* when finished.

3. If the data source requires a user name and password, or any other log in information, a login dialog box appears. Type in the information you usually use to access this database, and click *OK*.
4. If you did not specify a database with the ODBC data source, the Select Database dialog box appears. Use the *Drives*, *Directories*, and *Database Name* controls to select the database file, and click *OK*.
5. The Choose SQL Table dialog box appears. Choose a database table from the *SQL Tables* list and click *OK* to add the table to your report, or

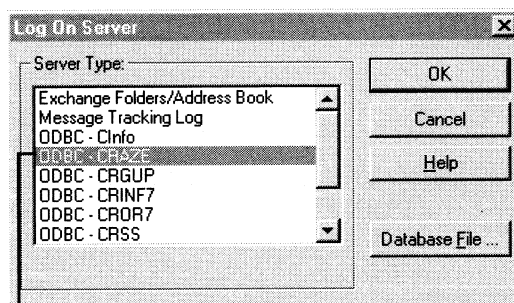
How to add an ODBC database table to a report

1. Choose the **ADD DATABASE TO REPORT** command from the Database menu. The Choose Database File dialog box appears.



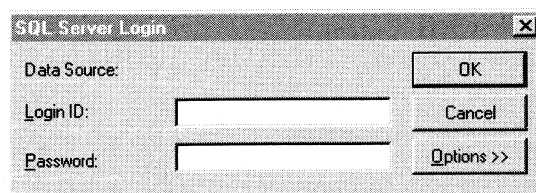
2. Click the SQL Table button.

3. The Log On Server dialog box appears.



4. Choose the ODBC data source from this list that is appropriate for the database file you want to open, then click OK when finished.

5. If the data source requires a user name and password, or any other log in information, the SQL Server Login dialog box appears.



6. Enter the required information then click OK when finished to log in

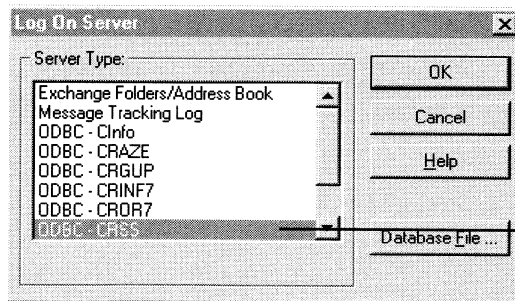
7. If the data source you selected includes a database file specification, or if you specified a database table in Step 4, skip to Step 7.

8. The Select Database dialog box appears. Use the *Drives*, *Directories*, and *Database Name* controls to select the database file, and click *OK*.
9. The Choose SQL Table dialog box appears.
 - Choose a database table from the *SQL Tables* list and click *OK* to add the table to your report, or
 - Click the *Log On Server* button to log on to another ODBC data source.

How to log on to MS SQL Server via ODBC

NOTE: *This section is intended as an example of how to log on to a Microsoft SQL Server data source from Crystal Reports. Your SQL server application or other password protected data source may require different steps. This is intended only as an example of one type of SQL database.*

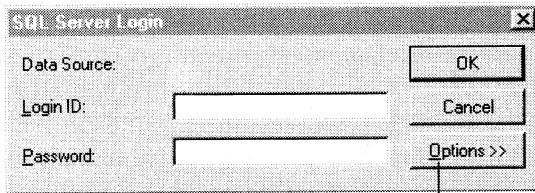
1. Verify the settings for the Microsoft SQL Server data source using ODBC Administrator. See *How to check settings for an ODBC data source*, Page 455.
2. From Crystal Reports, choose the LOG ON SERVER command from the Database menu. The Log On Server dialog box appears.



3. Choose the ODBC - CRSS data source and click OK when finished.

The ODBC - CRSS data source is automatically created during the Crystal Reports setup procedure. This data source allows you to open MS SQL Server databases.

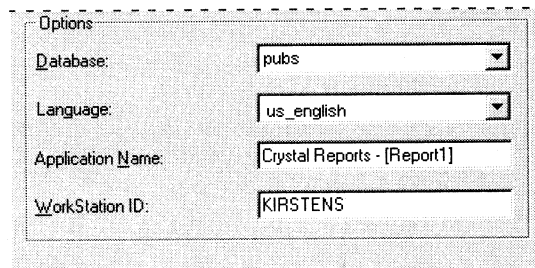
4. The SQL Server Login dialog box appears.



5. Type your MS SQL Server login ID and password.

6. Then, click the Options button.

The Options section of the SQL Server Login dialog box appears.

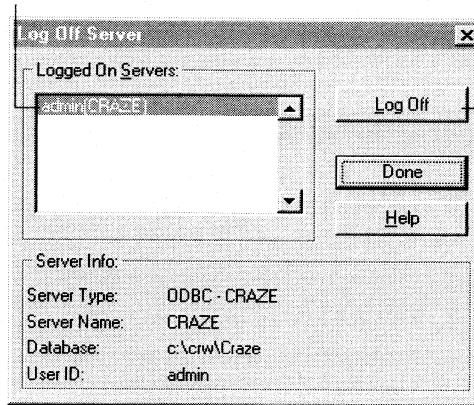


7. Verify the settings for your SQL Server.

8. Verify the name of the SQL Server database you want to open, the database language you want to use, the name of the application you are using (Crystal Reports), and the name of your computer workstation. Make any changes that are necessary.
9. Click *OK*, and the Choose SQL Table dialog box appears. Choose a SQL Server database table from the *SQL Tables* list and click *OK* to add the table to your report.

How to log off an ODBC data source

1. Choose the LOG OFF SERVER command from the Database menu. The Log Off Server dialog box appears.
2. Highlight the ODBC data source you want to log off from this list.



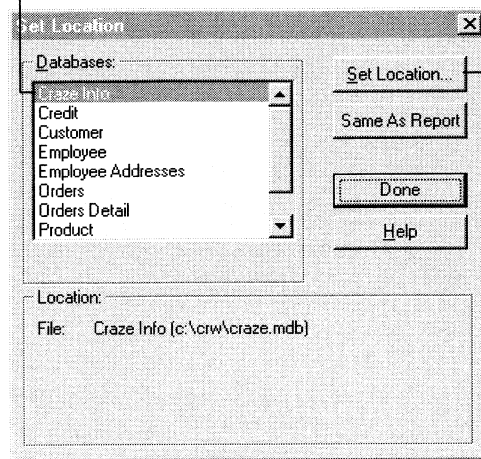
3. Then click the Log Off button.

4. The ODBC data source is removed from the *Logged On Servers* list.
5. Click *Done* when finished.

How to change the ODBC data source accessed by a report

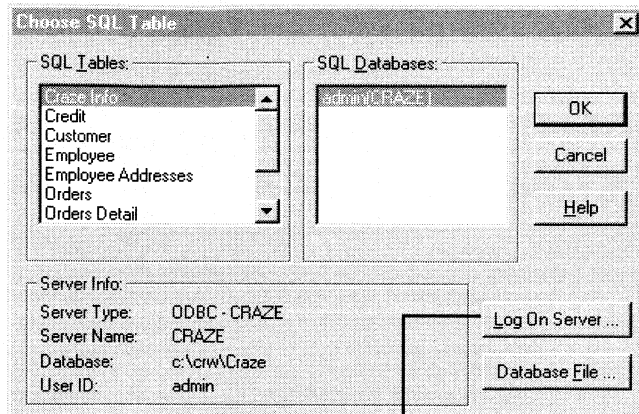
1. Choose the SET LOCATION command from the Database menu. The Set Location dialog box appears with a list of table aliases for the tables in the report displayed. Table location and log on information for the table you highlight will also be displayed below the list of tables. Select the first table in the list and click the *Set Location* button.

2. Highlight the first table in this list.



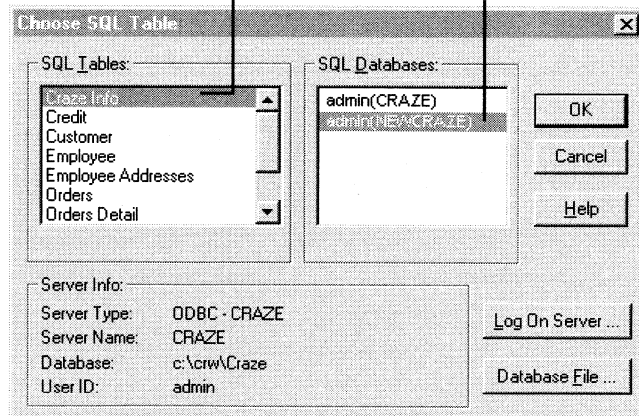
3. Click the Set Location button.

4. Log on to the original ODBC data source if you are not logged on already. You must first log on to the old data source before you can change a report to use a new data source. Since you must be logged on to the old data source, the old data source must be available on the local machine.
 - If the machine is no longer connected to the old server, you can install an ODBC data source with the name of the old data source and have it point to the new database server. Log on to this data source as the "old" data source.
 - If you are using the same data source name to connect to a new server, you must configure that data source under ODBC Admin or the ODBC Control Panel, and make it point to the new server.
5. After logging on to the old data source, you will see the Choose SQL Table dialog box with a list of tables in the *SQL Tables* list box and the old data source in the *SQL Databases* list box.



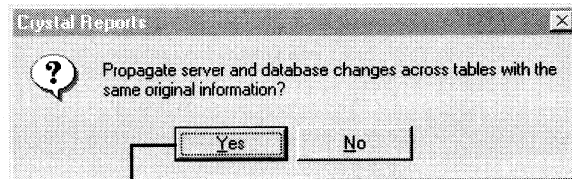
6. Click the Log On Server button and log on to the new data source.

7. You should now be logged on to both the old and the new data sources.
8. The new data source should now appear in the *SQL Databases* list box of the Choose SQL Table dialog box.
9. Click the new data source to display its tables, then click the appropriate table from this list.



The table you choose should correspond to the first table you selected in the Set Location dialog box.

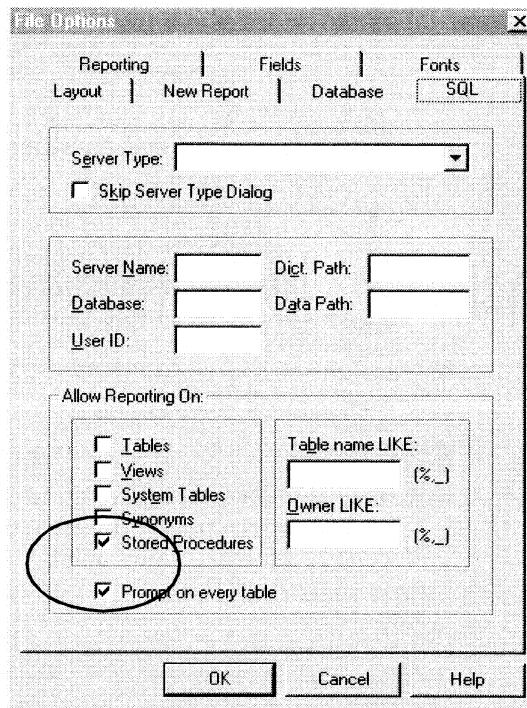
10. Click *OK* and you are prompted to change the location of all tables in the report to the location you specified for the first table.



11. Click *Yes* to set the location of all tables in the report to the same database.
12. Close the *Set Location* dialog box, and choose the *VERIFY DATABASE* command from the *Database* menu to refresh the table definitions in the report.
13. Log off the old data source.
14. Save the report.

How to select a stored procedure from an SQL database and change stored procedure parameters

1. Choose the OPTIONS command from the File menu. The File Options dialog box appears. Click the SQL Tab to activate it.



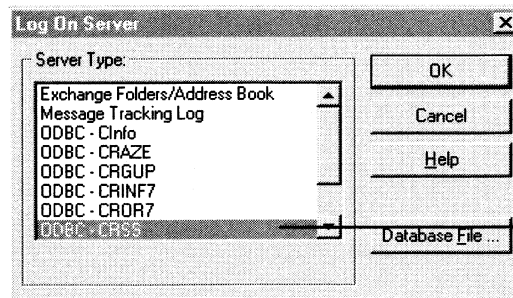
2. Make sure either the Stored Procedures or the Prompt on every table check box is selected.

- The Stored Procedures option automatically displays any available stored procedures when you log on to an SQL database.
- The Prompt on every table option displays the Allow Reporting On dialog box every time you log on to an ODBC data source. This dialog box provides the same options as the Allow Reporting On section of the File Options dialog box, but provides those options for every ODBC data source you log on to.

3. In addition, you can specify *Table name LIKE* and *Owner LIKE* options if you wish.
 - *Table name LIKE* is based on the SQL LIKE clause. This option allows you to specify the kinds of table names you want to appear in the Choose SQL Table dialog box. Use the underscore character (_) or the percent sign character (%) as wildcards with this function. The underscore character specifies any single character, while the percent sign signifies any character string. For example, DAV_ matches DAVE only, while DAV% matches DAVE and DAVID. *Table name LIKE C%* would display only those tables that have a table name beginning with the letter C.
 - *Owner LIKE* is also based on the SQL LIKE clause. The *Owner LIKE* option allows you to select the Owner (or Creator or Alias) of the table, not the table name itself. For example *Owner LIKE C%* would display only those tables that had an owner beginning with the letter C.

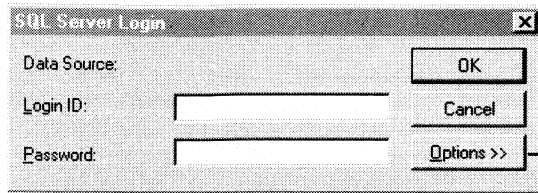
NOTE: For more information on the SQL LIKE clause, refer to your SQL documentation.

4. Click OK to exit the File Options dialog box.
5. Create a new report based on the SQL database that contains the stored procedure you want to use. When you choose *SQL/ODBC* as the source of your data, the Log On Server dialog box appears.



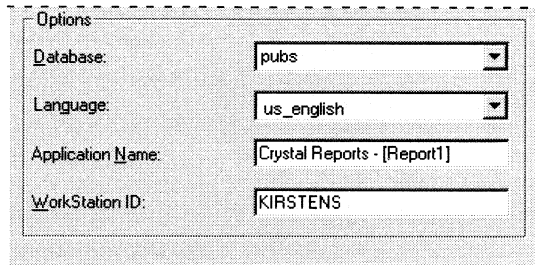
6. Highlight the ODBC data source for your SQL Server from this list, then click OK.

7. The SQL Server Login dialog box appears.



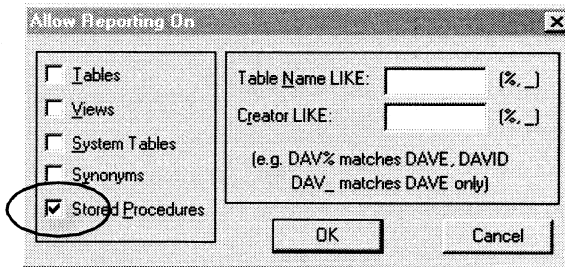
8. Enter required information then click OK when finished to log in
9. Click the Options button.

The Options section of the SQL Server Login dialog box appears.



10. Verify the settings for your SQL Server.

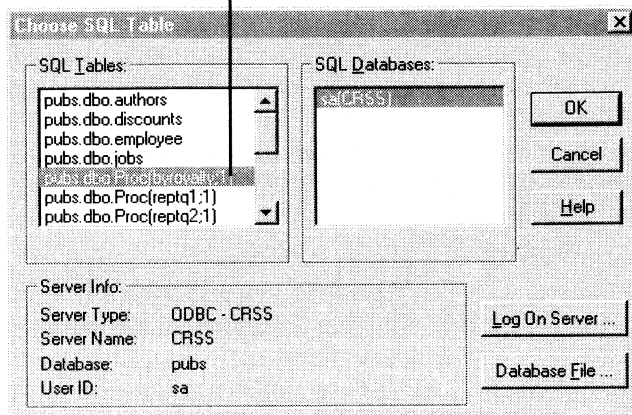
11. Click *OK*. If you selected the *Prompt on every table* check box in the File Options dialog box, the Allow Reporting On dialog box now appears. Otherwise, skip to Step 12.



12. Make sure the Stored Procedures check box is selected, then click *OK*.

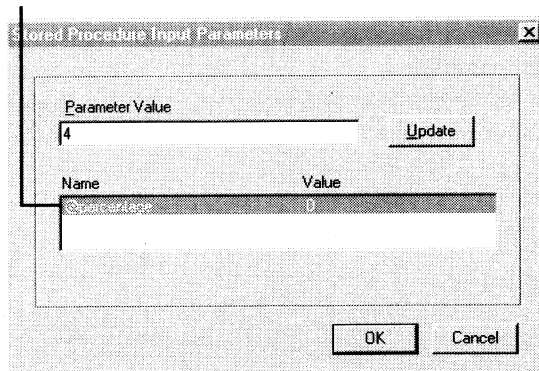
The Choose SQL Table dialog box appears.

13. Highlight a SQL stored procedure from this list, then click OK.



14. The stored procedure is added to your report, and the Stored Procedure Input Parameters dialog box appears.

15. Highlight a parameter from this list box.



16. Assign a value by typing into the Parameter Value text box and click the Update button.

17. The value displayed in the *Name* list box will be updated.
18. Repeat steps 16 and 17 for each parameter that appears in the *Name/Value* list box.
19. Click *OK* when finished. You can change parameter values at any time by choosing the **STORED PROCEDURE PARAMETERS** command on the Database menu.

How to set up an A to B, A to C link

NOTE: This tutorial demonstrates how to set up an A to B, A to C report using the Customer, Credit, and Orders tables in the CRAZE.MDB sample database. If you installed the Crystal Reports sample files (installed by default during a full installation), the CRAZE.MDB database will be installed in the \CRW directory. Use the instructions here as a guideline for creating A to B, A to C reports with your own database files.



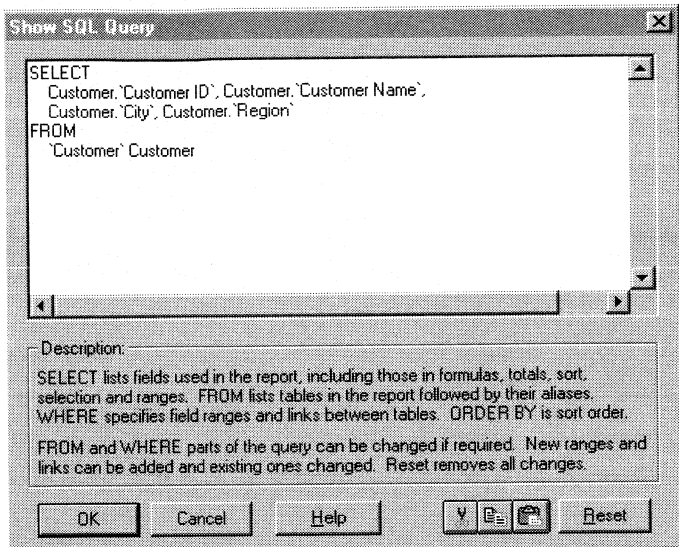
1. Click the NEW button on the Crystal Reports toolbar. The Report Gallery appears.
2. Click Custom, and the dialog box is expanded.
3. Select the type of custom report you wish to create, and click Data File. The Choose Database File dialog box appears.
4. Locate and select the CRAZE.MDB database file from the \CRW directory, and click OK.
5. The Crystal Reports Design Tab appears with the Insert Fields dialog box. All tables from the CRAZE database appear on the Database Tab of the Insert Fields dialog box.
6. We only need three tables from the CRAZE database, so we will remove the unneeded tables. Choose the REMOVE FROM REPORT command on the Database menu. The Remove from Report dialog box appears.
7. In the *Databases* list box, select each unneeded database table, and click Remove to remove each table from your report. Be sure to leave the following tables:
 - Customer
 - Credit
 - OrdersThese are the tables we will use to create an A to B, A to C link.
8. Click *Done* when you are finished removing tables. Only the three tables mentioned in Step 7 should now appear on the Database Tab of the Insert Fields dialog box, along with the fields for those tables.

9. Choose the **VISUAL LINKING EXPERT** command from the Database menu. The Visual Linking Expert appears displaying the three tables.
10. If link lines do not already appear between the tables, click the *Smart Linking* button, and Crystal Reports will link the tables according to the best field matches it can find. Link lines should appear linking the Customer table to the Credit table by the Customer Credit ID field and linking the Customer table to the Orders table by the Customer ID field.
11. Click one of the link lines in the Visual Linking Expert, and click *Options*. The Link Options dialog box appears.
12. Click the *Look up all of one, then all of others* option button in the *When linking to two files from this file* section of the Link Options dialog box. This option establishes an A to B, A to C link.
13. Click *OK* to return to the Visual Linking Expert. The option you selected affected all links from the Customer table. To make sure, click the link line that you did not select in Step 11.
14. Click *Options*, and verify the *Look up all of one, then all of others* setting in the Link Options dialog box.
15. Click *OK*, and click *OK* in the Visual Linking Expert. You have now established an A to B, A to C relationship among the three tables. See *Look up all of one, then all of others (A to B, A to C)*, Page 424.

How to edit an SQL query

NOTE: This section is only valid for reports using ODBC data sources. An SQL query is automatically generated by Crystal Reports when you design a report based on one or more ODBC data sources, This query is sent to ODBC as an instruction to gather data needed by the report.

1. Choose the **SHOW SQL QUERY** command from the Database menu. The Show SQL Query dialog box appears.



2. Click anywhere inside the *SQL Query* edit box to begin making changes.
3. Use the ANSI SQL language to fine tune the SQL query.

NOTE: You can not change the *SELECT* clause of the *SQL statement*.

4. Cut, copy or paste any part of the query to or from the Clipboard if needed. See *The SQL language*, Page 441.

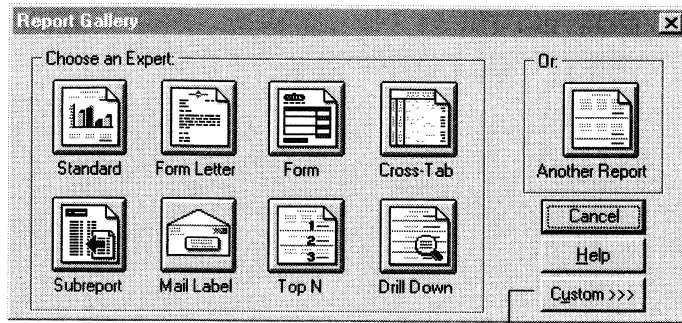
How to use an ACT! database

ACT! is a powerful Contact Management application that stores all of your contact information in a database format similar to xBASE databases (dBASE, Clipper, and FoxPro). See *ACT!*, Page 492.

This section demonstrates how to create a custom report using ACT! data.

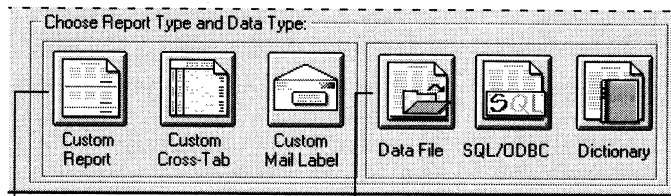


1. Click the **NEW** button on the toolbar in Crystal Reports. The Report Gallery appears.



2. Click the Custom button.

The Report Gallery will expand.



3. Click a Custom Report button.

4. Click the Data File button.

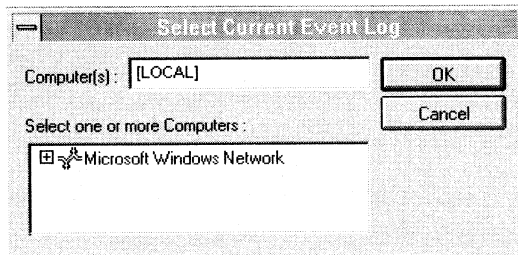
5. The Choose Database File dialog box appears.
6. Use the controls in the Choose Database File dialog box to locate and select the file CRW.ACT. This file should be located in the same directory as Crystal Reports (\CRW by default). Click OK when finished.
7. The Choose File for ACT! dialog box appears.
8. Use the controls in this dialog box to locate and select your ACT! database.
9. Click OK when finished, and a new Design Tab appears in the Crystal Reports window. Create your report using fields from your ACT! database.

How to open the NT Event Log

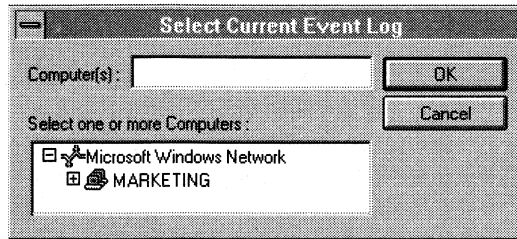
If you are using a Windows NT system, the 32-bit version of Crystal Reports gives you the ability to generate reports based on the NT Event Log. For your convenience, the program comes with a pre-designed report that you can run using your own Event Log as a data source. The report, EVENTDTL.RPT, is located in the \REPORTS\BO\NTEVENT directory on the Crystal Reports CD.

NOTE: This report is only available with the Professional version of Crystal Reports 5.0 and is not available on floppy disks.

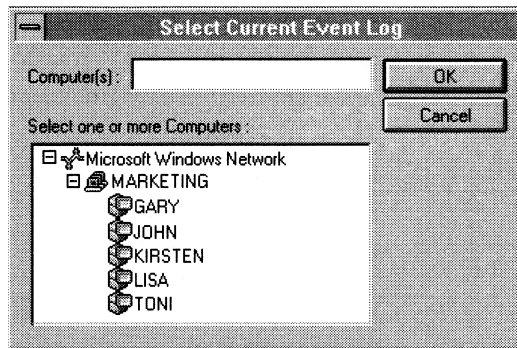
1. Open the EVENTDTL.RPT report file in Crystal Reports.
2. Click the REFRESH button on the toolbar. A dialog box will appear asking if you want to refresh data.
 - If you click *No*, the original report will appear.
 - If you click *Yes*, the Select Current Event Log dialog box appears:



3. The computer referenced in the *Computer(s)* box is the computer the sample report is based on. To change the computer, click on the plus (+) sign to the left of Microsoft Windows Network. The directory structure will expand to display the network group that your computer is connected to. The Select Current Event Log dialog box will now look similar to the following:



4. You can now click on the plus (+) sign to the left of the network group to display individual machines connected to the group. The Select Current Event Log dialog box will now look similar to the following:



5. From this point, you simply need to select the NT machine for which you would like an NT Event Log Report. When you select a machine by clicking on it, that name will appear in the *Computer(s)* text box.

NOTE: If you select a non-NT machine, the following error message appears: *The RPC server is unavailable. You can run this report only from NT machines.*

6. Click OK. A new NT Event Log Report will appear. This report will contain information for the machine that you selected via the Select Current Event Log dialog box.
7. You can now print the report if you wish or review it in the Preview Tab. When you are ready to save the report, save the report to a new file name. When you want to update the report, simply refresh the report data. When you want to run a report on a different NT event log, open the original report (EVENTDTL.RPT) and repeat steps 2-7.

20

Data Sources

What you will find in this chapter...

Introduction 476

Four types of data 477

Direct access database files 478

ODBC data sources 498

Crystal Query data sets 514

Crystal Dictionaries 516

Introduction

Crystal Reports can access data stored in almost any common database format, as well as many uncommon formats. In addition, Crystal Reports exploits the full benefits of Crystal Query (.QRY) files and Crystal Dictionaries (.DC5). See Chapter 17, *Queries*, Page 359, and Chapter 18, *Dictionaries*, Page 383.

This chapter discusses the many different types of data that Crystal Reports can access and explains the data access layers involved in connecting to the data. If you are not sure what Database Management System (DBMS) your company uses, contact your IS manager or network administrator.

Why you should read this chapter

The principal purpose of Crystal Reports is simply to access data stored in databases and produce reports on that data. This goal is one of the oldest uses of computers and remains one of the most common and necessary. Crystal Reports is designed to make that task easier, less time consuming, and more powerful.

This idea of accessing data remains at the root of every report produced using Crystal Reports. By understanding how the application accesses data, you will gain a better knowledge of the reporting process as well as a better knowledge of the type of data that Crystal Reports can work with.

In addition, understanding the data access process will help you troubleshoot problems you may encounter while trying to open a particular database file from Crystal Reports. This is especially useful for IS managers and anyone providing data access support for a group of users.

Most of the information in this chapter is designed for experienced Crystal Reports users and IS managers and covers technical aspects of Database Management Systems (DBMS) and data storage techniques. A familiarity with computers, the Windows, Windows 95, or Windows NT operating system, and at least one DBMS application is assumed.

A note to developers

This chapter concentrates on the principles of data access from Crystal Reports. However, most of the same concepts can be applied to any application accessing data through the Crystal

Report Engine (CRPE.DLL), or any of the Crystal Custom Controls (CRYSTAL.VBX, CRYSTLxx.OCX, UCRPExx.DCU (VCL)) which access data through the Report Engine. For that reason, Crystal Reports, as used in this chapter, refers both to the Crystal Reports application and the Crystal Report Engine unless otherwise specified. See Chapter 29, *Application Development with Crystal Reports*, Page 631.

NOTE: Most of the file names mentioned in this chapter are for the 16-bit version of Crystal Reports unless otherwise specified. File names for 32-bit Crystal Reports are similar, but will have some aspect indicating 32-bit. For example, PDSODBC.DLL is the 16-bit ODBC translation file, while P2SODBC.DLL is the 32-bit ODBC translation file.

Four types of data

The types of data that Crystal Reports can access fall into four general categories:

- Direct access database files
- ODBC data sources
- Crystal Query data sets
- Crystal Dictionaries

Each type of data must be accessed by Crystal Reports using a specific set of DLLs and other data access related files. Once you understand the process Crystal Reports uses to access each type of data, you will have a better understanding of the report creation process and the elements used to turn your data into powerful reports. See Chapter 17, *Queries*, Page 359, and Chapter 18, *Dictionaries*, Page 383.

NOTE: When accessing any type of data, Crystal Reports relies on the database drivers to provide field names, field types, and field lengths. This information is provided by either the database engine or the ODBC database driver.

Direct access database files

Crystal Reports can access many of the most common PC database formats directly. In other words, Crystal Reports has built-in capabilities to directly open database files and tables designed in dBASE, FoxPro, Clipper, Btrieve, Paradox, and Microsoft Access, among others. This functionality exists as soon as you install Crystal Reports. Once it is installed on your system, you can immediately begin creating reports based on these databases simply by selecting the appropriate file.

Advantages

Accessing the database directly is the fastest route to reading the data. Crystal Reports only needs to talk to a single data access layer that provides contact with the data. Report results can be obtained quickly in almost any system environment.

In addition, data access is simple. Direct access database files are point-and-click data sources. You need only select the required database files and Crystal Reports reads all of the stored data.

Disadvantages

When you access a database directly from Crystal Reports, only that database type can be used by the report. You can not switch to a different type of database or table without creating a new report.

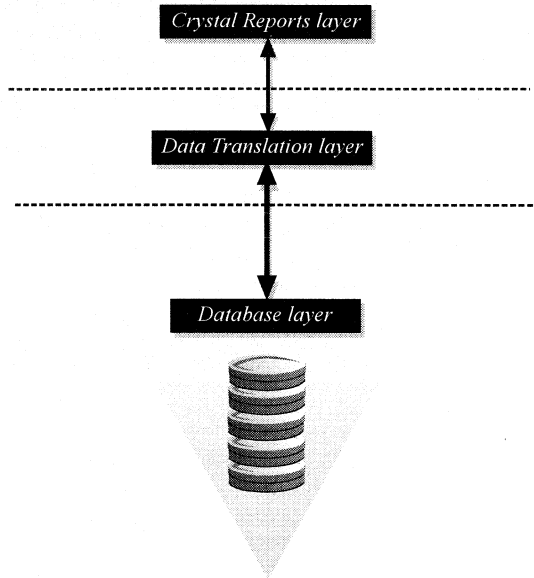
For example, if you design a report based on Btrieve data, you can not change the tables accessed by the report to Paradox data. Crystal Reports communicates with Btrieve data using a Btrieve specific syntax, a syntax that is not compatible with Paradox data.

If you access data through ODBC, on the other hand, the syntax used is always the SQL language regardless of the actual database type. See *ODBC*, Page 498.

Three layers

Direct access of database files from Crystal Reports requires three layers:

Direct Data Access Layers



Crystal Reports uses the data translation layer to talk to the database and access its data.

Crystal Reports

Crystal Reports operates as an interface through which you format, arrange, select, and sort the data stored in database files. It obtains data by communicating with one or more files in the data translation layer that can actually read the database. Since Crystal Reports can work with many forms of data, it must rely on other files to work directly with the data. Crystal Reports can then use a native method of communication to talk to the translation files.

Data Translation

Data is translated through a set of DLLs specific to Crystal Reports. Crystal Reports uses the DLLs specific to a certain data type to understand how data is organized for that type and to present it correctly when your report is printed, previewed, or exported.

NOTE: Crystal Reports comes with all data translation files for each of the direct access database types that it supports. For complete information on all required files, refer to the Runtime File Requirements Help file.

Database

The database file consists of one or more tables. Different DBMS applications store database information differently. For example, dBASE stores each database table as a separate file. Access, on the other hand, can store several tables, along with queries, macros, and other database elements, all in a single file.

When Crystal Reports accesses a database file directly, it automatically retrieves information about all of the tables and fields in that file. You may not use all of the tables or fields, but Crystal Reports will make them available to you. In other words, when a dBASE file is opened, only the one table in the dBASE file is available. However, when an Access file is opened, every table in that file is available, even if you never use them all.

NOTE: Crystal Reports will also open queries in an Access database through the DAO engine or ODBC and will allow you to report on query fields, just like table fields. See DAO, Page 485, and Access, Page 504.

Common database formats

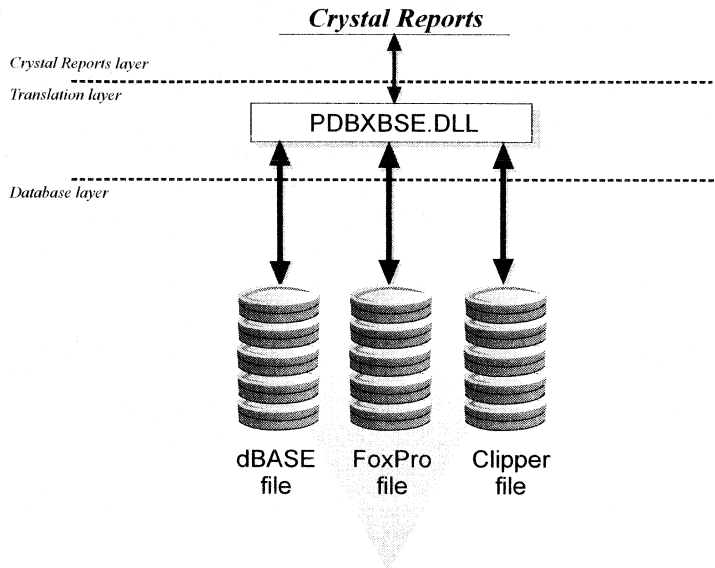
Although Crystal Reports uses the same three-tiered system for obtaining data from all direct access database file formats, each format requires a different set of Dynamic Link Libraries (DLL). However, some formats expand the basic three-tiered structure. The following sections cover the system used by Crystal Reports to access data from some of the most popular database formats.

dBASE, FoxPro, Clipper

The dBASE database format remains one of the most popular database management systems used in business. For that reason, Crystal Reports has been designed to open dBASE data simply and directly through the xBase engine (inside PDBXBSE.DLL). FoxPro and Clipper are dBASE compatible database formats, and Crystal uses the same DLL to access files created by any of these three DBMS applications.

NOTE: The PDBXBSE.DLL translation layer supports FoxPro files up through version 2.5. See Visual FoxPro, Page 513.

dBASE ,FoxPro, and Clipper



The file PDBXBSE.DLL handles all translation between Crystal Reports and the dBASE, FoxPro, or Clipper files. Each database file contains only a single database table, but there is no limit on the number of files that can be accessed by a report.

NOTE: dBASE data can also be accessed through the Borland Database Engine (BDE) using the translation file PDBBDE.DLL. To see how the BDE communicates with database data, see the next section on Paradox data. The BDE, however, does not support FoxPro or Clipper data.

dBASE Query By Example

A dBASE Query By Example (.QBE) file is a query on dBASE data that is stored as a separate data set. The purpose of a QBE is similar to a Crystal Query file (Page 514). Crystal Reports can open and report on the data retrieved by a dBASE QBE via the dBASE for Windows or Visual dBASE QBE engine.

NOTE: For more information on dBASE QBE files, refer to your dBASE for Windows or Visual dBASE documentation.

The layers involved in opening QBE data are similar to the layers used for opening a dBASE database file (Page 480). The primary difference is the file used for data translation. A dBASE database is translated through the PDBXBSE.DLL file, while a dBASE QBE is translated through the PDQQBE.DLL file.

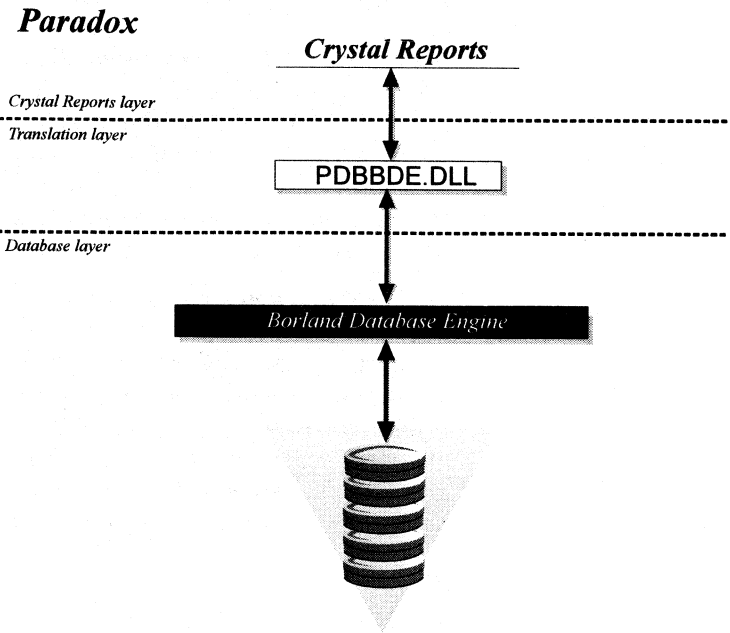
PDQQBE.DLL uses the dBASE for Windows or Visual dBASE database engine to read data, evaluate formulas, and sort data. For this reason, you must have dBASE for Windows or Visual dBASE installed on your system.

Paradox

Files created with Paradox (.DB) are made available to other applications through the Borland Database Engine (BDE). The BDE is made up of several files installed by Crystal Reports in the \IDAPI directory by default.

- ILD01.DLL
- IDR10009.DLL
- IDASCI01.DLL
- IDBAT01.DLL
- IDDBAS01.DLL
- IDODBC01.DLL
- IDAPI01.DLL
- IDQRY01.DLL

The BDE does the actual work with the Paradox data, retrieving the requested tables and fields. Since the BDE works so closely with the actual data, it combines with the Paradox database file to create the database layer in the three layer data access model. Crystal Reports accesses the BDE through the PDBBDE.DLL translation file.



The 32-bit version of the BDE uses different files. Otherwise, Crystal Reports uses 32-bit Paradox files in the same way, accessing them through the P2BBDE.DLL 32-bit translation file.

The files that make up the 32-bit version of the BDE are:

- BLW32.DLL
- CEEUROPE.BLL
- CHARSET.BLL
- EUROPE.BLL
- IDAPI32.DLL
- IDAPIINST.DLL
- IDASCI32.DLL
- IDBAT32.DLL
- IDDBAS32.DLL
- IDDR32.DLL
- IDODBC32.DLL
- IDPDX32.DLL
- IDQBE32.DLL
- IDR20009.DLL
- IDSQL32.DLL
- OTHER.BLL
- USA.BLL
- CW3215.DLL

Microsoft Access

Microsoft Access provides several means for opening its database files. Each method has its advantages and disadvantages, and the technique that you should use can depend on how your data is set up. Below are descriptions of two techniques for opening Access data from Crystal Reports, through the Jet engine and through the DAO engine. The third technique uses Microsoft's Open Database Connectivity (ODBC) standard, and is described on Page 504.

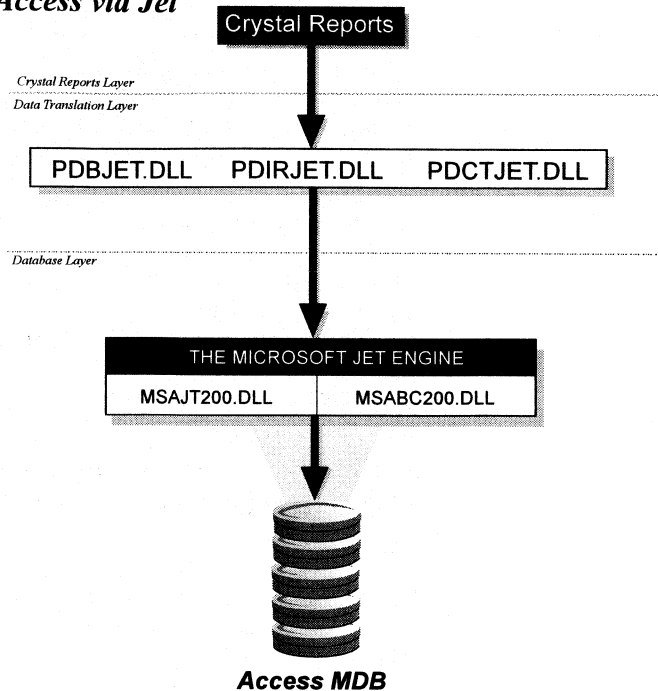
NOTE: When you open an Access database using the Jet or DAO engine, Crystal Reports opens the entire database and loads information about all tables and queries from the database. To do this, Crystal Reports must reserve a large section of your computer's memory, called a buffer, in advance. Computer memory restrictions limit this buffer to 65,536 bytes (64k). On average, this restricts the size of your Access database to about 80 tables (depending on the number of fields in your tables).

Jet

The Microsoft Jet Database Engine is the part of the Microsoft Access Database Management System that actually handles your database data. As a user, you usually do not work directly with the Jet engine. It acts as a gateway through which Access data is made available to applications. For this reason, the Jet engine must be used regardless of the overall method used to access your Access data. You will see the Jet engine in other sections of this chapter that discuss Access data.

Since the Jet engine is so closely tied to Access data, Crystal Reports considers it a part of the actual database. In the diagram, the files for the Jet engine appear in the database layer. Crystal Reports uses three files, PDBJET.DLL, PDIRJET.DLL, and PDCTJET.DLL, to translate information to and from the Jet engine.

Access via Jet



Reading Access data through the Jet engine is the most direct route, and, therefore, the fastest method, for reading the data. However, Jet does not allow you to read Access queries. If you need to open Access queries from Crystal Reports, you should use the DAO engine (discussed next) or ODBC (Page 504).

NOTE: *If you check the Use Indexes check box on the Database Tab of the File Options dialog box, Crystal Reports can pass much of the data retrieval process, including simple selection formulas, down to the Jet engine, improving performance and speed.*

NOTE: *Crystal Reports provides all necessary files for reading Access tables through the Jet engine. For complete information on necessary files, refer to the Runtime File Requirements Help file.*

DAO

Microsoft's new Data Access Object (DAO) Engine for Access 2.0 files and later provides all of the functionality of the Jet engine but

also adds many new data access features. DAO uses Microsoft's Object Linking and Embedding (OLE) technology (installed with Windows 95 and Windows NT) to provide access to Access data through an object-oriented approach.

NOTE: The Microsoft DAO engine ships only with the 32-bit version of Crystal Reports. For that reason, the description of the DAO engine here refers to 32-bit files only. The 16-bit DAO engine is available from the 16-bit version of Microsoft Visual Basic 4.0. If you have these files, you can use the 16-bit DAO engine with 16-bit Crystal Reports.

In addition to Access database tables, Crystal Reports can open and use Access queries through DAO. If you are not familiar with Access queries, refer to your Access documentation. See *How to open Access Queries through DAO*, Page 444.

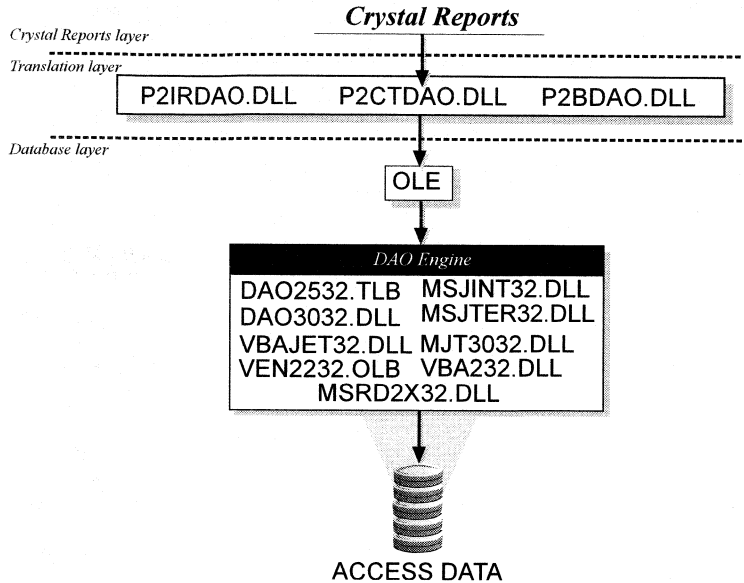
NOTE: Access Parameter queries and Cross-Tab queries can only be opened through an ODBC connection. For information on how Crystal Reports opens Access databases through ODBC, see Page 504. See How to open Access parameter queries, Page 448. Access Action queries are not supported by Crystal Reports.

NOTE: When opening Access queries in a report, make sure the Views and Stored Procedures boxes are checked on the SQL Tab of the File Options dialog box in Crystal Reports. This ensures that the queries will be visible when you open the Access database.

The DAO engine greatly broadens the possibilities available to Access database users. Because DAO expands and builds on the basic functionality of the Jet engine, primarily working with the actual Access database data, DAO is also considered part of the database layer. To translate information and data to and from DAO, Crystal Reports uses the DAO translation files P2BDAO.DLL, P2CTDAO.DLL, and P2IRDAO.DLL. Compare these files to the translation layer for accessing MS Access data directly through the Jet engine (Page 484).

NOTE: The Jet engine is incorporated into the DAO engine and does not appear as a separate engine in the diagram below.

Access via DAO

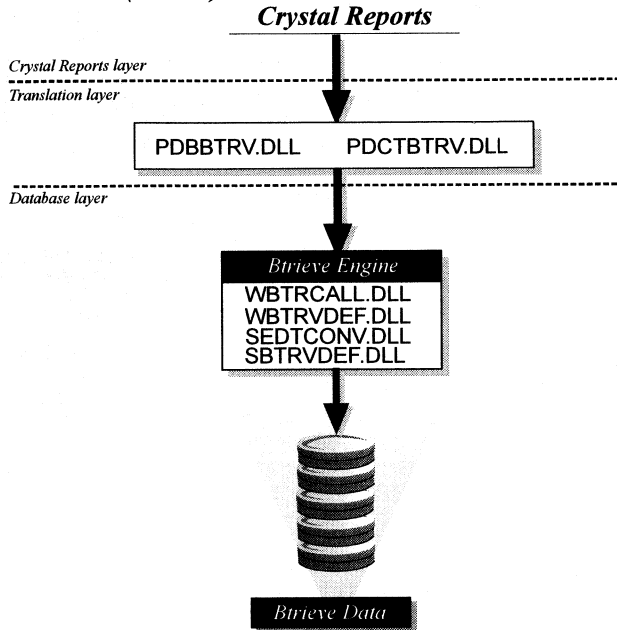


NOTE: *Crystal Reports provides all necessary files for reading Access data through the DAO engine. For complete information on necessary files, refer to the Runtime File Requirements Help file.*

Btrieve

The 16-bit version of Crystal Reports uses two translation files to communicate with the 16-bit Btrieve engine, PDBBTRV.DLL and PDCTBTRV.DLL. These files work with the Btrieve files WBTRVDEF.DLL, WBTRCALL.DLL, SBTRVDEF.DLL, and SEDTCONV.DLL for most data access operations.

Btrieve (16-bit)

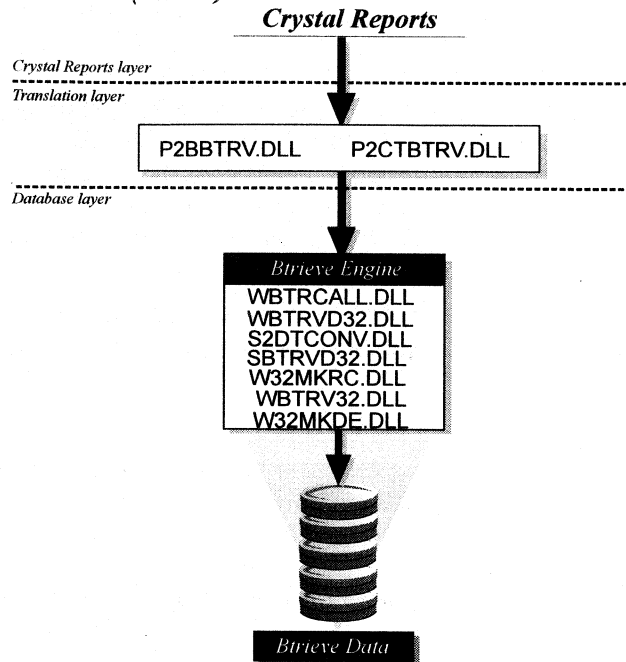


The rest of the Btrieve engine is a complex collection of DLLs and EXEs that are specific to the version of Btrieve you are using. For complete information on different 16-bit Btrieve engine files, search for *Btrieve* in Crystal Reports Help, and refer to your Btrieve documentation.

NOTE: Btrieve must be configured correctly for Crystal Reports to read Btrieve databases. If Btrieve is already configured correctly on your system, Crystal Reports can use your Btrieve data upon installation. Crystal Reports installs the Btrieve files that it requires to read Btrieve data, but you should refer to your Btrieve documentation to make sure the Btrieve engine is configured correctly.

The 32-bit version of Crystal Reports connects to the 32-bit Btrieve engine through a similar set of Btrieve translation files:

Btrieve (32-bit)



The primary difference between 32-bit Btrieve and 16-bit Btrieve is the Btrieve engine itself. For complete information on the Btrieve engine, refer to your Btrieve documentation.

NOTE: When you open a Btrieve database, Crystal Reports opens the entire database and loads information about all tables from the database. To do this, Crystal Reports must reserve a large section of your computer's memory, called a buffer, in advance. Computer memory restrictions limit this buffer to 65,536 bytes (64k). On average, this restricts the size of your Btrieve database to about 80 tables, depending on the number of fields in each table.

Btrieve DDF files

Crystal Reports does not determine the definitions of Btrieve data files directly from the data files themselves. It needs a set of Btrieve Data Dictionary Files (.DDF) that contain file, field and index information. Crystal Reports uses WBTRVDEF.DLL and SBTRVDEF.DLL to parse these DDF files. The following are the required DDFs which must all reside in the same directory:

FILE.DDF

FIELD.DDF

INDEX.DDF

A set of DDFs normally contain definitions for multiple Btrieve data files. Once any of the DDFs is selected when creating a new report, Crystal Reports immediately adds all the data files defined in the DDFs into the report. Crystal Reports also takes the path defined in the DDFs as the default path of the data files. The DDFs and data files can reside in different locations.

NOTE: Be sure to study your Btrieve documentation for more information on Btrieve DDFs and configuring the Btrieve engine.

Exchange

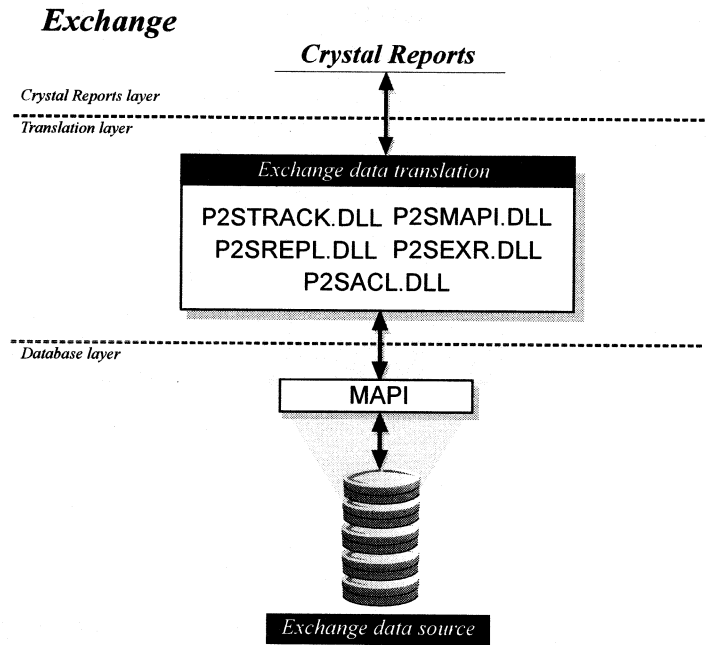
Exchange is Microsoft's successor to MS Mail. Exchange, however, includes not only electronic mail, but also management of group scheduling, electronic forms, groupware, and Internet connectivity. An Exchange folder can contain standard notes (mail), files, and instances of Exchange forms. All of this data is stored in the Exchange Information Store.

Crystal Reports can report on data contained in the Exchange Information Store. Exchange data sources available for reporting include:

- Message Tracking Logs (32-bit only)
- Address Lists
 - Personal Address Books
 - Global Address Lists
 - Distribution Lists
- Exchange Folder Contents
 - mail messages
 - Exchange Form applications
 - properties of OLE documents
- Exchange Administrator (32-bit only)
 - properties of Exchange mailboxes on the Exchange Server

- properties of public folders on the Exchange Server
- replica list of public folders
- ACL (Access Control List) of public folders

Each Exchange data source can be used like a database table and can be linked to other data sources. For example, the Message Tracking Log may be joined to an Address List by using an e-mail address as the link field.



The data translation file used to access the Exchange data source depends on what data source is being accessed. The following table lists each of the Exchange data translation files and describes their purpose:

File name	Data Source
P2STRACK.DLL	Message Tracking Logs (32-bit only)
P2SMAPI.DLL / PDSMAPI.DLL	Exchange Folder Contents and Address Lists

File name	Data Source
P2SEX.R.DLL	Properties of Exchange Mailboxes and Public Folders (32-bit only)
P2SREPL.DLL	Replica Lists of Public Folders (32-bit only)
P2SACL.DLL	Access Control Lists (ACL) of Public Folders (32-bit only)

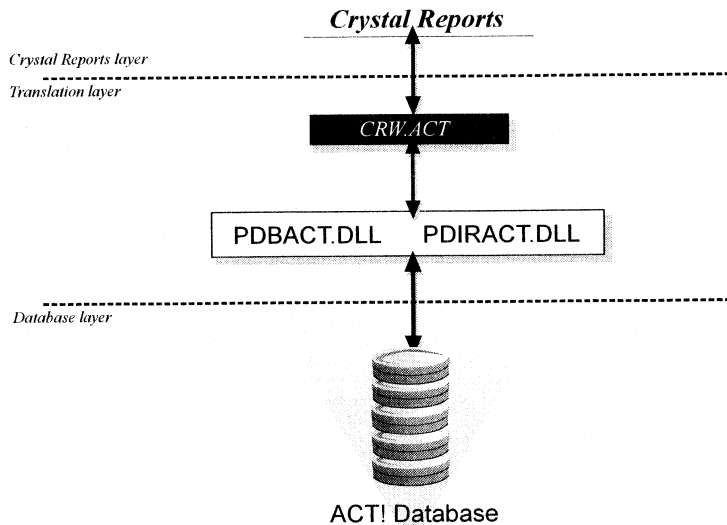
NOTE: PDSMAPI.DLL works with 16-bit Crystal Reports while P2SMAPI.DLL works with 32-bit Crystal Reports. Also, PDSMAPI.DLL is the only Exchange driver available for 16-bit Crystal Reports.

Exchange translation files work directly with the Microsoft Messaging API (MAPI). MAPI acts as a database engine for Exchange data.

ACT!

Symantec's ACT! contact management software stores information in a relational database format similar to the xBase format. See *dBASE, FoxPro, Clipper*, Page 480). Crystal Reports can read this data and let you produce reports based on your contact information.

ACT!



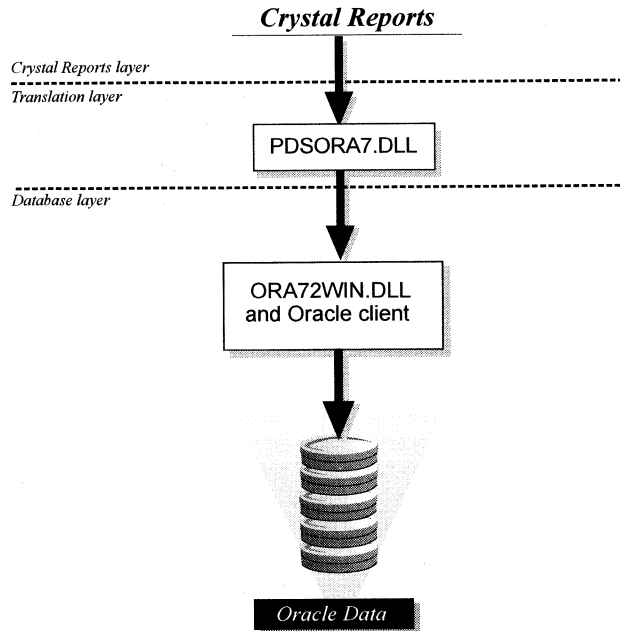
You open ACT! data by selecting the CRW.ACT file from Crystal Reports when you create a new report. CRW.ACT must be installed in the same directory as Crystal Reports. This file forces Crystal Reports to load the PDBACT.DLL and PDCTACT.DLL translation files. This step is important, because ACT! data looks like xBase data to Crystal Reports, so Crystal Reports will use the PDBXBSE.DLL translation file unless instructed otherwise. If this happens, the data will be translated as xBase data rather than ACT! data, and may not appear correctly in your report. See *How to use an ACT! database*, Page 471.

Oracle 7

Crystal Reports 5.0 now supports direct access of Oracle 7 SQL databases. This accessibility is provided by the PDSORA7.DLL translation file (installed with Crystal Reports). This file can communicate directly with the Oracle 7 database driver ORA72WIN.DLL, which works directly with Oracle databases and clients, retrieving the data you need for your report.

NOTE: The Oracle client software must be installed on your system, and the location of the ORA72WIN.DLL file must be in the PATH statement of your AUTOEXEC.BAT file.

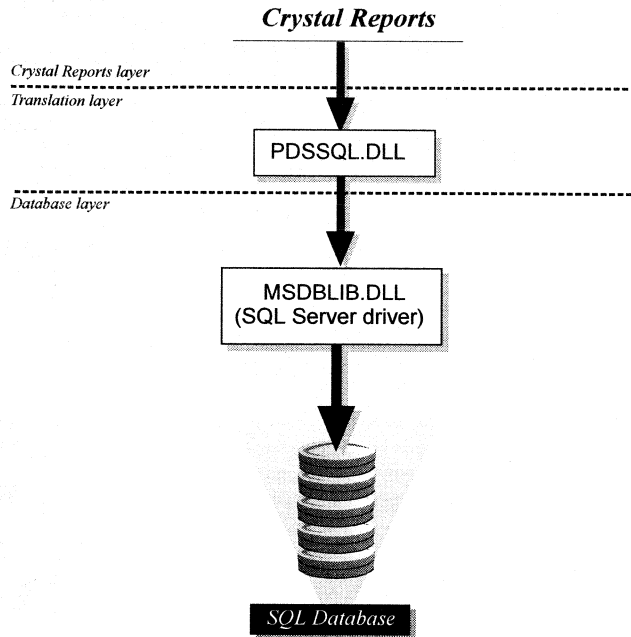
Oracle 7



Microsoft SQL Server 6.x

Databases created by Microsoft's SQL Server, versions 6.0 and 6.5, can be read directly from Crystal Reports. The PDSSQL.DLL file installed with Crystal Reports translates requests to the SQL server and returns data from the SQL server. It communicates directly with the Microsoft SQL Server driver MSDBLIB.DLL.

Microsoft SQL Server 6.x



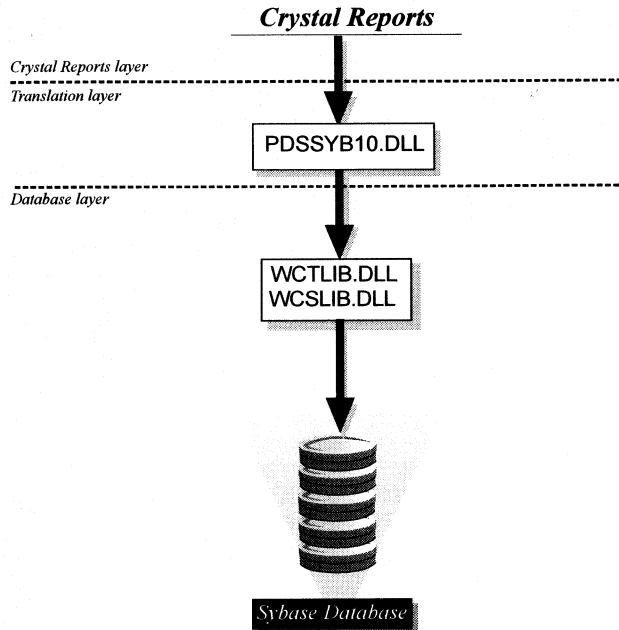
NOTE: Network administrators and IS managers who run Microsoft's Systems Management Server for BackOffice can run reports off of Systems Management Server data. Systems Management Server uses SQL Server to store system data, so the files used to access that data are the same files shown here for SQL Server.

Sybase System 10/11

Crystal Reports 5.0 opens SQL data created by Sybase System 10 or System 11 directly through the PDSSYB10.DLL, installed with Crystal Reports. This translation file works with the Sybase database drivers WCTLIB.DLL and WCSLIB.DLL to read Sybase System 10/11 data. If your Sybase server is correctly configured, you will be able to read Sybase data as soon as Crystal Reports is installed.

NOTE: The 32-bit Sybase drivers are LIBCT.DLL and LIBCS.DLL.

Sybase System 10/11

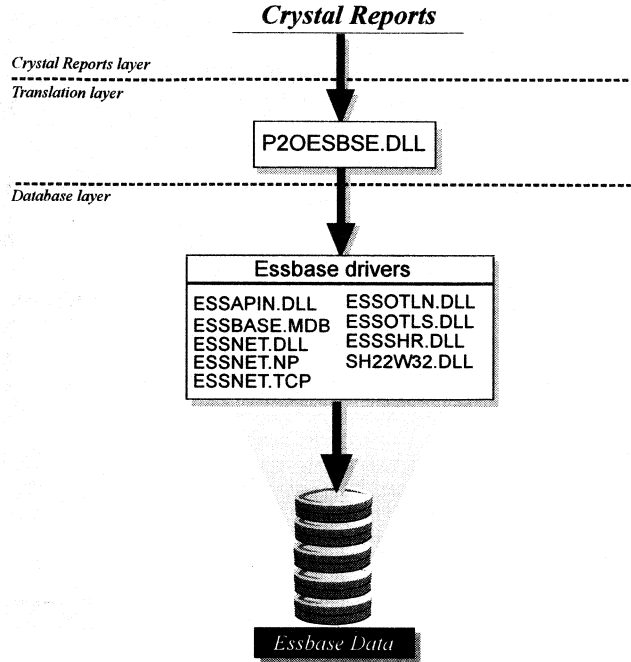


Arbor Essbase

Essbase is a highly powerful database format designed on the OLAP (On-Line Analytical Processing) model. OLAP presents data in dimensions, rather than tables. Users can look at relationships between data on an as-needed basis, drilling-down on dimensions to find exactly the data they need in a matter of seconds.

Crystal Reports opens Essbase data directly through the P2OESBSE.DLL translation file. This file works with the complex engine that drives Essbase. The power of OLAP is combined with the power of Crystal Reports to produce the most informative and comprehensive reports.

Arbor Essbase



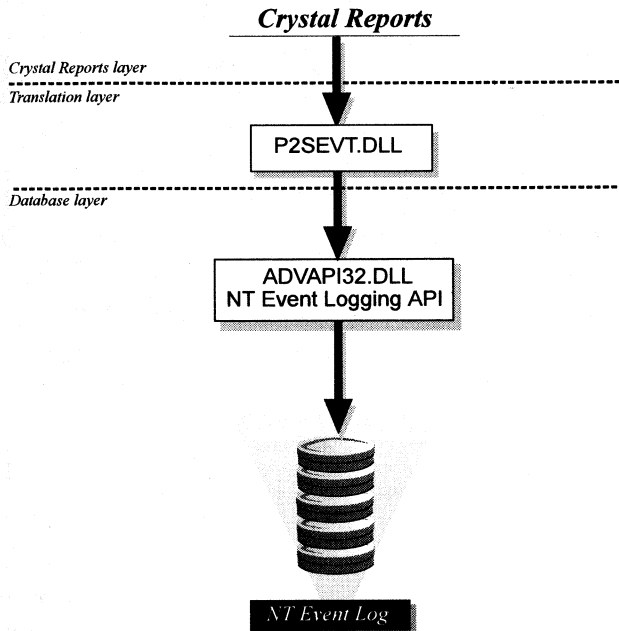
NOTE: Essbase data can only be accessed from the 32-bit version of Crystal Reports.

NT Event Log

If you use Windows NT version 3.51 or later, you can use Crystal Reports to report on the NT Event Log. The Event Log is a database used by network administrators to record and keep track of different types of events that can occur on a Windows NT Workstation or a Windows NT server.

Crystal Reports provides the translation file P2SEVT.DLL for working with NT Event Log data. This file communicates with the Event Logging API in ADVAPI32.DLL, a part of the Windows NT operating system.

NT Event Log



NOTE: The NT Event Log can only be reported on with the 32-bit version of Crystal Reports.

ODBC data sources

Open Database Connectivity (ODBC) is a standard developed by the Microsoft Corporation through which many different types of data can be accessed by a single application. An application need only communicate with one set of files, ODBC, to instantly be able to work with any source of data that can be accessed by ODBC.

There are hundreds of Database Management Systems (DBMS) available for personal computers, and thousands of applications that access DBMS data. Normally, a company that designs an application that accesses data, such as Crystal Reports, must develop a means for the application to communicate with each type of data that a customer might want to use. Crystal Reports does this with the databases that it can access directly.

On the other hand, if a DBMS simply provides a means by which ODBC can access its data, the DBMS data becomes an ODBC data source. Any application, such as Crystal Reports, can communicate with ODBC, and instantly have access to any ODBC data source. With ODBC drivers available for most common DBMS products, the possible types of data that Crystal Reports can use is almost unlimited.

Advantages

Perhaps the biggest advantage to accessing data through ODBC is the ability to access a wide range of data with just one interface. Since most popular Database Management Systems now offer ODBC drivers, with more appearing every day, Crystal Reports can use any type of data you might have.

Because of the extreme flexibility built into ODBC as well, you can use the same report file with different ODBC data sources. For example, you might design a report using an Oracle data source, then, later, if your company switches to Microsoft SQL Server, you can simply change the ODBC data source used by your report. The only requirement is that the new data source must have the same structure (tables and fields) that the original data source had (although table names can be different). See *How to change the ODBC data source accessed by a report*, Page 461.

Experienced SQL (Structured Query Language) programmers also benefit from the ODBC standard. Since Crystal Reports uses SQL to communicate with ODBC, SQL programmers and Database Administrators can view and edit the SQL statement that is sent to ODBC, controlling exactly how data is retrieved from the data source.

Finally, by using SQL pass-through technology to send an SQL statement to ODBC and retrieve an initial set of data, Crystal Reports off-loads much of the data retrieval and sorting work onto the server system, freeing up local memory and resources for your more important tasks. In addition, only the data specified by the SQL statement is returned to Crystal Reports, reducing network traffic and the use of network resources. By working more efficiently with the original data, Crystal Reports saves you time and effort and lets you concentrate on the design process and your more important work.

Disadvantages

Because of the many layers involved in passing data through ODBC from a database to an application, ODBC data sources can often take more time to return data. First, Crystal Reports must request some data. The request must be translated by the ODBC translation layer to a format that ODBC understands (an SQL statement). ODBC must determine where the requested data exists, and pass the request on to the ODBC data source (Page 503). The data source must analyze the request and translate it again into a format that can be understood by the DBMS. This complex process not only takes time, it can fail at any of several possible levels.

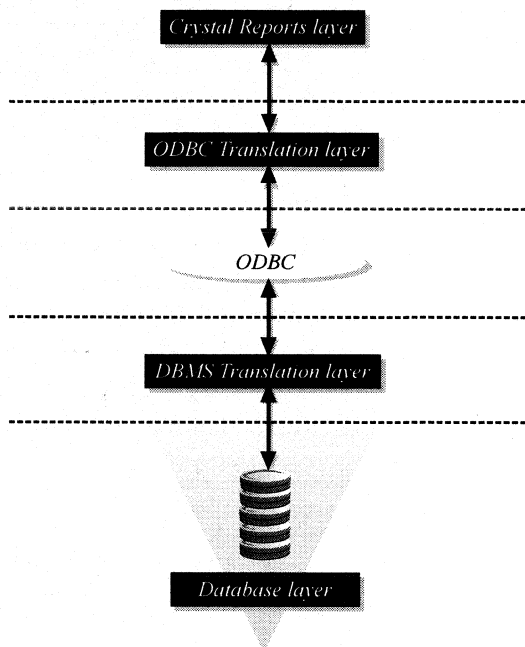
In addition, ODBC data sources must be configured and correctly set up in the ODBC.INI and ODBCINST.INI files before they can be used. If you create a report on one system and try to open it on another system that does not have the same ODBC data source set up, Crystal Reports will not be able to connect to the data.

When working with ODBC, you should also be aware that the SQL language used by ODBC is based on the standards set for the SQL language by the American National Standards Institute (ANSI). Some SQL-based DBMS applications, however, provide additional features to the SQL language that are specific to that DBMS. If your data uses features unique to your DBMS, ODBC will not be able to translate those features (though in many cases it will still retrieve most of the data). See *The SQL Language*, Page 441.

Five layers

The process by which Crystal Reports accesses data from an ODBC data source consists of five layers:

ODBC Data Source access layers



By using the Structured Query Language (SQL) to communicate, all five layers can conveniently pass data from the database to your report.

Crystal Reports

When working with ODBC data, Crystal Reports generates an SQL statement that requests the appropriate data from ODBC. The powerful SQL generator built into Crystal Reports is designed to create an SQL statement that will let the ODBC data source or ODBC itself do as much of the report generation as possible, returning only the data needed to produce the report. This SQL statement can be viewed and fine-tuned by anyone familiar with the SQL language. See *How to edit an SQL query*, Page 470.

ODBC translation

Crystal Reports uses the Dynamic Link Library PDSODBC.DLL to communicate with ODBC. This file is unique to the Crystal Reports environment and provides your report with access to any ODBC data source. This is the driver that actually passes data to and from ODBC.

ODBC

ODBC is a set of several DLLs and INI files built into the Windows environment that act as a gateway through which database requests and data can pass. Any database file or format that is to be used via ODBC must be set up as an ODBC data source.

In 16-bit Windows (Windows 3.x), the primary Dynamic Link Libraries that make up ODBC are:

- ODBC.DLL, and
- ODBCINST.DLL

Information about an ODBC data source is stored in:

- ODBC.INI, and
- ODBCINST.INI

ODBC uses these files to get and use information about the data source.

In 32-bit Windows (Windows 95 and Windows NT), the ODBC DLLs are:

- ODBC32.DLL (32-bit version of ODBC.DLL),
- ODBCCP32.DLL (32-bit version of ODBCINST.DLL), and
- ODBCINT.DLL (ODBC 2.5 and later)

Although information regarding data sources is still recorded in ODBC.INI and ODBCINST.INI, 32-bit ODBC uses the Windows Registry database to retrieve information about individual data sources.

NOTE: For complete information on ODBC and the ODBC files, see the Microsoft ODBC documentation.

ODBC uses the SQL language for all transactions between Crystal Reports and ODBC. Even if the database does not normally use SQL to create and work with tables, the ODBC driver provided by the database (the DBMS translation layer) must communicate with ODBC using SQL. For most users, this feature of ODBC is transparent, but some more advanced users often take advantage of the features of the SQL language used by ODBC.

DBMS translation (ODBC data source)

This layer consists of one or more drivers provided by a DBMS that allow ODBC to communicate with the database. Crystal Reports ships with several ODBC drivers for many of the most common database formats. If you are unsure whether or not you can use an ODBC driver to access the data in your database, refer to the documentation for your DBMS application. Most DBMS applications that run on a Windows-based platform offer an ODBC driver.

When a DBMS provides an ODBC driver, it must register the driver with ODBC on the machine it has been installed on. It does this by assigning a name to the driver and recording the filename in the ODBCINST.INI file. Usually, this step is handled automatically when the DBMS application is installed on the system. However, your network system or DBMS application may require that you register the ODBC driver manually using the ODBC Setup application. For complete information on registering an ODBC driver with ODBC, refer to the documentation for your DBMS.

Once an ODBC driver is registered, you need to establish an ODBC data source using that driver. The ODBC data source is the object that you connect to when accessing data from Crystal Reports through ODBC. Data sources are recorded in the ODBC.INI file. The data source keeps track of the DBMS translation files (ODBC drivers) and, sometimes, the database itself. An ODBC data source can specify just a database format, such as Oracle, Gupta, Sybase SQL, or MS SQL Server. Some users, however, prefer to actually specify a certain database. In this case, the ODBC data source extends across both the DBMS translation layer, and the database layer.

If you are using a client / server database, such as an SQL server, the ODBC drivers communicate with the database server through the Database Communication Layer, the same layer that your database client uses to communicate with the database server.

NOTE: ODBC drivers find their specific DBMS client files on the local machine mainly through key directories that the DBMS client has installed in the search path (specified in AUTOEXEC.BAT). The important thing is that a workstation client on a local PC must be able to connect to its server successfully. If you are not sure how to verify this, contact your IS manager.

For Crystal Reports and the ODBC drivers, it does not matter what kind of a platform your database server exists on. It is the DBMS client that connects and communicates with the server, and Crystal Reports and the drivers need only communicate with the DBMS client. See *Using SQL and SQL databases*, Page 435.

The Database

The database file referred to by the ODBC data source can be located anywhere on a system. Once the ODBC data source is set up, Crystal Reports does not need to know the actual location and format of the data. Thus, the database can have any format and be located anywhere on a network, as long as ODBC can communicate with it through the ODBC data source.

Common ODBC database formats

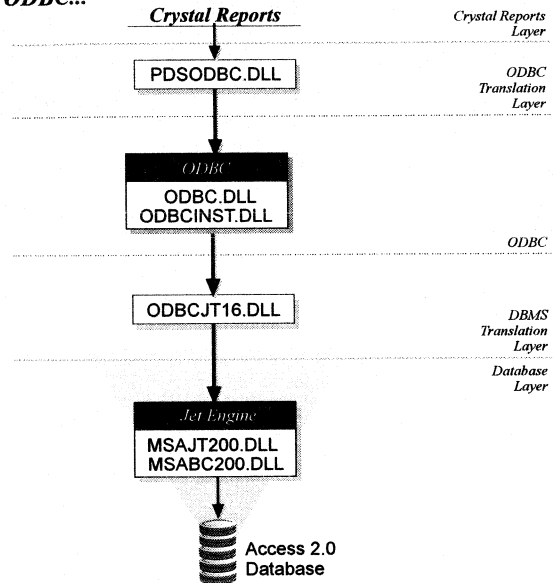
Access

Microsoft Access 2.0 files use ODBCJT16.DLL to communicate with ODBC. This file handles all communication between ODBC and the Microsoft Jet Database Engine. It is an example of a DBMS specific ODBC driver.

NOTE: If you are using a version of MS Access other than 2.0, refer to the Runtime File Requirements Help file for information on files used to open data in your version of Access.

The Jet engine is a part of the Access DBMS that does all of the actual work with an MS Access database. Jet is a required part of MS Access regardless of the method used to read the Access data. Since the Jet engine is an integral part of all Access databases, it is considered a part of the database layer.

Access via ODBC...



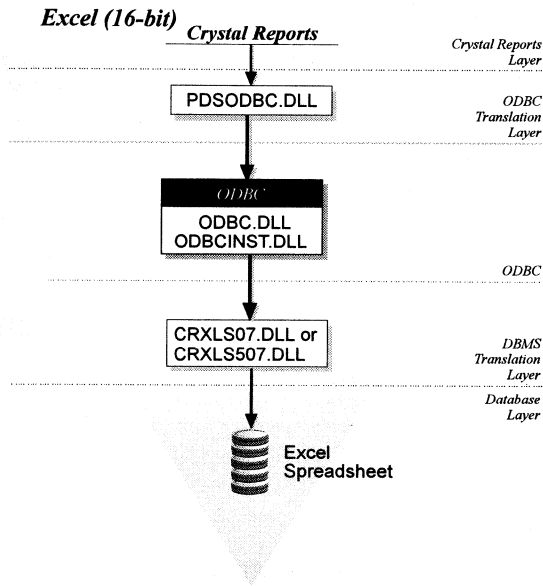
Excel

Microsoft Excel spreadsheets can be converted into databases that can be read by Crystal Reports through ODBC. In Excel 4.0 and earlier, use the Set Database command on the Data menu. In Excel 5.0, use the Name | Define command on the Insert menu. Once converted, spreadsheet rows become records, and spreadsheet columns become fields. For more information on converting your spreadsheets to database format, refer to your Excel documentation. Once the spreadsheet is converted, you can set up an ODBC data source for the file, then select it from Crystal Reports.

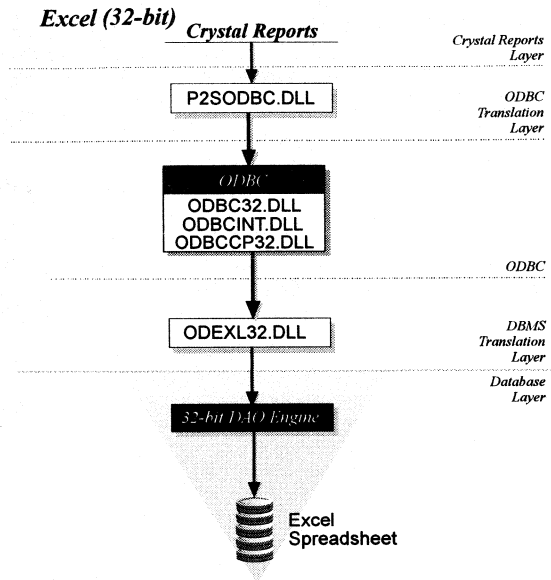
NOTE: If you are using Excel 7.0, you can export your spreadsheets as Access database tables, and read them from Crystal Reports as you would read other Access tables. Refer to your Excel documentation for more information.

Excel databases for 16-bit Windows environments are read through the CRXLS07.DLL translation file, for version 4.0 and earlier of Excel, or through the CRXLS507.DLL for version 5 of

Excel. These drivers are installed by Crystal Reports. ODBC can communicate with this driver to read the converted Excel spreadsheet.



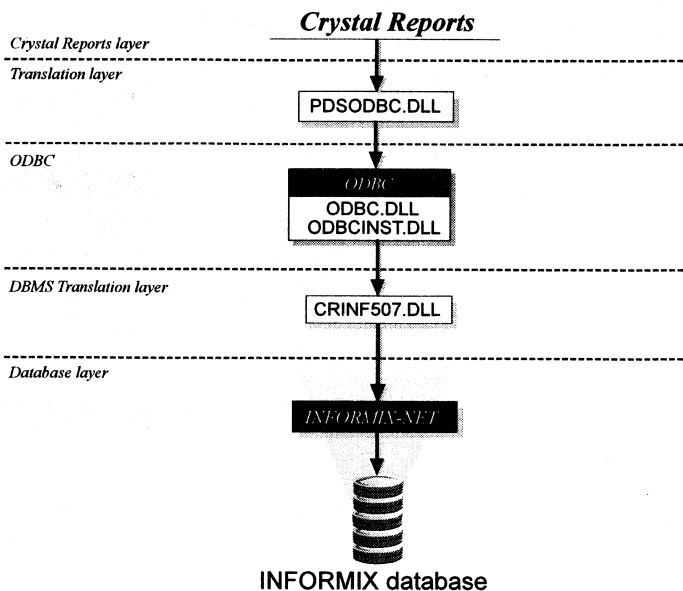
If you are using a 32-bit Windows system, you can set up an ODBC data source for 32-bit Excel spreadsheets using the 32-bit DAO engine. The DAO engine is installed on your system when you install the 32-bit version of Crystal Reports. You must, however, set up an ODBC data source manually for your Excel spreadsheet. See *How to set up an ODBC data source*, Page 453.



INFORMIX

Crystal Reports accesses INFORMIX databases through ODBC. The INFORMIX client, called INFORMIX-NET, must be installed on your machine. Otherwise, Crystal Reports provides the necessary driver that ODBC uses to communicate with the INFORMIX database engine.

INFORMIX



NOTE: The 32-bit version of Crystal Reports provides the 32-bit ODBC driver CRINF509.DLL for reading 32-bit INFORMIX data.

SQL Databases through INTERSOLV DataDirect

There are many different SQL Database Management Systems available on the market, and most of them can be accessed through ODBC. Crystal Reports automatically installs and sets up several ODBC data sources allowing you to access many of the most popular SQL databases, including:

- Oracle
- Sybase SQL Server
- MS SQL Server
- Gupta SQLBase
- Scalable SQL
- DB2/2
- ASCII Text

NOTE: ASCII Text is not really an SQL database format, but text files can be read by Crystal Reports using ODBC in the same way that many SQL databases are.

Crystal Reports provides the INTERSOLV DataDirect ODBC drivers for these SQL database formats. Not all SQL databases are accessed through these drivers, but, as a convenience, Crystal automatically sets up these drivers for you when installed. Crystal Reports still communicates with ODBC using the PDSODBC.DLL translation file, but the DataDirect drivers provide ODBC with easy access to the actual databases.

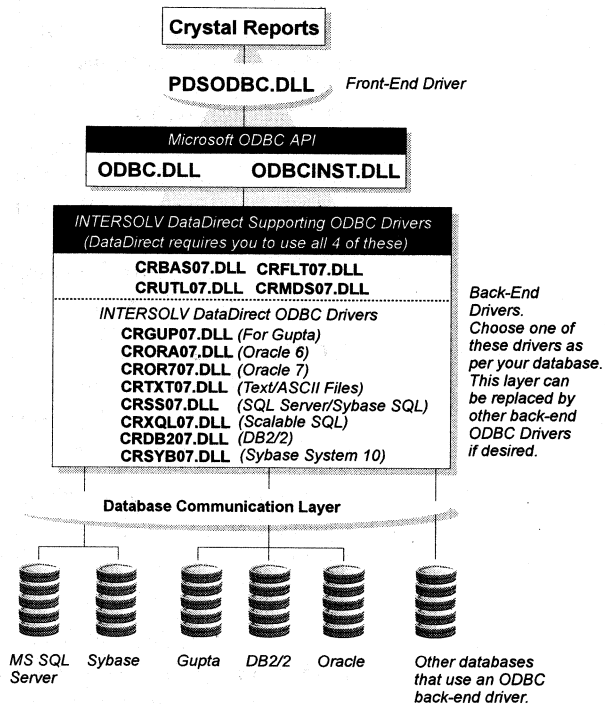
NOTE: If you are using an SQL database not accessed by the INTERSOLV drivers, refer to your DBMS documentation for complete information on the ODBC drivers required. See How to set up an ODBC data source, Page 453.

Do not confuse SQL databases with the SQL language. Structured Query Language (SQL) is a specialized computer language used to design, build, and read database files. See *Using SQL and SQL database*, Page 435.

The term SQL databases is used to refer to the collection of computer applications that depend on the SQL language for database creation and manipulation. SQL DBMS applications are usually designed to run as large client/server applications spread across a network.

To open an SQL database, ODBC uses a set of four DataDirect ODBC support drivers (these are the same for every SQL database format), a DataDirect ODBC driver specific to the database format, and the Database Communication Layer that actually communicates directly to the database file. The files Crystal Reports provides for this DBMS translation layer are all INTERSOLV DataDirect drivers.

NOTE: ODBC does not have to use the INTERSOLV DataDirect drivers to access the SQL databases shown here. Most SQL DBMS applications provide their own ODBC drivers. However, Crystal Reports sets up the INTERSOLV drivers for you automatically.



NOTE: The 32-bit version of Crystal Reports provides 32-bit INTERSQLV drivers for accessing SQL databases. These 32-bit drivers are named CR*09.DLL instead of CR*07.DLL.

As mentioned previously, Crystal Reports automatically creates several ODBC data sources for the SQL database formats supported by INTERSQLV DataDirect. These data sources appear in the Log On Server dialog box:

- ODBC-CRDBM supports DB2/2
- ODBC-CRGUP supports Gupta
- ODBC-CROR7 supports Oracle 7
- ODBC-CRORA supports Oracle 6
- ODBC-CRSS supports MS SQL Server and Sybase SQL Server
- ODBC-CRXQL supports Scalable SQL
- ODBC-CRSYB supports Sybase System 10

INTERSOLV DataDirect Library

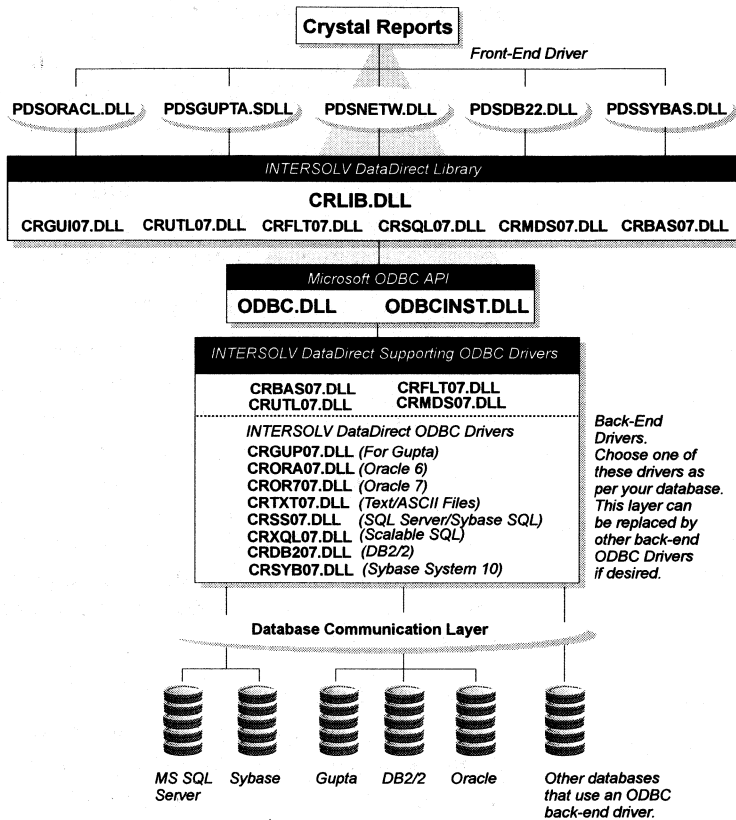
Instead of communicating directly with ODBC through the PDSODBC.DLL file, Crystal Reports can use the INTERSOLV DataDirect Library that ships with Crystal Reports. These front-end drivers use the ODBC standard as an underlying layer. However, instead of using PDSODBC.DLL to communicate with ODBC, Crystal Reports uses a specific PDS*.DLL file appropriate to the SQL database being accessed.

DataDirect Library vs. ODBC

The strength of the DataDirect Library over ODBC direct is mainly in its ability to pass proprietary SQL syntax to specific servers. For example, because Oracle has its own PDS driver, an SQL developer working with Oracle data is able to pass, at times, more precise SQL statements to the server. This way, SQL language features unique to a particular DBMS can be passed from Crystal Reports. When using ODBC direct, the PDSODBC.DLL is a common DLL that is used for all DBMS types, and proprietary SQL syntax based on specific servers may not be entirely possible.

DataDirect Library structure

With the INTERSOLV DataDirect Library, Crystal Reports uses an ODBC translator specific to the SQL database format being accessed. This file, in turn, communicates with the DataDirect Library. The DataDirect Library makes the final translation of the SQL statement to ODBC.



NOTE: The 32-bit version of Crystal Reports provides 32-bit INTER SOLV drivers for accessing SQL databases. These 32-bit drivers are named CR*09.DLL instead of CR*07.DLL.

INTER SOLV DataDirect Library databases

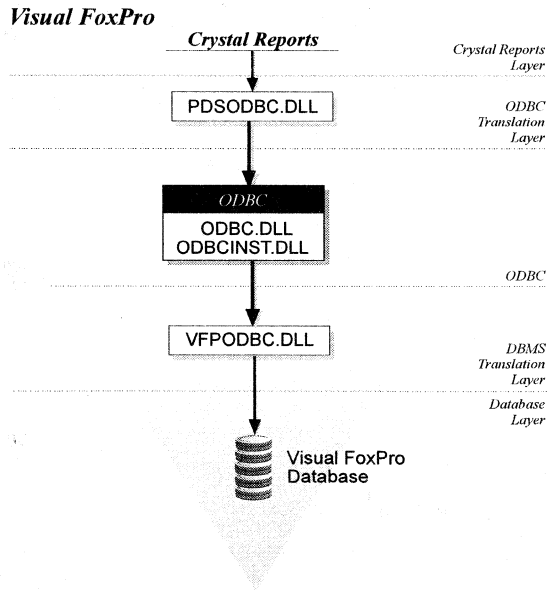
- Oracle
- Sybase SQL Server
- MS SQL Server
- Gupta SQLBase
- Scalable SQL
- DB2/2

NOTE: The ODBC data sources that Crystal Reports installs for the INTERSOLV DataDirect Library appear in the Log On Server dialog box without the "ODBC-" prefix used by the data sources that go through ODBC direct.

Visual FoxPro

Microsoft Visual FoxPro data is accessed through ODBC, whereas FoxPro data from version 2.5 and earlier is accessed directly through the xBase engine. If you are using FoxPro version 2.5 or earlier, see Page 480.

Crystal Reports provides the ODBC driver VFPODBC.DLL to allow ODBC to work with Visual FoxPro data.

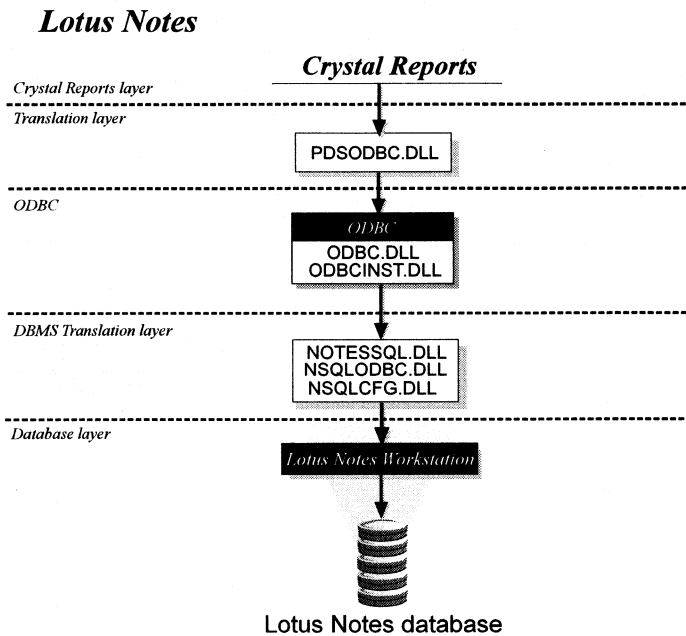


Lotus Notes

A Lotus Notes database can be read by Crystal Reports through ODBC. The Lotus Notes DBMS translation layer consists of three files:

- NOTESSQL.DLL
- NSQLODBC.DLL
- NSQLCFG.DLL

These files use the drivers installed by the Lotus Notes Workstation to work with the Lotus Notes database. The Workstation component of Lotus Notes must be installed on the local machine.



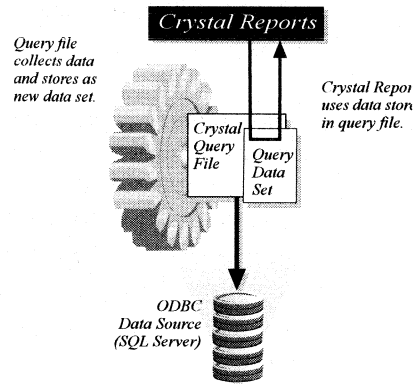
Crystal Query data sets

A query is simply a request for a specific set of data from a database. Once the data is gathered, it can be stored as a Crystal Query file (.QRY). The Crystal Query file can then be used much like a database table. The data it gathers from your SQL databases becomes a new data set that can be added to reports.

A Crystal Query gathers data from ODBC data sources, such as SQL servers, by means of the Structured Query Language (SQL). An SQL statement with instructions to gather and return a specific

set of data is sent directly to the SQL DBMS. The DBMS handles all of the actual data gathering, sorting, and grouping according to the instructions in the SQL statement. Only the final set of data is returned to the Crystal Query Designer and stored as a query file.

NOTE: If you are not familiar with SQL, see Page 426.



First, you refine an SQL statement and create a query file that contains only the data you need. Then, you design your reports based on the query file instead of the original databases. Most of the data gathering process is done on the SQL server when you design the query. When you design your report, Crystal Reports needs only work with a small set of data, saving you time and trouble. See *How to create a new query*, Page 364.

NOTE: The term query has been used through out this chapter to refer to several similar concepts. For this section, however, the term query refers strictly to a Crystal Query file unless otherwise indicated.

Advantages

While there are many reasons for using queries, the primary reason is the ability to off-load most of the data retrieval process to a server and store the resulting data as a separate data set. Using this SQL pass-through technique, where data retrieval tasks are passed through to the server, you retrieve your data faster and more efficiently. Anytime you are working with SQL data, you should consider building a query file before designing your report.

In addition, the Crystal Query Expert provides for more control over your data with the SQL language. Crystal Reports provides some SQL language features when accessing SQL or ODBC data, but the Crystal Query Expert is a powerful query tool that provides more powerful SQL data access features. The Query Expert allows aggregate functions in an SQL statement and supports all data query elements of the SQL language. If you know the SQL language, you can perform complex SQL tasks using UNION operations and sub-queries. See *The SQL Language*, Page 441.

Disadvantages

Since a query represents a complete data set, any records, fields, or tables that were not included when the query was created are not available when you use the query to build a report. A report can not be any more complex than the data it accesses, so by using a query, you limit your reporting options to the data in the query.

In addition, a query can not be used with any other source of data. If you decide to use a query in your report, you can not select an additional source of data, such as a database table or another query file, to use in the report.

Crystal Dictionaries

A Crystal Dictionary file (.DC5) is a structured and simplified view of organizational data that you can create for some or all of the individuals in your organization that are using Crystal Reports. With an Dictionary, end users only see the subset of tables and fields they need.

NOTE: The Crystal Dictionaries application for creating Dictionary files is only available with the Professional edition of Crystal Reports. This application is designed for IS and Database Managers who need to provide alternatives to accessing data for users. Both the Standard and Professional editions of Crystal Reports can open and use Dictionary files created with Crystal Dictionaries.

A Dictionary is an optional source of data for Crystal Reports. It provides a convenient filter, clarifying and simplifying complex data access techniques for end users, but data can still be accessed directly from the database by the user. Dictionaries simply provide all the convenience without enforcing restrictions.

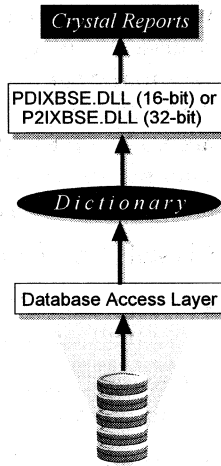
If you use a Dictionary to design a report, however, you can only use that Dictionary as a source of data for that report. In other words, you can not open both a Dictionary and an ODBC data source, a direct access database, or any other source of data from within the same report. This includes other Dictionaries. The purpose of a Dictionary is to provide fast, easy to understand access to a particular set of data. Connecting to other sources of data within the same report would defeat this purpose. See *How to create a new Dictionary*, Page 385.

Crystal Dictionaries let you:

- Design a single, dynamic view of all the data that is necessary to create organizational reports and queries,
- Create a Dictionary that contains multiple data sources, tables, and links,
- Organize the data and rename tables and fields to make it easier for users to understand the content and purpose of the data, and
- Create complex data-manipulation formulas that users can access without the need to understand formula concepts.

Dictionaries reduce support cost and time, increase user productivity, and reduce data misuse, loss, and damage.

Once created, the Dictionary acts as a filter, providing a view of complex data that is clear and easy for any user to understand. Complicated data will not stop end users from creating powerful reports.



Crystal Reports reads the Dictionary file using the PDIXBSE.DLL file (P2IXBSE.DLL for 32-bit). This translation file is based on the xBase engine used to access dBASE, FoxPro, and Clipper databases (Page 480), but provides all the flexibility needed to read Dictionaries.

NOTE: If you upgraded from an earlier version of Crystal Reports that allowed Crystal Dictionary files, the older Dictionary files (.DCT) were based on the Btrieve engine. If you convert those older files to new Dictionary files (.DC5), the new files will use the xBase translation layer instead of Btrieve. However, the Btrieve engine installed by the earlier version of Crystal Reports must remain on your system for the new Crystal Dictionaries application to read the older Dictionary files.

The Database Access Layer is the layer through which the Dictionary file actually reads the original database data. Dictionaries must read database data through the same routes that Crystal Reports reads the data. The section on direct access database layers and ODBC data sources that are described throughout this chapter illustrate the files needed for Dictionaries to open data.

The benefit of the Dictionary is that this Database Access Layer is completely transparent to the user. The Dictionary displays a

view of data in Crystal Reports that may or may not match the actual data, but displays a view that is easily accessible and usable by users.

Finally, Dictionaries provide an easy means of changing the original data accessed without changing the view of data seen by users. For example, your original database may contain Price, Quantity, and Cost fields, but you decide that Cost can be calculated from Price and Quantity, so you eliminate the Cost field. By simply updating the Dictionary to calculate Cost instead of getting it directly from the database, your users never know the difference. This also helps when the entire underlying database format is changed. Once again, just update the Dictionary.

Crystal Dictionary files are often created by the IS manager who controls a company database. The IS manager can work with the data they have organized and classified to create simple, easy Dictionaries for company employees. The employees need only open the Dictionary file with the data they need, without the difficulty of understanding and working with the original data.

21

Graphing

What you will find in this chapter

Graphing Overview 522

Before you create your graph 522

The Graph/Chart Expert 522

Where to place your graph 524

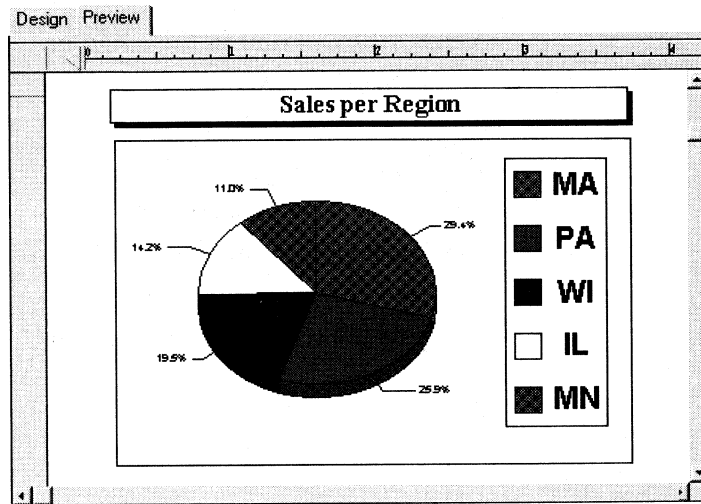
The data you can graph on 524

Editing your graphs - PGEEditor 525

Using the underlay feature with graphs 525

Graphing Overview

Crystal Reports enables you to include sophisticated, colorful charts and graphs in your reports. For example, you can use graphs to show 1995 fiscal results or the sales within the fiscal quarters. You can use graphs any time you want to improve the usability of your report.



Before you create your graph

Graphing is done on summarized data at the group level. Before you can create a graph, you must have at least one group and one summary or subtotal in your report. In other words, you are graphing summary and subtotal information.

For example, if you have a sales report grouped by Region and a subtotal of Last Year's Sales for each Region, you can quickly create a graph that will display sales by Region.

The Graph/Chart Expert



Creating and modifying your graphs is easy in Crystal Reports with the use of the Graph/Chart Expert. Use the Expert every

time you create or modify graphs. The Expert walks you through the creation of your graphs step-by-step. Click the INSERT CHART button on the toolbar to access the Graph/Chart Expert.

NOTE: If your report does not contain a group and a summary or subtotal for that group, you will not be able to access the Graph/Chart Expert.

Types Tab

When you begin creating your graphs, you will need to choose one of the many graph types available. While you can graph the same data using many different graph types, there are specific graph types that are best used for graphing certain types of data. For example, if you are displaying the total sales for your top four products, you may use a pie chart. Whereas, if you wanted to display monthly sales over the last year you could use the line chart. Search for *When to use different graph types* in Crystal Reports online Help. To select a graph type, click the button that corresponds to the graph type you want.

The PG Editor button takes you into the PGEEditor, an add-on product that gives you sophisticated graph editing capabilities. For a basic overview of the PG Editor, see the *Editing your graphs - PGEEditor* section later on in this manual. For specific information, use PGEDITOR.HLP, the Editor's own help system that is located in the same directory as the program.

Data Tab

In the Data Tab you select what data you want to graph on, where you want the graph displayed, and where you want to show the graph.

The *graph on* drop-down box displays all the summaries and subtotals on your report. Simply select which one you want to base your graph on.

Text Tab

The Text Tab allows you to enter labels for your graph by inserting titles, footnotes, and axis titles when appropriate. Simply type the text in to the appropriate boxes. Only the labels appropriate for the graph type that you chose earlier will be used when graphing. The others will be ignored.

Options Tab

If you want a legend on your graph, a maximum or minimum limit on the graph values, or a number of other options, use the Options Tab.

For additional information on the Graph/Chart Expert, search for *Graph/Chart Expert* in Crystal Reports online Help.

NOTE: The Options Tab will disappear if you go into the PGEditor as these options can be set in the Editor as well.

Where to place your graph

Where you place your graph determines what data gets displayed and where it is printed. For example, if you place a graph in the Report Header section, the graph includes data for the entire report. If you place it in a Group Header or Group Footer section, it displays group specific data. Search for *Where to place your graph* in Crystal Reports online Help.

NOTE: If your report contains subreports, you can place graphs in those subreports as well. See Chapter 15, Subreports, Page 335.

Drill down with graphs

Not only is a graph a powerful way of presenting data, it is also a powerful analysis tool. When you place your mouse pointer over a segment of the graph in the Preview Tab your mouse will become a drill down cursor. You can double-click to see the details supporting that section of the graph.

NOTE: Drilling down on graphs is only available within Crystal Reports. If you use the Crystal Report Engine to distribute reports with your application, the drill down feature is not available in those reports.

The data you can graph on

You can graph only on summaries and subtotals that are created using the Insert Summary or Insert Subtotal commands. You can subtotal or summarize formulas and then graph those summaries, but you must use the summary/subtotal commands to do so. You cannot graph on summaries that you create using summary functions in formulas.

Editing your graphs - PGEEditor

The PGEEditor is a powerful graph editor offering you 80 graph types and full formatting control of every graph element. The Editor is commonly used to format the graph title, change the colors of graphs, and resize portions of the graph.

A very popular feature of the Editor is the ability to save graph formatting templates. These templates are like stylesheets for graphs. They contain custom formatting instructions that can be applied instantly to new graphs. If you continually create pie charts, for example, and you need the charts to appear in custom colors, you can create a template with those colors. Then, whenever you create a new graph, you simply apply the template and the graph is formatted to your template specifications.

Using the underlay feature with graphs

The ability to format a section so that objects in the section underlay other sections makes graphs a more powerful tool than ever before. Now, instead of having a graph print before the data it represents, you can have the graph appear beside it, to make the data more understandable. Search for *Using the underlay feature with graphs* in Crystal Reports online Help.

22

OLE

What you will find in this chapter...

OLE Objects Overview 528

How to insert a graphic/picture as an OLE object 530

OLE Objects Overview

An OLE object is, broadly speaking, a presentation of data that was created in another application and that maintains a relationship with the application that was used to create it. When you place an OLE object in a report, the object becomes part of your report. If you want to edit the object, however, you simply double-click it and modify it using the editing tools from the application that you used to create the object in the first place (or a similar application that allows such editing). This applies to all but static OLE objects. Using OLE objects in your report enables you to create robust documents that are easy to maintain.

Crystal Reports is an OLE container application; it is not an OLE server application. You cannot at this time create a report and use that report as an object in another application.

There are several ways of getting OLE objects into your application.

- You can cut the objects from the OLE server application and paste it in your report using Paste Special on the Edit menu. This creates an embedded object. If the object could be pasted in multiple formats, the program decides which format to use.
- You can cut the object from the server application and paste it in your report using the PASTE SPECIAL command on the Edit menu. If the object could be pasted in multiple formats, you decide which format to use. Using this command you can place either embedded or linked objects.
- You can create a new object or import an existing one using the OBJECT command on the Insert menu. You can place either embedded or linked objects this way.
- Finally, you can drag and drop an object from an OLE server application. You do this by opening Crystal Reports in one window and the server application in another, then dragging the object between the two. When you drag an object into a report, the object is embedded, not linked.

NOTE: When you have a linked object and you break the link using the Links dialog box, the program turns the object into a picture, not into an embedded object.

LINKED OBJECTS

When you insert a linked object, Windows copies a snapshot of the data from a file that already exists. Only the image of the object is added to your report. The actual data remains with the original file.

When you open the object from within your report, the original file is opened inside the application that was used to create it. Any changes you make directly change the original file.

If you want the data in your object to remain available to other applications, and to always reflect the most current changes to the data, link the object to your report.

EMBEDDED OBJECTS

You can create an embedded object from within Crystal Reports, or can be created by copying information from a file that already exists. The data, as well as the image of the object, is saved with your report. The object can not be accessed outside of your report.

If you want to create a new object for your report from scratch, or if you want to copy the information from an existing file and edit the data without ever affecting the original file, embed the object in your report.

- If you change the original object, the embedded object does not change.
- If you edit an embedded object, changes you make to the object are stored with the *.RPT file; they do not affect the original.

OLE and the Insert Picture command

If you place pictures in your report using the PICTURE command on the Insert menu, Crystal Reports converts them into static OLE objects to enhance online and printing performance. Static objects do not support in place editing, however. If you want to edit the object in place, you will have to convert it to an editable type of object using the CONVERT command on the Edit menu.

NOTE: *If you place database bitmap fields into your report, the program displays them as bitmaps; it does not convert them to OLE objects.*

OLE CONSIDERATIONS

- When you double-click an embedded OLE object, Crystal Reports changes its menus and tool bars to those of the object's server application. When you are finished editing, click outside the object and the Crystal Reports tools reappear.
- When you double-click a linked OLE object, the program opens the object's server application with the object displayed, ready for editing. You can not edit a linked object in place in Crystal Reports because you are working on the original object. Since the object could be linked to multiple documents, and since, conceivably more than one person could want to edit it at a given time, displaying the original in the server application limits access to one editor at a time.
- Windows 3.x shipped PBRUSH.EXE as the native bitmap editor. Windows 95 is shipping MSPAINT.EXE as the new bitmap editor. While Windows 95 knows to register and convert PBRUSH.EXE OLE bitmaps as MSPAINT OLE bitmaps, the opposite is not true. While you can open a report created in Windows 3.1 containing OLE bitmaps using Windows 95, you can not open a Windows 95 OLE bitmap report in Windows 3.x. Thus, if you are going to be working back and forth between these two operating systems, it is best to create the reports under Windows 3.1 instead of under Windows 95.

HANDS-ON

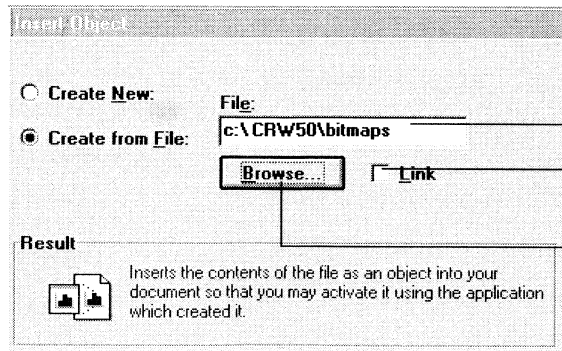
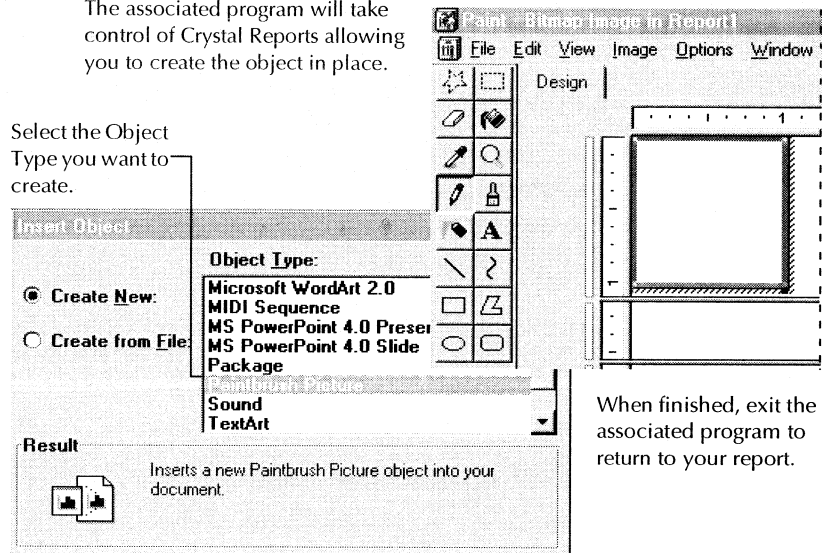
How to insert a graphic/picture as an OLE object

Often when you insert a graphic/ picture into a report, it will be necessary to change that graphic. Normally to edit this graphic/ picture you would have to exit Crystal Reports, open the original application, change the graphic, return to Crystal Reports, delete the graphic originally inserted and insert the newly revised graphic. All of these steps can be avoided using Object Linking and Embedding (OLE). OLE allows you to insert data or images into a report from another application and then use that application from within Crystal Reports to edit the graphic if

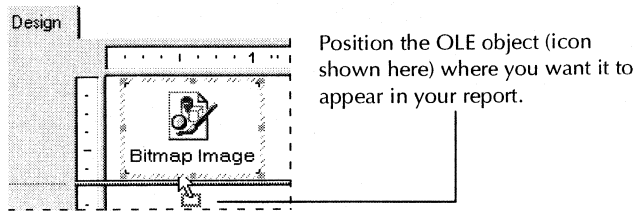
necessary. Crystal Reports is an OLE container and can receive graphics created in any application that supports OLE.

Choose the OBJECT command from the Insert menu. The Insert Object dialog box appears. You have two options when it comes to OLE objects: you can insert an already made object or create a new one directly in Crystal Reports using in-place editing.

The associated program will take control of Crystal Reports allowing you to create the object in place.



A frame approximating the size of the object (or icon) appears.



- If you have chosen to display your object as an icon, the icon will appear in your report.
- If you have created your object from an existing file, the data from that file (or an icon) will be displayed in your report. You can edit that data if you wish by double-clicking the object or its icon.
- If you are creating a new object, the object application for the object type that you chose will open, and you can begin creating your object. When finished, close or exit the application. Your object or its icon will be displayed in your report.

Related Topics

Search for *Insert Object dialog box* in Crystal Reports online Help.

23

Commands

What you will find in this chapter...

File Menu Commands 534

Edit Menu Commands 539

View Menu Commands 544

Insert Menu Commands 546

Format Menu Commands 553

Database Menu Commands 557

Report Menu Commands 560

Window Menu Commands 564

Help Menu Commands 565

Menu commands are the tools you use to create, customize, print, and save your reports. Each Crystal Reports menu contains a list of commands that are related by their function (i.e., all those commands related to files such as opening, closing, saving, etc., are found in the File menu).

NOTE: If a command is also available as a button on the toolbar, the button is pictured with the command.

NOTE: To get to a menu without using your mouse, press the ALT key and the underlined letter in the menu name.

FILE MENU COMMANDS

New



When you choose the NEW command, the Report Gallery appears. Choose from the following experts, or click *Custom* to create a customized report of your own:

- Standard Expert
- Form Letter Expert
- Form Expert
- Cross-Tab Expert
- Subreport Expert
- Mail Label Expert
- Top N Expert
- Drill Down Expert

Related Topics

For more information on experts, search for the expert by name in Crystal Reports online Help.

Open



Use the OPEN command to open an existing report. When you choose the OPEN command, the File Open dialog box appears. Use this dialog box to select the report file you wish to open.

Close

Use the CLOSE command to close the current report while keeping the program active. The program will prompt you to save the report if it needs saving.

Save



Use the SAVE command to save the active report to disk under its current name.

- All changes you have made while working on the report will be saved.
- The previous version of the report will be overwritten.

If you have not previously saved your report, the program will prompt you to do so.

Save As

Use the SAVE AS command to save the active report to disk under a new name. All changes you have made while working on the report will be saved to the new file. Your original report file will remain unchanged.

Save Data with Report

Typically, when you save a report, Crystal Reports saves a report definition and the underlying data. The definition contains all of the information needed for printing your report. The Data is a snapshot of the data as it existed when you ran the report. If you turn the *Save Data with Report* option off via the OPTIONS command on the File menu, the program saves only the report definition. The data must be refreshed each time the report is run. To toggle the option on for the current report, use this command.

There are some things to consider before using the SAVE DATA WITH REPORT command.

- If you save only the definition:
 - your report will require somewhat less disk space (enough for the template only), but
 - the program will need to retrieve the data before it prints the report.

- If you save your data with the report:
 - your report will require somewhat more disk space (enough for the report template and the data), but
 - the program will not need to retrieve data before it prints the report.

Toggle this command on or off to suit your needs. This option is toggled on by default. To change the default, choose Options from the File menu and change the Save Data With Closed Report option on the Reporting tab of the File Options dialog box when it appears.

NOTE: When the program saves data with the report, it compresses the data to take up less disk space. It decompresses the data when it opens the report.

Print Preview



Use the PRINT PREVIEW command to see what your report will look like before you print it and to fine tune the report while working with the actual data.

Print



Use the PRINT command to select a printing option.

Preview Sample

Use the PREVIEW SAMPLE command if you want to print a draft report to a window using only a subset of the data you will use in your final report. This feature can save you a great deal of time during the report creation process.

Printer

Use the PRINTER command to print a hard copy of your report to the currently selected printer.

Export



Use the EXPORT command to export your report in one of several popular word processing, database, or spreadsheet formats, in HTML or ODBC format, or in one of a number of standard data

interchange formats. For example, you can export your report in Microsoft Excel format and then open it in Excel as a normal spreadsheet file. In addition to selecting your export format, the program lets you pick your export destination as well, either to disk file or to e-mail.

Mail

Use the MAIL command to export your report to e-mail. When you choose the MAIL command, the Export dialog box appears. You use that dialog box to specify an export format for your report.

Printer Setup

Use the PRINTER SETUP command to select the printer you want to use to print the report. If you do not select a printer, Crystal Reports will print to the Windows default printer.

Set Label Layout

The SET LABEL LAYOUT command is available on the File menu whenever you are working on mailing labels or other label-type items. This command returns you to the Mailing Labels dialog box, the dialog box you used to set your label specifications when you used the Custom Mailing Label command. Using the Mailing Labels dialog box, you can adjust the label dimensions, the gaps between labels, the page margins, and /or the label printing direction.

Page Margins

Use the PAGE MARGINS command to set page margins for your report. Margins define the white space between your report and the edges of the page.

NOTE: Crystal Reports uses the non-printing areas established for your printer as default printer margins. Those margin settings appear in the Printer Margins dialog box. Although you can set margins that fall inside the non-printing areas, parts of your report may be clipped off if you do.

Options

Use the `OPTIONS` command to modify dozens of settings that apply to the Crystal Reports environment. Using these settings, you can customize Crystal Reports to fit the way you work.

This command takes you to the File Options dialog box. Once there you can change default settings in the following areas: Layout, New Report, Database, SQL, Reporting, Fields, and Fonts. Your settings here affect new reports and fields added to existing reports, but they do not affect the work done before you change the options.

Report Options

When you choose the `REPORT OPTIONS` command, the Report Options dialog box appears. Use this dialog box to set a number of technical options that apply only to the active report. Your specifications are saved with the report and do not affect any other reports. This is a useful command for individuals who are creating reports for distribution.

Summary Info

Use the `SUMMARY INFO` command to:

- add summary data that can help you and other users identify the report, and
- view the statistical information the program generates for each report.

SUMMARY DATA

You can supply any or all of the following kinds of information about your report: author, keywords, comments, title, subject, and template. Summary information does not print with the report; it is simply a readable part of the electronic report file. You can use this command to view and modify summary information as well.

STATISTICAL INFORMATION

Crystal Reports tracks such things as who last worked on the report, when it was created and last saved, and so forth. This information is read-only; you can not modify it directly.

Exit

Use the EXIT command to end your Crystal Reports session and return to the Windows environment.

EDIT MENU COMMANDS

Undo



The UNDO command is a dynamic command. The text changes on the menu based on the last action you performed in Crystal Reports. Undo reverses the effects of an action and restores your document to the way it was before you performed the action. Crystal Reports supports multiple levels of Undo.

Redo



The REDO command is a dynamic command. The text changes on the menu based on the last Undo you performed in Crystal Reports. Redo restores the effects of an Undo and returns your document to the way it was before you used Undo. Crystal Reports supports multiple levels of Redo.

Cut

Use the CUT command any time you wish to cut or delete selected text temporarily and hold it in the Clipboard for later use.

NOTE: The Cut command can be used in the Formula Editor (for any formula element) using the Windows keyboard command Ctrl-X.

Copy

Use the COPY command any time you wish to hold a copy of text in the Clipboard for later use without disturbing the text as it appears currently in the report.

NOTE: *The Copy command can be used in the Formula Editor (for any formula element) using the Windows keyboard command Ctrl-C.*

Paste

Use the PASTE command any time you want to paste (insert) text from the Clipboard into your report.

NOTE: *The Paste command can be used in the Formula Editor (for any formula element) using the Windows keyboard command Ctrl-V.*

Paste Special

The PASTE SPECIAL command allows you to paste an OLE object into your report from the Clipboard.

Formula



Use the FORMULA command to edit a formula that you have highlighted in your report. When you choose this command, the Formula Editor appears with the formula text displayed, ready for editing.

Text Object

Use the TEXT OBJECT command to edit the currently selecting text object.

Summary Field



The SUMMARY FIELD command allows you to change the operation for the selected summary. Using this option, for example, you can change a sum operation to one that determines the maximum (highest) value in the group.

Subreport Links

Use the SUBREPORT LINKS command to link the data in a subreport to data in the parent report. This enables you, for example, to create a subreport that provides the details behind summary data in the parent report. This command works in conjunction with the PARAMETER FIELD command on the Insert menu.

Browse Field Data

Use the BROWSE FIELD DATA command to review the values in a report field.

NOTE: *This command is active only when you have selected a field on your report.*

Show/Hide Sections



Use the SHOW/HIDE SECTIONS command to toggle the *Hide (Drilldown OK)* property for a report section on and off. When the *Hide* property is toggled on, the selected section will print.

This command takes you to the Section Expert that you use to select the section you want to show/hide.

Move Section



Use the MOVE SECTION command when you have multiple sections in an area and you want to change their order. For example, if you have three Detail sections, Detail A, Detail B, and

Detail C and you want to move Detail C so it runs before Detail B, you use this command to move the sections.

This command takes you to the Section Expert that you use to select the section you want to move.

Merge Section



Use the MERGE SECTION command when you have multiple sections in an area and you want to combine two of the sections into a single section. For example, if you have three Detail sections, Detail A, Detail B, and Detail C and you want to combine Detail B and Detail C into a single section, you use this command to merge them.

This command takes you to the Section Expert used to select the sections you want to merge.

Delete Section



Use the DELETE SECTION command when you have multiple sections in an area and you want to delete one of them. For example, if you have three Detail sections, Detail A, Detail B, and Detail C and you want to delete Detail B, use this command to delete it.

This command takes you to the Section Expert used to select the section you want to delete.

NOTE: You cannot delete any of the sections that appear originally when you create a new report.

Object

Use the OBJECT command to modify an OLE object that you have embedded in a report. The command is active only when you select an embedded object. The command has three subcommands that appear on a flyout menu

Edit

The EDIT command enables you to edit the object in Crystal Reports using the tools from the object's originating (server) application.

Open

The OPEN command opens the object's originating (server) application with the object active, ready for editing. The difference between the OPEN command and the EDIT command is that:

- EDIT allows for in-place editing: you edit the object without ever leaving Crystal Reports while
- OPEN takes you out of Crystal Reports to do your editing.

Convert

CONVERT enables you to convert the object to another object type. You use this command typically when you want to edit an object but you do not have the server application installed on your computer. Using this command, you can change the object to another format so you can edit it in an application you do have.

NOTE: When you select an object, the program identifies the Object command with the object type. For example, if you have selected a bitmap picture object, the program will identify the Object command as Bitmap Image Object.

Links

Use the LINKS command to make changes in existing OLE object links. Using this command, you can change the source data for a link and the way a link is updated. This command is only active when you have an object link in the report.

NOTE: This command is not related to editing a database link. To edit a database link you must choose the Visual Linking Expert command from the Database menu.

VIEW MENU COMMANDS

Toolbar

Use the TOOLBAR command to toggle the toolbar on and off. By default the toolbar is on (a check mark appears beside it on the menu). See Chapter 4, *Getting to Know Crystal Reports*, for further information.

Format Bar

Use the FORMAT BAR command to toggle the format bar on and off. By default the format bar is on (a check mark appears beside it on the menu). See Chapter 4, *Getting to Know Crystal Reports*, for further information.

Status Bar

Use the STATUS BAR command to toggle the status bar on and off. By default the status bar is on (a check mark appears beside it on the menu).

Rulers in Design

Use the RULERS IN DESIGN command to toggle the rulers in the Design Tab on and off. By default the rulers in the Design Tab are on (a check mark appears beside it on the menu).

Rulers in Preview

Use the RULERS IN PREVIEW command to toggle the rulers in the Preview Tab on and off. By default the rulers in the Preview Tab are on (a check mark appears beside it on the menu).

Guidelines in Design

Use the **GUIDELINES IN DESIGN** command to toggle the guidelines in the Design Tab on and off. By default the guidelines in the Design Tab are on (a check mark appears beside it on the menu).

Guidelines in Preview

Use the **GUIDELINES IN PREVIEW** command to toggle the guidelines in the Preview Tab on and off. By default the guidelines in the Preview Tab are on (a check mark appears beside it on the menu).

Grid in Design

Use the **GRID IN DESIGN** command to toggle the grid lines in the Design Tab on and off. By default the grid in the Design Tab is on (a check mark appears beside it on the menu).

Grid in Preview

Use the **GRID IN PREVIEW** command to toggle the grid lines in the Preview Tab on and off. By default the grid lines in the Preview Tab are off (no check mark appears beside it on the menu).

Zoom



Use the **ZOOM** command to zoom in and out on your report. Using this command you can view your report from 25% to 400% of its actual size. At the lower magnifications you have more emphasis on layout; at the higher magnifications, you have more emphasis on details.

Close

Use the CLOSE command to close the active tab unless the active tab is the Design Tab of the primary report.

Close All

When you are creating a report and the program has created multiple tabs (for Preview, Subreport, Design, Subreport Preview, etc.), CLOSE ALL closes all open tabs except the Design Tab for the primary report.

INSERT MENU COMMANDS

Database Field



Use the DATABASE FIELD command to place data fields from an active database table into your report.

Text Object



Use the TEXT OBJECT command to insert a text object anywhere on your report. A text object is an object that holds text: a single character, a word, a paragraph, or an entire letter. But a text object can hold database objects and formula objects as well as text. Thus, a text object can be used for a variety of reporting needs such as labels, titles, annotations, callouts, disclaimers, footnotes, and even mail merge form letters.

NOTE: While the effective limit on text object size is 32K (including carriage return and line feed characters), it is recommended that text objects be used only for more manageable sized blocks of text.

Formula Field



Use the **FORMULA FIELD** command to create and/or insert a formula into your report. Using this command, you can insert an existing formula or create a new formula at the time you use the command.

The program also enables you to edit, delete, or rename existing formulas with this command.

Parameter Field



Use the **PARAMETER FIELD** command to create and/or insert a parameter (prompting) field into the report itself, into a formula, or into a record selection formula. When you use a parameter field, the program prompts you for a value when you first run the report and every time thereafter when you refresh the data.

The program also enables you to edit, delete, or rename existing parameter fields with this command.

Special Field

Use the **SPECIAL FIELD** command to insert any of five special fields anywhere you want them on your report.

Page Number Field

Use the **PAGE NUMBER FIELD** command to insert a field that prints the current page number. These fields are most often placed in the Page Header or Page Footer sections.

Total Page Count Field

Use the **TOTAL PAGE COUNT FIELD** to print the total number of pages in the report. You can use this field in a variety of instances. For example, you can use it in combination with the **PAGE NUMBER FIELD** command to create a text object that reads "Page x of y" where x is the Page Number Field and Y is the Total Page Count Field. These fields can be placed in any section of your report, depending on how often you want them to print.

Record Number Field

Use the RECORD NUMBER FIELD command to number each record that gets printed in the Details section of your report.

Group Number Field

Use the GROUP NUMBER FIELD command to number each group in your report. You can place this field in either the Group Header or Group Footer section.

Print Date Field

Use the PRINT DATE FIELD command to print the current date when the report prints. This fields can be placed in any section of your report, depending on how often you want them to print. The date can be changed using the SET PRINT DATE command on the Report menu (Page 563).

Print Time Field

Use the PRINT TIME FIELD command to print a field that contains the time when the report prints. This field can be placed in any section of your report, depending on how often you want them to print.

Data Date Field

Use the DATA DATE FIELD command to print a field that contains the date the data was last retrieved (refreshed).

Data Time Field

Use the DATA TIME FIELD command to print a field that contains the time the data was last refreshed.

Last Modification Date

Use the LAST MODIFICATION DATE command to print a field that contains the date the report was last modified. Modified refers to any modification (including something as simple as moving a field). When you modify the report and print it, the program prints the modification date even if you haven't saved the report before printing.

Last Modification Time

Use the LAST MODIFICATION TIME command to print a field that contains the time the report was last modified. This command works the same way as Last Modification Date.

Report Title

Use the REPORT TITLE command to print a field that contains the title you specified using Summary Info on the File menu.

Report Comments

Use the REPORT COMMENTS command to print a field that contains the comments you specified using Summary Info on the File menu.

Subtotal

Use the SUBTOTAL command to insert a subtotal in your report. A subtotal is a summary that can be used with numeric fields only. The program sorts, groups, and subtotals in a single step. Crystal Reports allows you to set the conditions under which a subtotal will print. When you choose the SUBTOTAL command, the Insert Subtotal dialog box appears.

Grand Total

Use the GRAND TOTAL command to print a report grand total (or a report average, a report count, etc.) at the end of the report. When you choose the GRAND TOTAL command, the Insert Grand Total dialog box appears.

Summary



Use the SUMMARY command to summarize data and print the summary in your report. The program sorts, groups, and summarizes in a single step. You can use summaries:

- to count the number of values in a group,
- to calculate the sum, average, standard deviation, or variance value in a group, and
- to identify the minimum or maximum value in a group.

This command is active only after you have selected a field to summarize. When you choose the SUMMARY command, the Insert Summary dialog box appears.

Group

Use the GROUP command to sort your data and break it into groups. This is a useful command, for example, for dividing a customer list into state or ZIP code groups. Crystal Reports allows you to set the conditions under which your data is grouped. When you choose the GROUP command, the Insert Group dialog box appears.

Section



Use the SECTION command to insert a new section in your report. This command takes you to the Section Expert, which can be used to insert a new section in your report.

Group Name Field



Use the GROUP NAME FIELD command to insert a field in your report that contains the name of the current group. The name is the field or formula value that triggers each grouping (that is, the value in the *sort and group by* field). When you insert this field in a group header or footer section, it becomes a "live" header for your groups.

A live group header is a header that changes dynamically with the contents of a group. If you group your data by state, for example, a typical live group header would print the name of the state at the beginning of each group.

By default, the program automatically inserts a group name field in the group header section whenever you create a group.

Cross-Tab



Use the CROSS-TAB command to insert a Cross-Tab object into your report. A Cross-Tab is a spreadsheet-like report used for identifying trends. You can create a report in which the Cross-Tab object is the entire report, or you can insert one or more Cross-Tab objects as subreports in a primary report.

Subreport



Use the SUBREPORT command to create a subreport object that you can place in your report. A subreport object is a self-contained report that has all of the same characteristics of any other report. You can import an existing report to use as a subreport object, or you can create your subreport from scratch as you would any other report.

You can insert multiple subreport objects in your parent report if you wish. You can coordinate subreport data with the data in the parent report, or you can leave your subreport freestanding, unrelated to any data in the parent report.

Line



Use the LINE command to draw horizontal and vertical lines anywhere on your report. You can use these lines to enclose field data or to create other graphic effects.

While you can create lines via the BORDER AND COLORS command (Page554), the lines you draw via the LINE command differ in several important ways:

- border lines cannot stand alone (they are always attached to a field object or a picture) whereas graphic lines can stand alone,
- a border line can be as high as the line, as high as the data, as wide as the field or as wide as the data while a graphic line can be any length or height you wish,
- graphic lines do not increase the height of a text line (making for fewer lines per inch) whereas lines created as borders do, and
- a border (field border) cannot span multiple sections whereas a graphic line can.

Box



Use the BOX command to draw freeform boxes anywhere on your report. You can use these boxes to enclose field data or to create other graphic effects.

While you can create boxes via the BORDER AND COLORS command (Page 554), the boxes you draw via the BOX command differ in several important ways:

- borders come in fixed sizes and shapes whereas boxes can be created in any size or shape you need,
- a border can be used to enclose individual field values whereas a box, placed around a field, encloses all the values in the field,
- borders cannot stand alone (they must surround a field or a picture) whereas boxes can,
- boxes do not increase the height of a text line (making for fewer lines per inch) whereas borders do, and
- a border (field border) cannot span multiple sections whereas a box can.

Picture



Use the PICTURE command to insert bit-mapped artwork (graphics, logos, etc.) in your report. Crystal Reports enables you to enhance the visual impact of your report using artwork in the following popular formats: *.bmp, *.gif, *.pcx, *.tif, *.tga.

Graph/Chart Expert



Use the GRAPH/CHART EXPERT command to insert a new graph/chart into your report. When you choose the GRAPH/CHART EXPERT command, the Graph/Chart Expert appears. Use the Expert for setting up your graph.

Object

The OBJECT command allows you to embed or link an OLE object in your report.

Crystal Reports is an OLE container application. As such, it can receive objects created in any object application that supports OLE. See Chapter 22, *OLE*, Page 527.

FORMAT MENU COMMANDS

Report Style Expert

Use the REPORT STYLE EXPERT command to apply one of several professionally designed styles to your report.

When you choose the REPORT STYLE EXPERT command, the Report Style Expert dialog box appears. You use this dialog box to select the style and apply it to your report.

When you select a report style, the preview box on the right side of the dialog box changes to show you what a report looks like formatted in the style you selected.

Simply select the report style and click *OK*. You can change the report style at any time.

Auto Arrange Report

Use the AUTO ARRANGE command to:

- adjust the spacing between fields,
- reposition the fields, and then
- center your report on the page, automatically.

This is a useful feature because when you place a field on your report, Crystal Reports allocates a space equal to the field width as specified in your database. Often that field width is far larger than the values that actually appear in each of the fields. For neat looking reports, it is often necessary to resize the fields so the space allotted more closely matches the size of the field values. Once you have resized the fields you often need to reposition them for proper balance.

Format (Object)

Use Format (Object) to format a selected object. When you choose this command, the program takes you to the Format Editor which presents you with various formatting properties that you can

apply to the object. Those properties are different for different kinds of objects, but they can include such things as applying borders and colors, changing the font for text and numbers, suppressing the object so it doesn't print, cropping the size of the object, or keeping an object one size or letting it grow.

The name of this command changes to match the object you have selected (FORMAT FIELD, FORMAT BOX, FORMAT GRAPHIC, etc.).

Border and Colors

Use the BORDER AND COLORS command to set up borders, background fill, and drop shadows for fields on your report. You can also use it to customize the border for a selected field. For example:

- you can enclose the field in a box,
- you can highlight the field with any portion of a box you want (only the top, only the left side, the top and bottom together with no sides, etc.),
- you can add drop shadows that print below and to the right of the field value,
- you can change the color of the border, text, and/or add a background fill color to a field, and
- you can specify the border width (the full width allotted for the field or the width of the data only), and the border height (the full height allotted for the field or the height of the data only).

Using the border options, you can create a variety of striking effects that can enhance the look of your report and highlight important data.

Font

Use the FONT command to change the font, font size, style, effects, and/or color for selected text and numbers on your report. When you choose the FONT command the Text Format dialog box appears with the Font Tab active.

NOTE: To use this command with a text object, you have to double-click the object. When you click a text object, you select it for resizing or moving. When you double-click the object, you activate the editing capabilities within the text object.

Paragraph

Use PARAGRAPH to format paragraphs in a text object. For each paragraph you can set the font, tabs, and indents. In order to activate this command you have to double-click the text object.

NOTE: When you click a text object, you select it for resizing or moving. When you double-click the object, you activate the editing capabilities within the text object.

Tabs

Use the TABS command to set the tab stops for paragraphs in a text object. You can specify tab stop position and tab alignment. In order to activate this command you have to double-click the text object.

NOTE: When you click a text object, you select it for resizing or moving. When you double-click the object, you activate the editing capabilities within the text object.

Graph/Chart

Use Graph/Chart when you want to return to the Graph/Chart Expert to change the format of your Graph/Chart.

Cross-Tab

Use the CROSS-TAB command when you want to return to the Cross-Tab dialog box to change the format of your Cross-Tab.

Section

Use the SECTION command to insert, delete, merge, and move report sections and to set a variety of properties for individual sections. Using this command you can:

- hide a section (keep it from printing while retaining drilldown capabilities)
- suppress a section (keep it from printing and eliminate drilldown capabilities),
- print subtotals or group values only at the bottom of the page,
- insert a page break before the section is printed,
- insert a page break after the section is printed,
- reset the page number to one (1) after a group value prints,
- prevent page breaks from spreading data from a single record over two pages,
- prevent blank sections from printing,
- cause an object to underlay the following section(s) when it prints,
- format the section with multi-column capabilities.

Move Backward

When you have multiple objects in a stack, use the MOVE BACKWARD command to move the selected object backward (down) one layer in the stack. In order to access an object you have moved backward, you can do one of the following:

- move objects in front (on top) of the object out of the way, or
- move the other objects behind it using this command.

Move To Back

When you have multiple objects in a stack, use the MOVE TO BACK command to move the selected object to the back (bottom) of the stack.

Move Forward

When you have multiple objects in a stack, use the **MOVE FORWARD** command to move the selected object forward (up) one layer in the stack.

Move To Front

When you have multiple objects in a stack, use the **MOVE TO FRONT** command to move the selected object to the front (top) of the stack.

DATABASE MENU COMMANDS

Visual Linking Expert



Use the **VISUAL LINKING EXPERT** command to:

- create Smart Links automatically between databases (in many cases),
- display the links that have been set up among active databases,
- create new links between active databases,
- update (modify) existing links,
- delete existing links,
- add new databases/tables to the report,
- select the index to use when multiple indexes are available for a field,
- set the way partial text matches are to be handled,
- specify lookup options when database (A) is linked to two others (B) and (C), and
- specify SQL join types.

When you choose the Visual Linking Expert command, the Visual Linking Expert appears.

Add Database to Report

Use the ADD DATABASE TO REPORT command to select additional database table(s) for use in your report. When creating a new report, you select the first database you want to use in your report from the Choose Database File dialog box that appears when you begin a new report. You specify additional database tables with this command.

Remove from Report

Use the REMOVE FROM REPORT command to delete database tables from the active list so they can no longer be used in your report.

Set Location

Use the SET LOCATION command to change the location of a database that is active in a report. This option is convenient if you need to change the directory or disk location of a direct access (PC) database to avoid file name conflicts, better utilize disk space, etc. It is also a handy option to use if someone sends you a report based on a direct access database that is located in different disk / directory location on their system than on yours.

The SET LOCATION command can also be used when the name of an ODBC data source has changed or when you need to change the ODBC data source that is accessed by your report. See *How to change the ODBC data source accessed by a report*, Page 461.

NOTE: This option does not physically move the database(s). It simply directs Crystal Reports to look for the database(s) in a different location than you specified when setting up the report.

Set Alias

Use the SET ALIAS command to change the alias you are using for one or more of your active databases. In Crystal Reports, an alias is an alternative name assigned to a database. Aliases make it easier for you to use a report created with a database whose name and / or location has changed since the report was created.

Verify Database

Use the `VERIFY DATABASE` command to make certain your report prints with the current version of the active database and resets its record buffers to the current record size.

When you first create a report, the report draws its fields from the database as it exists at that time. It uses the structure of the database (number of fields, field position, data type, etc.) to identify and select those fields you want to appear on the report. If you change the structure of the database (by adding or deleting fields) after you create the report, the program needs to adapt the report to the new structure. The `VERIFY DATABASE` command is the tool you use for adapting the report to the new database structure.

- If the current version of the database has more fields than it had when the report was first created, Crystal Reports attempts to identify and use the correct fields from the new database. The aim is to print an unchanged version of the report even though the underlying database has changed.
- If the current version of the database has fewer fields than it had when the report was first created, it uses those fields that are still available when it prints the report and ignores those that are no longer available.

Verify on Every Print

The `VERIFY ON EVERY PRINT` command is a lock that triggers the `VERIFY DATABASE` command every time you print.

- If there is a check mark beside the `VERIFY ON EVERY PRINT` command, the command is active (it will trigger the `VERIFY DATABASE` command every time you print).
- If there is no check mark beside it, the command is inactive. By default the command is inactive.

Log On Server

Use the LOG ON SERVER command to log on to a SQL or ODBC data source server. When you choose the LOG ON SERVER command the Log On Server dialog box appears.

Log Off Server

Use the LOG OFF SERVER command to log off an active SQL or ODBC data source.

Show SQL Query

Use the SHOW SQL QUERY command to view and to edit the SQL query Crystal Reports is sending to your SQL data source.

Stored Procedure Parameters

Use the STORED PROCEDURE PARAMETERS command if you want to change the parameters for the stored procedure you are using in your report.

Crystal Reports enables you to build reports using stored procedures from SQL data sources that support them. The program assumes that the stored procedure used for creating a report will generate a result set; it makes that result set available as a table. If the stored procedure requires parameters, those parameters are stored with the report.

REPORT MENU COMMANDS

Report Expert

Use ReportExpert to return to the Expert you used for creating the active report so you can modify the report. If you did not use an Expert for creating the report and you choose this command, the program brings up the Standard Report Expert.

Select Expert



The SELECT EXPERT command enables you to select the records or groups you want to include in your report and filter out those you do not want. For example, you may have a customer database that contains records for customers from every state and Canadian province but you want to do a report only on Texas customers. The Select Expert enables you to restrict your report so that only Texas customers are included.

When you choose the SELECT EXPERT command, the Select Expert appears. Crystal Reports automatically generates a record or group selection formula based on settings you specify in the dialog box.

NOTE: This command allows you to set up reasonably complex selection criteria, but to have more control over the formula that is generated, use EDIT SELECTION FORMULA on the Report menu and set up your selection formula using the Record or Group Selection Formula Editor.

Edit Selection Formula

Use the EDIT SELECTION FORMULA command to create a record or group selection formula using the Crystal Reports formula language. You use the formula to select the records or groups that you want included in your report (if you do not want them all included). You can also use this command to edit an existing selection formula.

Record

Use the RECORD command to create a record selection formula or to edit an existing record selection formula. When you choose this command, the Record Selection Formula Editor appears where you can build a Boolean selection formula.

Group

Use the GROUP command to create a group selection formula or to edit an existing group selection formula. When you choose this command, the Group Selection Formula Editor appears where you can build a Boolean selection formula.

Change Group Expert

Use the CHANGE GROUP EXPERT command to change the sorting and grouping specifications (sort and group by field, sort direction, etc.) for any of the groups on your report. You can also use this command to change the order in which the groups are processed. If you insert a group section, a subtotal, or a summary and later want to modify it in any way, use this command to do so.

Top N/Sort Group Expert

Use the TOP N/SORT GROUP EXPERT command to set up reports that print only the Top N or Bottom N groups.

Before you can use the Top N/Sort Expert dialog box, your data must be subtotaled or summarized.

Sort Records



Use the SORT RECORDS command to change the order in which records appear in your report. You can add and remove sort fields and define the sort direction (ascending or descending) for the data in your report.

When you choose the SORT RECORDS command, the Record Sort Order dialog box appears.

NOTE: *You can not remove group sorts using this command.*

Refresh Report Data



Use the REFRESH REPORT DATA command to retrieve fresh data for your report display the report in the Preview Tab.

By design, Crystal Reports only retrieves data when necessary. When you preview your report in the Preview Tab, print the report, or export it to a file, the program runs your report and retrieves the required data. Once you are working with the data in the Preview Tab, the program reruns the report and retrieves the data again under the following conditions:

- if you add fields to the report after the report was run,
- if you add a formula that references a field that was not in the report when you ran it the first time,
- if you expand your record selection criteria to include more records than were needed when you first ran the report, or
- if you toggle the *Refresh data on every print* option on.

Aside from these circumstances, the program will not automatically rerun a report.

There may be times, however, when you want to retrieve new data for the report. For example, you may have created a report several days earlier and want to see it now with the most current data. Refresh Report Data is intended for times such as this.

Set Print Date

Use the SET PRINT DATE command if your report contains a Print Date field and you want to change the date that appears in this field. This allows you, for example, to post-date a report that you need to print prior to the date it will be reviewed in a meeting or conference.

Compile Report

Use the COMPILE REPORT command to create an executable version of your report. An executable version is a report that can be printed on demand (to printer, window, or file) by simply clicking an icon without the need to first open Crystal Reports.

This command enables you to create a report and share it with others, even if the others do not know how to use Crystal Reports. In fact, you can create a report that others can use, even if they do not have Crystal Reports on their system. This is a useful command for MIS departments who want to create reports that can be used company wide, for individuals who want to send copies of reports to others who do not have Crystal Reports, or for anybody who needs to create a report that appears at the click of a mouse.

You can print a compiled report to a window, a printer, or a file, just like any other report. Additionally, for maximum flexibility, you can set up the program to print the report immediately (on demand) or at a later time (i.e., when printers are more readily available). If you select a later time, Crystal Reports monitors your computer's internal clock and prints the report at the time specified.

Finally, if your report includes a record selection formula, the compiled version of the report gives you a chance to change certain values in the selection formula before you print the report. With a carefully constructed selection formula, many people in an organization will be able to tailor a single compiled report to their own individual needs.

Report Distribution Expert

When you choose the REPORT DISTRIBUTION EXPERT command, the Report Distribution Expert dialog box appears. Each report that you distribute with an application or as a compiled report requires a number of files to make it run properly. The Report Distribution Expert helps you identify and gather those files.

WINDOW MENU COMMANDS

Tile Vertically

Use the TILE VERTICALLY command to display your Crystal Reports windows side by side on screen. With the TILE VERTICALLY command, windows may be resized so they all fit on screen; all windows are visible.

Tile Horizontally

Use the TILE HORIZONTALLY command to display your Crystal Reports windows one above the other on screen. With the Tile HORIZONTALLY command, windows may be resized so they all fit on screen; all windows are visible.

Cascade

Use the **CASCADE** command to stack and overlap your Crystal Reports windows. With the **CASCADE** command, the entire top window is visible but only the title bars of the remaining windows are visible.

Arrange Icons

If you have minimized any document windows, use the **ARRANGE ICONS** command to arrange the document icons neatly at the bottom of the Crystal Reports window.

Close All command

Use the **CLOSE ALL** command to close all open windows. Crystal Reports gives you an opportunity to save any work in each window that you have not already saved. When you choose the **CLOSE ALL** command:

- If you have saved all your reports, Crystal Reports closes all the windows.
- If you have not saved all your reports, Crystal Reports prompts you to save them, report by report.
 - Select *Yes* to save the report before closing the window.
 - Select *No* to close the window without saving the report.

HELP MENU COMMANDS

Contents

Use the **CONTENTS** command to call up the Crystal Reports Online Help Index. Using this index as a starting point, you can quickly find any help topic of interest. Once in online Help, you can always return to the main index by clicking the *Contents* button in the upper left corner of the Help window.

Context Help



Use the CONTEXT HELP command to activate a context pointer that you can use to identify elements of the interface that you need help with.

Search

The SEARCH command displays the Search dialog box for Crystal Reports online Help. You can search through every subject in online Help to find the information you need quickly. Jump straight to the topic of interest from this dialog box.

Using Help

Use the USING HELP command if you are not familiar with the Windows Help system. A separate window will appear explaining how to use the Help system. Follow the instructions to learn more about Windows Help systems and how to get around in Crystal Reports online Help.

Register/Change Address

When you choose the REGISTER/CHANGE ADDRESS COMMAND, Crystal Reports displays an electronic form that you can use to register and /or to notify the company of a change of address. You can then print this form and mail or fax it in. You can also transmit the information to the company via modem. Instructions for sending the form by mail or fax are printed on the form.

NOTE: You must send in the registration form in order to receive technical support on your product.

Technical Support Request

When you choose the TECHNICAL SUPPORT REQUEST command, Crystal Reports displays an electronic form that you can use to enter a technical support request. Once you are finished, you can

send the request to the company via modem using the built in communications program. You will receive your response back in the same manner when you check the support queue for responses.

NOTE: You must send in the registration form in order to receive technical support on your product.

About More Crystal Products

Use the ABOUT MORE CRYSTAL PRODUCTS command to find information on our other products such as Crystal Info.

About Crystal Reports

Use the ABOUT CRYSTAL REPORTS command to find the version number and other pertinent information about Crystal Reports.

24

Crystal Reports Functions

What you will find in this chapter...

Introduction 570

Functions are listed in alphabetical order. The references below show what page each letter begins on.

A 570

B 573

C 573

D 576

G 580

H 580

I 580

L 581

M 584

N 587

O 590

P 590

R 593

S 595

T 600

U 605

V 605

W 606

Y 607

Introduction

This chapter provides an in-depth look at all Crystal Reports functions.

Functions are built-in procedures or subroutines used to evaluate, make calculations on, or transform data. When you specify a function, Crystal Reports performs the set of operations built into the function without you having to specify each operation separately. In this way, a function is a kind of shorthand that makes it easier and less time consuming for you to create formulas. Simple, straightforward examples are provided for each function. In addition, for many functions, references are provided to formulas in the Formulas in Action topic in Crystal Reports online Help (search for *Formulas in Action*). Formulas in Action shows advanced uses of the functions to solve complex, real-world reporting problems.

Abs (x)

Arguments

x is any number.

Returns

Absolute value of *x*.

Action

Abs returns the absolute value of *x*, eliminating any negative value.

Aged0To30Days

Aged31To60Days

Aged61To90Days

Returns

None

Action

Specifies a range of date values that fall within a certain time period before the present date. If the current date is 12/30/98,

Aged0To30Days specifies the period from 12/1/98 to the present date, Aged31To60Days specifies the period from 11/1/98 to 11/30/98, and Aged61To90Days specifies the period 10/2/98 to 10/31/98.

AllDatesFromToday

Returns None

Action Specifies a range of date values that includes any date from the present day to any future date value that may appear in a field. AllDatesFromToday includes the present day.

AllDatesFromTomorrow

Returns None

Action Specifies a range of date values that fall after the present day. AllDatesFromTomorrow does not include the present day, but does include any future date.

AllDatesToToday

Returns None

Action Specifies a range of date values that includes every day up through the present day. AllDatesToToday includes the present day.

AllDatesToYesterday

Returns

None

Action

Specifies a range of date values that includes every day up through the previous day. AllDatesToYesterday includes all dates before today, but does not include the present day.

Asc (str)

Arguments

str is a text string.

Returns

Whole Number (the ASCII value for the first character in the string)

Action

The Asc (str) function returns the ASCII code of the first character in the string.

Average (fld)

Average (fld, condFld)

Average (fld, condFld, cond)

Average (x)

Action

Crystal Reports enables you to average the values that appear in your report. For example:

- If a sales report includes a field that shows the amount of each order, you can compute the average of all the orders that appear on the report (a grand total average). See *SummaryFunction (fld)*, Page 598.

- If you break orders into groups (for example, orders grouped by the state that they come from), you can compute the average order per group (in this case, per state). See *SummaryFunction (fld, condFld)*, Page 598.
- If you break orders into date or Boolean groups (for example, orders grouped by the month in which they were placed), you can compute the average order per group based on a particular change in the date or Boolean field (in this case, per month). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify an array of individual values, you can also compute the average value in the set. See *SummaryFunction (x)*, Page 599.

BeforeReadingRecords

Returns

None

Action

Specifies that the formula is to be evaluated before the database records are read.

Calendar1stHalf Calendar2ndHalf

Returns

None

Action

Specifies a range of date values that includes all dates that fall within the first or second half of the calendar year, respectively. The first half of the calendar year includes all dates from January 1st through June 30th. The second half of the calendar year includes all dates from July 1st through December 31st.

Calendar1stQtr, Calendar2ndQtr, Calendar3rdQtr, Calendar4thQtr

Returns

None

Action

Specifies a range of date values that falls within the 1st, 2nd, 3rd, or 4th quarter of the calendar year. The first quarter of the calendar year includes all dates from January 1st through March 31st. The second quarter of the calendar year includes all dates from April 1st through June 30th. The third quarter of the calendar year includes all dates from July 1st through September 30th. The fourth quarter of the calendar year includes all dates from October 1st through December 31st.

Chr (x)

Arguments

x is a whole number, specifically, it is any ASCII value.

Returns

Text String (one character long)

Action

The Chr (x) function returns the single character text string associated with the ASCII value passed as *x*.

Count (fld), Count (fld, condFld), Count (fld, condFld, cond), Count (x)

Action

Crystal Reports enables you to count the values that appear in your report. For example:

- If a sales report includes all orders made and the amount of each order, you can compute the total number of orders that appear on the report (a grand total count). See *SummaryFunction (fld)*, Page 598.

- If you break orders into groups (for example, orders grouped by the state that they come from), you can compute the number of orders per group (in this case, per state). See *SummaryFunction (fld, condFld)*, Page 598.
- If you break orders into date or Boolean groups (for example, orders grouped by the month in which they were placed), you can compute the number of orders per group based on a particular change in the date or Boolean field (in this case, per month). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can compute the number of values in the set. See *SummaryFunction (x)*, Page 599.

CurrentDate

Returns Date Value

Action CurrentDate returns the current date on a report. The date is taken from your computer's internal clock.

CurrentDateTime

Returns dateTime Value

Action CurrentDateTime returns the current date and time on a report. The date and time is taken from your computer's internal clock.

CurrentTime

Returns Time Value

Action CurrentTime returns the current time on a report. The time is taken from your computer's internal clock.

Date (dateTime), Date (year, month, day)

Arguments

Date(dateTime)

— *dateTime* is a date/time value.

Date(year, month, day)

— *year* is a whole number representing a year, e.g., 1996.

— *month* is a whole number representing a month, e.g., 12 for December.

— *day* is a whole number representing a day of the month, e.g., 05.

Returns

Date Value

Action

Date returns a date value based on either a date/time value or the individual elements of a date provided: year, month, and day. If the individual elements of the date are provided:

- The year must be four digits.
- The month must be a number from 1 to 12.
- The day must be a number from 1 to 31.

DateTime (date, time), DateTime (year, month, day), DateTime (year, month, day, hour, min, sec)

Arguments

DateTime(date, time)

— *date* is a date value.

— *time* is a time value.

DateTime(year, month, day)

DateTime(year, month, day, hour, min, sec)

— *year* is a whole number representing a calendar year, e.g., 1996.

- *month* is a whole number representing a month, e.g., 1 for January.
- *day* is a whole number representing a day of the month, e.g., 10.
- *hour* is a whole number representing an hour of the day, e.g., 12.
- *min* is a whole number representing a minute, e.g., 59.
- *sec* is a whole number representing seconds, e.g., 30.

Returns dateTime Value

Action The DateTime function creates a date/time value based on either a date and time value or separate whole numbers indicating the individual parts of a calendar date and time: year, month, day, hour, minute, and second.

Day (x)

Arguments *x* is a date value or a date/time value.

Returns Whole Number (the day of the month)

Action Day extracts the day from a date or date/time value and returns a whole number.

DayOfWeek (x)

Arguments *x* is a date value or dateTime value.

Returns Whole Number

Action

DayOfWeek extracts the day component of a date, determines the day of the week the date falls on, and converts the day of the week to a number (1 to 7) where Sunday is the first day of the week.

DistinctCount (fld), DistinctCount (fld, condFld), DistinctCount (fld, condFld, cond), DistinctCount (x)

Action

Crystal Reports enables you to get a distinct count of the values that appear in your report. For example:

- If a sales report includes all orders made by customers, you can compute the total number of distinct customers that appear in the report (a grand total distinct count), excluding any duplicate records. If a customer made more than one order, the duplicate occurrences of that customer are ignored. See *SummaryFunction (fld)*, Page 597.
- If you break orders into groups (for example, orders grouped by the state that they come from), you can compute the number of distinct customers per group (in this case, per state). Any customers that made more than one order and appear more than once in a group are only counted once. See *SummaryFunction (fld, condFld)*, Page 598.
- If you break orders into date or Boolean groups (for example, orders grouped by the month in which they were placed), you can compute the number of distinct customers in each group based on a particular change in the date or Boolean field (in this case, the number of customers that placed orders each month). If a customer placed more than one order within the month, duplicate instances of that customer are ignored. See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can compute the number of distinct values in the set. Duplicate values in the set are only counted once. See *SummaryFunction (x)*, Page 599.

DTSToDate (dateTime)

Arguments *dateTime* is a date / time value.

Returns Date Value

Action Evaluates the date / time value specified and returns only the date.

DTSToSeconds (dateTime)

Arguments *dateTime* is a date / time value.

Returns Whole Number

Action Evaluates the date / time value specified and converts the time to the number of seconds from 00:00:00 (12:00 midnight) to the specified time.

DTSTime (dateTime)

Arguments *dateTime* is a date / time value.

Returns Time Value

Action Evaluates the date / time value specified and returns only the time in military format (00:00:00).

GroupNumber

Returns Whole Number

Action Returns the current group number.

Hour (x)

Arguments *x* is a time value or dateTime value.

Returns Whole Number

Action The Hour function extracts the hour portion of a time value and returns it as a whole number.

InRepeatedGroupHeader

Returns Boolean

Action Returns TRUE when a group header section is repeated on a second, third, etc. page.

InStr (str1, str2)

InStr (start, str1, str2)

Arguments

- *start* is the character in *str1* where the search is to begin. This is a 1 based index. (This argument is optional.)
- *str1* is the text string to be searched.

- *str2* is the text string being sought.

Returns Whole Number

Action The InStr function returns the position of the first occurrence of one string within another. This position is a 1 based index of the characters in *str1*. If *str2* is not found in *str1*, the InStr function returns 0. The start argument sets the starting position for the search.

IsNull (fld)

Arguments *fld* is any valid database field.

Returns Boolean Value

Action Evaluates the field specified in the current record and returns TRUE if the field contains a null value.

Last4WeeksToSun

Returns None

Action Specifies a range of dates that includes the four weeks previous to last Sunday. A week begins on a Monday and ends on a Sunday. For example, September, 1996, begins on a Sunday.

- If today is September 22 (Sunday), Last4WeeksToSun begins on August 26 (Monday) and ends on September 22 (today, Sunday).
- If today is September 28 (Saturday), Last4WeeksToSun begins on August 26 (Monday) and ends on September 22 (the previous Sunday).

Last7Days

Returns None

Action Specifies a range of date values that includes all dates from seven days ago to today (including today).

LastFullMonth

Returns None

Action Specifies a range of date values that includes all dates from the first to last day of the previous month.

LastFullWeek

Returns None

Action Specifies a range of date values that includes all dates from Sunday to Saturday of the previous week.

LastYearMTD

Returns None

Action Specifies a range of date values in the previous year that matches the current month to date.

LastYearYTD

Returns	None
Action	Specifies a range of date values that includes all dates in the last year, up to the current date last year.

Left (str, length)

Arguments	<ul style="list-style-type: none">• <i>str</i> is a text string value.• <i>length</i> is a whole number indicating the number of characters to be extracted from <i>str</i>.
------------------	---

Returns	Text String
Action	Extracts the specified number of characters from the left side of the string.

Length(str)

Arguments	<i>str</i> is a text string or a field containing a text string.
Returns	Whole Number
Action	Length returns the number of characters in the specified text string. NOTE: Crystal Reports includes any blank spaces as part of the character count. NOTE: Text strings must be enclosed in either double ("") or single (') quotation marks.

LooksLike (string, mask)

Arguments

- *string* is the text string or field containing text string values that are being compared to the mask.
- *mask* is a text string that provides a mask for comparing the value in the string argument.

Returns

Boolean Value

Action

Enables you to locate field values using standard DOS wildcards (? = wildcard for single character, * = wildcard for any number of characters). It does this by comparing a string to a mask that contains one or more wildcards. The function returns True if the string matches the mask, False if the string does not match the mask.

LowerCase(x)

Arguments

x is a text string to be converted to all lower case characters.

Returns

Text String

Action

LowerCase prints the text string or text string value in the data field in all lower case letters.

Maximum (fld), Maximum (fld, condFld), Maximum (fld, condFld, cond), Maximum (x)

Action

Crystal Reports enables you to find the maximum value that appears in a set of values. For example:

- If a sales report includes a field that shows the amount of each order, you can find the highest order amount of all the orders that appear on the report (a grand total maximum amount). See *SummaryFunction (fld)*, Page 598.
- If you break orders into groups (for example, orders grouped by the state that they come from), you can find the highest order amount per group (in this case, per state). See *SummaryFunction (fld, condFld)*, Page 598.
- If you break orders into date or Boolean groups (for example, orders grouped by the month in which they were placed), you can find the highest order amount per group based on a specific change in the date or Boolean field (in this case, per month). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can find the highest value in the set. See *SummaryFunction (x)*, Page 599.

Mid (str, start), Mid (str, start, length)

Arguments

- *str* is the text string from which one or more characters is being extracted.
- *start* is a whole number indicating the position of the first character to extract.
- *length* is a whole number indicating the number of characters to be extracted from the text string. This argument is optional. If a value is not specified, the rest of the string from the starting position is extracted.

Returns

Text String

Action

The Mid function returns a specified number of characters from a string. The second argument is the character position where the part to be taken begins. The third argument is the length of the string you want to be taken out. If the third argument is not specified, everything from the start position to the end of the string is extracted.

Minimum (fld), Minimum (fld, condFld), Minimum (fld, condFld, cond), Minimum (x)

Action

Crystal Reports enables you to find the minimum value that appears in a set of values. For example:

- If a sales report includes a field that shows the amount of each order, you can find the lowest order amount of all the orders that appear on the report (a grand total minimum amount). See *SummaryFunction (fld)*, Page 598.
- If you break orders into groups (for example, orders grouped by the state that they come from), you can find the lowest order amount per group (in this case, per state). See *SummaryFunction (fld, condFld)*, Page 598.
- If you break orders into date or Boolean groups (for example, orders grouped by the month in which they were placed), you can find the lowest order amount per group based on a specific change in the date or Boolean field (in this case, per month). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can find the lowest value in the set. See *SummaryFunction (x)*, Page 599.

Minute(x)

Arguments

x is a time value.

Returns

Whole Number

Action

Retrieves just the minutes from a time value.

Month(x)

Arguments *x* is a date value or a dateTime value.

Returns Whole Number

Action Month extracts the month component of a date or dateTime value and converts it to a number.

MonthToDate

Returns None

Action Specifies a range of date values that includes all dates from the first day of the month to today.

Next(fld)

Arguments *fld* is any valid database or formula field.

Returns A field value of the same type as the argument.

Action Next(*fld*) returns the value of the specified field for the next record.

Next30Days, Next31To60Days, Next61To90Days, Next91To365Days

Returns None

Action Specifies a range of date values that includes all dates in the period specified starting from today (including today).

NextIsNull(fld)

Arguments *fld* is any valid database or formula field.

Returns Boolean Value

Action Evaluates the field specified in the next record and returns a TRUE if the field contains a null value.

NumberToCode39 (number, #places)

Arguments

- *number* is a fractional value to be converted into a bar code.
- *#places* is a whole number value that indicates how many decimal places in number are to be reserved when number is converted.

Returns A string containing a valid bar code value.

Action Converts numbers in number fields to the format needed to create bar codes. It converts the numbers to the characters supported by the font and formats the resulting value with leading and trailing asterisks.

NumberToPostnet (number)

- Arguments** *number* is a zip code number to be converted into a US mail zip code bar code.
- Returns** A valid US mail zip code bar code value.
- Action** Converts zip codes to the format needed to create zip code bar codes. It converts the numbers to the characters supported by the font and formats the resulting value for zip code bar codes.

NumericText(str)

- Arguments** *str* is a text string being tested for numeric text.
- Returns** Boolean Value
- Action** NumericText tests to see if the content of a text object is a number.
- If the entire content of the field is a number or if the characters extracted via the subscript operator are entirely a number the expression returns a TRUE value.
 - If any part of the content of the field or of the characters extracted are not a number, the expression returns a FALSE value.

OnFirstRecord

- Returns** Boolean
- Action** Returns TRUE when the current record being evaluated is the first record in the report.

OnLastRecord

Returns	Boolean
Action	Returns TRUE when the current record being evaluated is the last record in the report.

Over90Days

Returns	None
Action	Specifies a range of date values that includes all values that are more than 90 days older than the current date.

PageNumber

Returns	Whole Number
Action	PageNumber inserts the current page number as a field in a formula.

Picture (string, picture)

Arguments	<ul style="list-style-type: none">• <i>string</i> is a text string to be formatted according to the picture argument.• <i>picture</i> is a text string representing the format for the text string in the string argument.
Returns	String

Action

Picture (string, picture) prints a text string or values in a text string field in a predetermined format.

PopulationStdDev (fld), PopulationStdDev (fld, condFld) PopulationStdDev (fld, condFld, cond), PopulationStdDev (x)

Action

Crystal Reports enables you to find the population standard deviation of a set of values in your report. For example:

- You can calculate the grand total population standard deviation for all values in a field. See *SummaryFunction (fld)*, Page 598.
- You can calculate the population standard deviation for all values within a group (for example, sales grouped by the state that they come from). See *SummaryFunction (fld, condFld)*, Page 598.
- You can calculate the population standard deviation for all values within a group in which grouping is controlled by changes in a date or Boolean field (for example, sales grouped by the month in which they were made). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can compute the population standard deviation of the values in the set. See *SummaryFunction (x)*, Page 599.

PopulationVariance (fld), PopulationVariance (fld, condFld) PopulationVariance (fld, condFld, cond), PopulationVariance (x)

Action

Crystal Reports enables you to find the population variance of a set of values in your report. For example:

- You can calculate the grand total population variance for all values in a field. See *SummaryFunction (fld)*, Page 598.
- You can calculate the population variance for all values within a group (for example, sales grouped by the state that they come from). See *SummaryFunction (fld, condFld)*, Page 598.

- You can calculate the population variance for all values within a group in which grouping is controlled by changes in a date or Boolean field (for example, sales grouped by the month in which they were made). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can compute the population variance of the values in the set. See *SummaryFunction (x)*, Page 599.

Previous(fld)

Arguments	<i>fld</i> is any valid database or formula field.
Returns	A field value of the same type as that passed as the <i>fld</i> argument.
Action	Previous(<i>fld</i>) returns the value of the specified field in the previous record.

PreviousIsNull(fld)

Arguments	<i>fld</i> is any valid database or formula field.
Returns	Boolean Value
Action	Evaluates the field specified in the previous record and returns a TRUE if the field contains a null value.

PrintDate

Returns	Date Value
----------------	------------

Action PrintDate inserts the date the report is printed as a field in a formula.

RecordNumber

Returns Whole Number

Action RecordNumber returns the current record number.

Remainder (num, denom)

Arguments

- *num* is a fractional value.
- *denom* is a fractional value.

Returns Fractional Value

Action Remainder returns the remainder after the numerator (dividend) has been divided by the denominator (divisor). In a typical division situation, Crystal Reports expresses a quotient as a whole number (if any) and up to six decimal places. When using Remainder, however, Crystal Reports performs the division internally, determines the whole number quotient and the remainder, and returns only the remainder.

ReplicateString (str, #copies)

Arguments

- *str* is the text string to be replicated.
- *#copies* is a whole number indicating the number of times *str* is to be replicated.

Returns Text String

Action Replicates the string in `str` the number of times specified by `#copies`.

Right (`str`, `length`)

Arguments

- `str` is a text string.
- `length` is a whole number indicating the number of characters to be extracted from `str`.

Returns Text String

Action Extracts the given number of text characters from the right side of the specified string.

Round(`x`), Round(`x`, `#places`)

Arguments

- `x` is a fractional value to be rounded off.
- `#places` is a whole number indicating the number of decimal places `x` is to be rounded off to.

Returns Fractional Number

Action Round rounds to the nearest whole number if the `#places` argument is excluded. If the value to the right of the decimal point is .499 or below, Crystal Reports rounds to the next lowest number. If the value to the right of the decimal point is .5 or above, Crystal Reports rounds to the next highest number.

If the `#places` argument is used, the value in `x` is rounded to the nearest decimal place indicated by `#places`. Specifying `#places` as 0 works identical to leaving the `#places` argument off. If `#places` is negative, the number is rounded to the nearest tens, hundreds, and so on.

Second(x)

Arguments *x* is a time value or dateTime value.

Returns Whole Number

Action Extracts the seconds portion of a specified time value.

Soundex (string)

Arguments *string* is a text string.

Returns Text String

Action Soundex evaluates a text string and returns a four character value that symbolizes the way the string sounds.

Space(x)

Arguments *x* is a whole number indicating the number of spaces.

Returns Text String (one or more spaces)

Action The Space function returns a specified number of spaces.

StdDev (fld), StdDev (fld, condFld) StdDev (fld, condFld, cond), StdDev (x)

Action

Crystal Reports enables you to find the standard deviation of a set of values in your report. For example:

- You can calculate the grand total standard deviation for all values in a field. See *SummaryFunction (fld)*, Page 598.
- You can calculate the standard deviation for all values within a group (for example, sales grouped by the state that they come from). See *SummaryFunction (fld, condFld)*, Page 598.
- You can calculate the standard deviation for all values within a group in which grouping is controlled by changes in a date or Boolean field (for example, sales grouped by the month in which they were made). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can compute the standard deviation of the values in the set. See *SummaryFunction (x)*, Page 599.

StrCmp(str1, str2) StrCmp(str1, str2, compare)

Arguments

- *str1* is the first text string to be compared.
- *str2* is the second text string to be compared.
- *compare* is a whole number, either 0 or 1. By default, it is 0. Please see the Comments section below for more information on using the compare argument. (This argument is optional.)

Returns

Whole Number where:

- -1 = Less Than
- 0 = Equal To
- 1 = Greater Than

Action The StrCmp function compares two strings. If the third argument is 0, a binary comparison is done. If the third argument is 1, a textual comparison. By default it is 0.

StringToCode39 (string)

Arguments *string* is a text string containing a numerical string.

Returns A valid value for creating bar code fonts.

Action Converts numbers in string fields to the format needed to create bar codes. It converts the numbers to the characters supported by the font and formats the resulting value with leading and trailing asterisks.

StringToPostnet (string)

Arguments *string* is a zip code stored as a text string.

Returns A valid US mail zip code bar code value.

Action Converts zip codes to the format needed to create zip code bar codes. It converts the text strings to the characters supported by the font and formats the resulting value for zip code bar codes.

Sum (fld), Sum (fld, condFld) Sum (fld, condFld, cond), Sum (x)

Action Crystal Reports enables you to add the values that appear in your report. For example:

- If a sales report includes a field that shows the amount of each order, you can compute the sum of all the orders that appear

on the report (a grand total sum). See *SummaryFunction (fld)*, Page 598.

- If you break orders into groups (for example, orders grouped by the state that they come from), you can compute the sum of the order amounts per group (in this case, per state). See *SummaryFunction (fld, condFld)*, Page 598.
- If you break orders into date or Boolean groups (for example, orders grouped by the month in which they were placed), you can compute the sum of the order amounts per group based on a specific change in the date or Boolean field (in this case, per month). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can also compute the sum of the values in the set. See *SummaryFunction (x)*, Page 599.

SummaryFunction (fld)

Arguments

fld is any valid database or formula field that can be evaluated by the function.

Returns

Fractional Number

Action

SummaryFunction(*fld*) summarizes the values in the specified field for the entire report. It generates, in effect, a "grand total" summary.

SummaryFunction (fld, condFld)

Arguments

- *fld* is any valid database or formula field that can be evaluated by the function.
- *condFld* is any valid database or formula field used to group the values in *fld* by.

Returns Fractional Number

Action SummaryFunction(fld,condFld) summarizes each group of values that is generated when the specified summary condition is met.

SummaryFunction (fld, condFld,cond)

Arguments

- *fld* is any valid database or formula field that can be evaluated by the function.
- *condFld* is any valid date or Boolean field used to group the values in fld by.
- *cond* is a text string indicating the condition of condFld that controls grouping. Valid strings for this argument depend on whether condFld contains date or Boolean values.

Returns Fractional Number

Action SummaryFunction(fld, condFld, cond) summarizes each group of values that is generated when the specified summary condition is met. These functions work just like SummaryFunction(fld, condFld), but, because they use a date or Boolean field as a sort and group by (trigger) field, they require a condition in addition to the other arguments.

SummaryFunction (x)

Arguments *x* is an array of values that can be evaluated by the function being used.

Returns Fractional Number

Action SummaryFunction(x) summarizes the values in an array of constants, data field values, or formulas (a*b, c/d, etc.) separated by commas.

Time (dateTime)

Time (hour, min, sec)

Arguments

Time(dateTime)

— *dateTime* is a date/time value.

Time(hour, min, sec)

— *hour* is a whole number representing an hour of the day.

— *min* is a whole number representing a minute.

— *sec* is a whole number representing a second.

Returns

Time Value

Action

Creates a time value from the information provided.

ToNumber(x)

Arguments

x is a text string that holds numeric text.

Returns

Fractional Number

Action

ToNumber converts a text string to a number.

In a database, some numbers are stored in numeric fields, as numbers, and some are stored in string fields, as text. You make the determination which fields are to be numeric and which are to be text when you set up the database in the first place. Numbers on which you might wish to perform arithmetic (item cost, quantity ordered, etc.) are typically stored in numeric fields; numbers on which you do not expect to perform arithmetic (customer number, telephone number, etc.) are typically stored in text objects.

ToNumber allows you to convert a number stored as text to a number on which you can perform arithmetic.

ToText (x), ToText (x, y), ToText (x, y, z) ToText (x, y, z, w), ToText (x, y, z, w, q)

Arguments

- Converting Boolean values
 - *x* is a Boolean value that is converted to a string, either "True" or "False".
- Converting fractional and currency values
 - *x* is a fractional or currency value to be converted into a text string.
 - *y* is a whole number indicating the number of decimal places to carry the value in *x* to. (This argument is optional.)
 - *z* is a single character text string indicating the character to be used to separate thousands in *x*. Default is the character specified in your International or Regional settings control panel. (This argument is optional.)
 - *w* is a single character text string indicating the character to be used as a decimal separator in *x*. Default is the character specified in your International or Regional settings control panel. (This argument is optional.)
- Converting and formatting fractional and currency values
 - *x* is a fractional or currency value to be converted into a text string.
 - *y* is a text string used to indicate the format for displaying the value in *x*. Please refer to the Comments section below for information on creating a format string.
 - *z* is a whole number indicating the number of decimal places to carry the value in *x* to. (This argument is optional.)

- *w* is a single character text string indicating the character to be used to separate thousands in *x*. Default is the character specified in your International or Regional settings control panel. (This argument is optional.)
- *q* is a single character text string indicating the character to be used as a decimal separator in *x*. The default is the character specified in your International or Regional settings control panel. (This argument is optional.)
- Converting date values
 - *x* is a date value to be converted into a text string.
 - *y* is a text string that defines how the value in *x* is to be formatted. Please refer to the Comments section below for more information on creating this format string. (This argument is optional.)
- Converting time values
 - *x* is a time value to be converted into a text string.
 - *y* is a text string that defines how the value in *x* is to be formatted. Please refer to the Comments section below for more information on creating this format string. (This argument is optional.)
 - *z* is a text string to be used as a label for A.M. (morning) hours. (This argument is optional.)
 - *w* is a text string to be used as a label for P.M. (evening) hours. (This argument is optional.)
- Converting date/time values
 - *x* is a date/time value to be converted into a text string.
 - *y* is a text string of characters that indicate how the resulting text string will be formatted. Please refer to the Comments section below for more information on creating a format string. (This argument is optional.)
 - *z* is a text string to be used as a label for A.M. (morning) hours. (This argument is optional.)
 - *w* is a text string to be used as a label for P.M. (evening) hours. (This argument is optional.)

Returns Text String

Action The ToText function converts Boolean values, fractional numbers, currency, date, time, and date/time values to text strings.

ToWords (x)

ToWords (x, #places)

Arguments

- *x* is a fractional number to be converted into words.
- *#places* is a whole number indicating the number of decimal places to be converted. (This argument is optional.)

Returns Text String

Action You can use this function to convert a number or currency field value or the result of a numeric calculation to words so it can be used as text. The ability to adjust the number of decimal places can be useful when the number is the result of a calculation that may produce more decimal places than you want.

Trim (str)

Arguments *str* is a text string to be trimmed.

Returns Text String

Action The Trim function removes leading and trailing spaces from string arguments.

TrimLeft (str)

Arguments	<i>str</i> is a text string stored right-justified.
Returns	Text String
Action	TrimLeft removes all spaces to the left of a string or data field which is stored as a right-justified string in a database.

TrimRight (str)

Arguments	<i>str</i> is a text string stored left-justified.
Returns	Text String
Action	TrimRight removes all spaces to the right of a string or data field which is stored left-justified in a database.

Truncate (x)

Truncate (x, #places)

Arguments	<ul style="list-style-type: none">• <i>x</i> is a fractional number to be truncated.• <i>#places</i> is a whole number indicating how many decimal places are to remain after the value is truncated. (This argument is optional.)
Returns	Fractional Number
Action	Truncate(<i>x</i>) returns a whole number by truncating the number at the decimal point. If the <i>#places</i> argument is specified, the number

is truncated to the decimal place indicated and the function returns a fractional number. If #places is negative, the number is rounded to the first ten, hundred, and so on.

UpperCase (str)

Arguments

str is a text string.

Returns

Text String

Action

UpperCase prints the text string or text value in the data field in uppercase (capital letters).

Val (str)

Arguments

str is a text string.

Returns

Fractional Number

Action

This function reads a string containing numbers (e.g. an address, phone number, etc.) and converts them to a decimal value. Val stops reading the string when it finds the first character in the string that it finds that it can not recognize as a number or as a space.

Variance (fld), Variance (fld, condFld) Variance (fld, condFld, cond), Variance (x)

Action

Crystal Reports enables you to find the variance of a set of values in your report. For example:

- You can calculate the grand total variance for all values in a field. See *SummaryFunction (fld)*, Page 598.

- You can calculate the variance for all values within a group (for example, sales grouped by the state that they come from). See *SummaryFunction (fld, condFld)*, Page 598.
- You can calculate the variance for all values within a group in which grouping is controlled by changes in a date or Boolean field (for example, sales grouped by the month in which they were made). See *SummaryFunction (fld, condFld, cond)*, Page 599.
- If you specify a set of individual values, you can compute the variance of the values in the set. See *SummaryFunction (x)*, Page 599.

WeekToDateFromSun

Returns None

Action Specifies a range of date values that includes all days from last Sunday to Today (including today).

WhilePrintingRecords

Returns None

Action Forces the program to evaluate the formula while it is printing database record data.

WhileReadingRecords

Returns None

Action Forces the program to evaluate the formula while it is reading database record data.

Year (x)

Arguments

x is a date value or a date/time value.

Returns

Whole Number

Action

Year extracts the year from a date and returns it as a number.

YearToDate

Returns

None

Action

Specifies a range of date values that includes all days from the first day of the calendar year to today.

25

Operators and Variables

What you will find in this chapter...

Introduction 610

Operators are listed in alphabetical order.

Introduction

This chapter provides an in-depth look at all Crystal Reports operators.

Operators are special symbols or words that describe an operation or an action to take place between two or more values. Operators are used in formulas. Crystal Reports reads the operators in a formula and performs the actions specified.

This chapter begins with a brief summary of operators. Following the summary, there is an explanation of each operator. Operators are listed in alphabetical order.

NOTE: Crystal Reports operators (*Equal, In string, etc.*) are case sensitive. Thus "ABC" is not equal to "abc", "abc" is not in the string "ABCDEF", etc.

Add (+)

Usage

$x + y$

«Add values x and y.»

Example(s)

$5 + 6$

`{file.QTY1} + {file.QTY2}`

«Returns 1487, where Qty1 = 366 and Qty2 = 1121.»

`{file.AMT1} + {file.AMT2} + {file.AMT3} +
{file.AMT4}`

«Returns 20, where Amt1 = 2, Amt2 = 4, Amt3 = 6, Amt4 = 8.»

`{file.CLASS1} + 25`

«Returns 37, where Class1 = 12.»

`Date(1991,04,15) + 12 = Apr 27 91`

And (and)

Usage

x and y

False and False = False
False and True = False
True and False = False
True and True = True

Example(s)

```
If {file.CREDIT LIMIT} = 5000 And  
{file.SALESMAN} = "SP" Then  
    {file.AMOUNT}  
Else  
    0;
```

«If the credit limit is 5000 and the salesman is SP (both conditions true) then return the value in the Amount field, otherwise return zero. See *How to create If-Then-Else formulas*, Page 270.»

A > B and B > C = TRUE

«Where A = 10, B = 6, and C = 3 (both conditions true).»

A > B and B > C = FALSE

«Where A =10, B =6, and C =7 (only 1 of the 2 conditions true).»

(A>B) and (A*C-D>E) and (E/D<=B) = TRUE

«Where A = 7, B = 5, C = 3, D = 2, E = 10 (all three of the conditions are true).»

NOTE: See Page 264 for precedence rules.

Assignment (:=)

Usage

x := n

«Assigns the value n to the variable x. (x must have already been declared in the same formula.)»

Example(s)

```
Amount := 0
```

«Initializes (zero's out) the Amount variable.»

```
Amount := 100
```

«Assigns the value 100 to the Amount variable.»

```
Amount := Amount + {detail.QTY}
```

«Assigns the result of a calculation to the Amount variable. The calculation adds the value of the quantity field ({detail.QTY}) to the current value of the Amount variable.»

```
Amount := {detail.QTY1} + {detail.QTY2} +  
{detail.QTY3}
```

«Totals the three quantity fields and assigns the total to the Amount variable.»

```
Customer := "Westside Motors"
```

«Assigns the string "Westside Motors" to the Customer variable.»

```
Customer := TrimRight({customer.FIRST NAME}) +  
" " + {customer.LAST NAME}
```

«Trims the trailing blanks from the first name field {customer.FIRST NAME}, adds a space, concatenates the last name field {customer.LAST NAME}, and assigns the concatenated value of both fields to the Customer variable.»

```
Customer := "Mr. " + {customer.LAST NAME}
```

«Concatenates the string "Mr. " with the value of the last name field {customer.LAST NAME}, and assigns the concatenated value to the Customer variable.»

```
Amount := 100; Customer := "Westside Motors"
```

«Assigns the constant 100 to the number variable Amount and assigns the string "Westside Motors" to the string variable Customer. You can assign values to multiple variables by separating the assignment statements with semicolons.»

Comment (//)

Usage

```
// text
```

«The text that follows the operator is a comment; it is not treated as part of the formula, and it does not print.»

Example(s)

```
{file.SALES} - {file.COST}
// calculates the gross profit
```

«{file.SALES} - {file.COST} is the formula; the Formula Editor ignores everything else.»

```
{{file.SALES} - {file.QUOTA}} * .06
//calculates sales commission
```

«({file.SALES} - {file.QUOTA}) * .06 is the formula; the Formula Editor ignores everything else.»

```
// What follows is
// a formula. Note that when
// we force the line break we have
// to begin each new line with a
// comment operator
```

```
If {file.COST} > {file.SELLPRICE} Then
    "Loss"
Else
    "";
```

```
//That was a formula
```

«All commented lines that appear before or after the formula are ignored.»

Concatenate (+)

Usage

```
x + y
```

«Concatenates (connects) string x to string y to make one contiguous string.»

Example(s)

```
"Bread " + "and " + "butter"
```

«Returns "Bread and butter".»

```
"Your customer number is " +  
({customer.CUSTOMER ID}) + " and your company  
contact person is " + ({customer.FIRST NAME})  
+ "."
```

«Returns "Your customer number is 64 and your company contact person is Bob." where CustomerID = 64 and First Name = Bob.»

Divide (/)

Usage

x/y

«Divide value x by value y.»

Example(s)

```
25/5
```

«Returns 5.»

```
1/3
```

«Returns .33333.»

```
{file.SALES} / {file.FORECAST}
```

«Returns .875, where Sales = 52533, Forecast = 60000.»

```
{file.DAYS DUE} / 5
```

«Returns 22, where DaysDue = 110.»

NOTE: If the denominator = 0, the report will be halted with a divide by zero warning. If you want to avoid this type of problem, you should put in a test similar to the following:

```
If {file.FORECAST} = 0 Then  
    0  
Else  
    {file.SALES} / {file.FORECAST}
```


Equal (=)

Usage

`x = y`

«x is equal to y.»

The Equal operator tells Crystal Reports to evaluate an expression (`x = y`) and return a TRUE (if x is equal to y) or FALSE (if x is not equal to y).

Example(s)

`{file.QUANTITY} = 3`

«True, where Quantity has a value of 3.»

`{file.QUANTITY} = 3`

«False, in all other situations.»

`{file.YTD} = {file.LAST YEAR YTD}`

«True, where the value of the field YTD is identical to the value of the field Last Year YTD.»

`{file.YTD} = {file.LAST YEAR YTD}`

«False, in all other situations.»

`((file.SALES) - {file.COGS}) = 22,554`

«True, where Sales = 109,986 and COGS = 87,332.»

`((file.SALES) - {file.COGS}) = 22,654`

«False, in all other situations.»

`{customer.LAST NAME} = "Johnson"`

«True, where the text string in the Last Name field is Johnson.»

`{customer.LAST NAME} = "Johnson"`

«False, in all other situations.»

Greater Than (>)

Usage

`x > y`

«x is greater than y.»

Example(s)

`{file.WEIGHT} > 200`

«False, where Weight = 150, Weight = 199, or Weight = 200.»

`{file.WEIGHT} > 200`

«True, where Weight = 400 or Weight = 201.»

`{file.COST} > {file.PRICE}`

«True, where Cost = 350 and Price = 325.»

Greater Than Or Equal (>=)

Usage

`x >= y`

«x is greater than or equal to y.»

Example(s)

`{file.WEIGHT} >= 200`

«False, where Weight = 150 or Weight = 199.»

`{file.WEIGHT} >= 200`

«True, where Weight = 400, Weight = 200, or Weight = 201.»

`{file.COST} >= {file.PRICE}`

«False, where Cost = 350, Price = 400.»

If-Then-Else

Usage

If x Then y Else z

«If x is true Then do y. If x is not true (Else), do z.»

Example(s)

```
If {customer.POSTAL CODE} <= "49999" Then
    "Blue Label"
Else
    "Ground";
```

«Assigns method of shipping based on distance from ship point.»

```
If ToNumber({file.ITEM}) >= 2500 And
ToNumber({file.ITEM}) < 2600 Then
    "Seasonal"
Else
    "";
```

«If statement includes an And operator.»

```
If {file.COUNT} >= 25 Then
    {file.DISTRIBUTOR} * {file.COUNT}
Else
    {file.DEALER} * {file.COUNT};
```

«Quantity ordered determines price list used.»

```
If {file.ONHAND} > 10 Then
    {file.ORDERED}
Else
    If {file.ORDERED} < 5 Then
        {file.ORDERED}
    Else
        2;
```

«Allocation based on quantity ordered using nested If-Then-Else.»

In Array (in)

Usage

`x in [y]`

«Is x in the array y?»

Example(s)

`{customer.REGION} in ["CA", "HI", "AK"]`

«Is the value of the Region field in the array of state abbreviations listed in the brackets?»

`{file.COLOR} in ["Red", "White", "Blue"]`

«Is the value of the Color field in the array of colors listed in the brackets?»

`DayofWeek({orders.ORDER DATE}) in [2,4,6]`

«Is the value of the Order Date field, converted to a number that represents the day of the week, in the array of numbers listed in the brackets? (Sunday = 1, Saturday = 7)»

In Range (in)

Usage

`x in y`

«Tests a range of values (y) to see if a value (x) falls within the range specified.»

Example(s)

`Today in Date(1990,09,01) to Date(1990,09,20)`

«True, where Today = September 15, 1990.»

`Today in Date(1990,09,01) to Date(1990,09,20)`

«False, where Today = September 21, 1990.»

`{file.QTY} in {file.ONHAND} to
({file.BACKORDER} + {file.ONORDER})`

«True, where Qty=20, OnHand=10, Backorder=5, OnOrder=25. (is 20 in the range that begins with 10 and ends with the sum of 5 and 25?)»

```
{file.QTY} in {file.ONHAND} to  
({file.BACKORDER} + {file.ONORDER})
```

«False, where Qty=31, OnHand=10, Backorder=5, OnOrder=25.
(is 31 in the range that begins with 10 and ends with the sum of 5
and 25?)»

In String (in)

Usage

`x in y`

«Test for the presence of string x in string y.»

Example(s)

```
"Elm" in {customer.ADDRESS1}
```

«True, where Address is 1335 Elm Street.»

```
"elm" in {customer.ADDRESS1}
```

«False, where Address is 1335 Elmer Street.»

```
"elm" in {file.MOTTO}
```

«False, where Motto = "Feel more energy".»

(The "el" ending of "feel" and the "m" beginning of the word
"more" are separated by a space which itself counts as an element.)

```
"el m" in {file.MOTTO}
```

«True, where Motto = "Feel more energy".»

(The search string "el m" this time contains the space between the
l and the m which allows for a perfect match.)

**NOTE: The "in" operator can also be used to test for the presence
of a string in a text range (i.e., "V5B" in "V0A" to "V9Z").**

Less Than (<)

Usage

`x < y`

«x is less than y.»

The less than operator tells the Formula Editor to evaluate an expression ($x < y$) and return a TRUE (if x is less than y) or FALSE (if x is equal to or greater than y).

Example(s)

```
{file.WEIGHT} < 200
```

«True, where Weight = 150 or Weight = 199.»

```
{file.WEIGHT} < 200
```

«False, where Weight = 200 or Weight = 400.»

```
{file.COST} < {file.PRICE}
```

«True, where Cost = 350 and Price = 400.»

```
{file.COST} < {file.PRICE}
```

«False, where Cost=350 and Price=350, or where Cost=350 and Price=325.»

Less Than Or Equal (<=)

Usage

$x \leq y$

« x is less than or equal to y .»

Example(s)

```
{file.WEIGHT} <= 200
```

«True, where Weight = 150, Weight = 200, or Weight = 199.»

```
{file.WEIGHT} <= 200
```

«False, where Weight = 400.»

```
{file.COST} <= {file.PRICE}
```

«True, where Cost=350, Price=400, or where Cost=350 and Price=350.»

```
{file.COST} <= {file.PRICE}
```

«False, where Cost = 350 and Price = 325.»

Like

Usage

x like y

{file.FIELD} like "c?n*"

«This operator tests to see if the contents of {file.FIELD} matches a pattern that you specify in a character string "c?n*". If the contents of the field fit the pattern "c?n*", then the formula returns the value True. If the field starts with anything else, the formula returns False.

Use the wildcard symbols ? and * to stand for variable characters. The ? stands for a single character. The * symbol stands for any number of characters.»

Example(s)

{customer.FIRST NAME} like "D?n"

«True, where {customer.FIRST NAME} is Dan or Don.»

{customer.FIRST NAME} like "D?n"

«False, where {customer.FIRST NAME} is Doug or Rick.»

{customer.LAST NAME} like "*s?n*"

«True, where {customer.LAST NAME} is Johnson, Olson, or Olsen.»

{customer.LAST NAME} like "*s?n*"

«False, where {customer.LAST NAME} is Johnston or Smith.»

Make Array ([,])

Usage

[x, y, z,...n]

«Build an array containing the elements x, y, z, ... n.»

Example(s)

[100,200,300,400]

[{file.QTYA}, {file.QTYB}, {file.QTYC}]

```
[({file.AMT1} *.5),({file.AMT2} *.5),  
{file.AMT3} *.25]  
[500,({file.QTY} /3)]
```

NOTE: *You can not have more than one data type in an array.*

Make Range (to)

Usage

x to y

«Create the range x to y.»

Example(s)

100.00 to 250.00

«The range of consecutive numeric values beginning with 100.00 and ending with 250.00, including the beginning and end values.»

Date(1990,09,01) to Date(1990,09,20)

«The range of consecutive dates beginning with September 1, 1990 and ending with September 20, 1990. Both September 1 and September 20 are included in the range.»

"Aaron" to "Lusk"

«The range of consecutive string values beginning with Aaron and ending with Lusk, including the beginning and end values.»

Multiply (*)

Usage

x * y

«Multiply value x by value y.»

Example(s)

2883 * 1999

«Returns 5,763,117.»

{file.EXMPT} * 356.00

«Returns, 152,012, where Exmpt = 427.»

Negate (-())

Usage

`(- (x))`

«Multiply the value inside the parentheses by -1.»

Example(s)

`- (-1)`

«Returns 1; negative times negative = positive.»

`- (1)`

«Returns -1; negative times positive = negative.»

`- ({ file.QTYONHND })`

«Returns 144, where QtyOnHnd = -144.»

`- (- (15-18))`

«Returns -3; 15-18 = -3, -(-3)= +3, -(+3)= -3»

Not (not)

Usage

`Not (x)`

«Reverses the True or False value of x.»

`Not (True) = False`

`Not (False) = True`

`Not (Not (False)) = False`

`Not (Not (True)) = True`

Example(s)

`not (A > B and B > C)`

«If A = 5, B = 4, C = 3, the expression (A > B and B > C) is TRUE. Both conditions tied together by the Boolean operator And are TRUE, thus the entire statement has a value of TRUE. The Not operator changes the value of the expression to FALSE.»

`not (A > B and B > C)`

«If A = 3, B = 4, C = 3, the expression (A > B and B > C) is FALSE. One of the two conditions tied together by the Boolean operator And is FALSE, thus the entire statement has a value of FALSE. The Not operator changes the value of the expression to TRUE.»

```
not ({file.ONHAND} - {file.ORDER} > 0) = TRUE
```

«Where OnHand = 10 and Order = 11.»

```
not ({file.ONHAND} - {file.ORDER} > 0) = FALSE
```

«Where OnHand = 10 and Order = 9.»

Not Equal (<>)

Usage

`x <> y`

«x is not equal to y.»

The Not Equal operator tells Crystal Reports to evaluate an expression (`x<>y`) and return a TRUE (if x is not equal to y) or FALSE (if x is equal to y).

Example(s)

```
{order.ORDER AMOUNT} <> 400
```

«True, where Amount equals 200 or Amount equals 401, etc.»

```
{order.ORDER AMOUNT} <> 400
```

«False, where Amount equals 400.»

```
{file.DAY} <> "Thursday"
```

«True, when Day = Friday.»

```
{file.DAY} <> "Thursday"
```

«False, when Day = Thursday.»

```
{file.ONHAND} <> 0
```

«True, where the value of OnHand is 10.»

```
{file.ONHAND} <> 0
```

«False, where the value of OnHand is 0.»

```
{file.AVAILABLE} - {file.USED} <> 10
```

«True, where the value of the Available field less the value of the Used field gives a result other than 10.»

```
{file.AVAILABLE} - {file.USED} <> 10
```

«False, where it gives a value of 10.»

Or (or)

Usage

x or y

«Either x or y or both is true.»

False or False = False

False or True = True

True or False = True

True or True = True

Example(s)

```
If {file.CREDIT LIMIT} = 5000 or  
{file.SALESMAN} = "SP" Then  
    {file.AMOUNT}  
Else  
    0
```

«This means that if the credit limit is 5000, or, if the salesman is SP (either of the conditions are true) then return the value in the Amount field, otherwise return nothing.»

```
A > B or B > C
```

«True, where A = 10, B = 6, and C = 3 (both conditions are TRUE).»

```
A > B or B > C
```

«False, where A = 5, B = 6, and C = 7 (neither of the 2 conditions are TRUE).»

```
(A > B) or (A * C - D > E) or (E / D <= B)
```

«True, where A = 5, B = 5, C = 3, D = 2, E = 12.»

(At least one of the three conditions is true. In this case, only one condition (A * C - D > E) is true.)

Parentheses (())

Usage

$(x + y) * z$

«Performs the calculations inside the parentheses first.»

Parentheses are used to control the order in which the Formula Editor calculates a formula.

Example(s)

$8 + 6 * 3 - 6 / 2 = 23$

$(8 + 6) * 3 - 6 / 2 = 39$

$(8 + 6) * (3 - 6 / 2) = 0$

$(8 + 6 * 3 - 6) / 2 = 10$

$\{file.SALES\} - \{file.COGS\} - \{file.T\&E\} *.8$

«Returns 11,800, where Sales = 25,000, COGS = 12,000, and T&E = 1500.»

$\{file.SALES\} - ((\{file.COGS\} - \{file.T\&E\}) *.8)$

«Returns 16,600, where Sales = 25,000, COGS = 12,000, and T&E = 1500.»

$(\{file.SALES\} - \{file.COGS\} - \{file.T\&E\}) *.8$

«Returns 9200, where Sales = 25,000, COGS = 12,000, and T&E = 1500.»

Percentage (%)

Usage

$x \% y$

«Calculate value x as a percentage of value y $[(x / y) * 100]$.»

Example(s)

Balance outstanding % Credit Limit means the value of Balance Outstanding is what percent of the value of Credit Limit.

$\{file.BALANCE\ OUTSTANDING\} \%$

$\{file.CREDIT\ LIMIT\}$

«Returns 30.00, where Balance outstanding = \$1500 and Credit Limit = \$5000.»

Amount % Credit Limit means the value of Amount is what percentage of the value of Credit Limit.

```
{file.AMOUNT} % {file.CREDIT LIMIT}
```

«Returns 32.26, where Amount =2257.87 and Credit Limit =7000.»

NOTE: If the denominator = 0, the report will be halted with a divide by zero warning. If you want to avoid this type of problem, you should put in a test similar to the following:

```
If {file.FORECAST} = 0 Then  
    0  
Else  
    {file.SALES} % {file.FORECAST}
```

startsWith

Usage

x startsWith y

```
{file.FIELD} startsWith "abc"
```

«This operator tests to see if the contents of {file.FIELD} starts with a character string that you specify: "abc". If the contents of a field do start with the specified string, the formula returns the value True. If the field starts with anything else, the formula returns False.»

Example(s)

```
{customer.CUSTOMER NAME} startsWith "A"
```

«True, where {customer.CUSTOMER NAME} = ABC Inc.»

```
{customer.CUSTOMER NAME} startsWith "XYZ"
```

«False, where {customer.CUSTOMER NAME} = ABC Inc.»

Statement separator (;)

Usage

1 + 1; "abc";

«1 + 1 and "abc" are two different formula statements in a multiple statement formula. The semicolon between the statements specifies where one statement ends and the next one begins. Without the semicolon the statements would be treated together as an individual statement.»

Subscript ([])

Usage

x[y]

«Extract the y element from string x.»

x[y to z]

«Extract the y to z range of elements from string x.»

x[n]

«Extract the n element of array x.»

NOTE: The subscript ranges are 1 origin; they start at 1 rather than 0.

Subscript is used to extract one or more characters from a text string or to extract an element from an array.

Example(s)

[100,233,466,998][3]

«Returns 466; 466 is the third element in the array.»

LASTNAME [1]

«Returns S, where Last Name = Smith.»

Postal Code [6]

«Returns V, where Postal Code = T5A 9V2 (the space between A and 9 counts as an element).»

{customer.POSTAL CODE} [5 to 7]
«Returns 9V2, where Postal Code = T5A 9V2.»
{file.ITEMNUMBER} [4 to 5]
«Return 40, where ItemNumber is A1/4020/B10.»

Subtract (-)

Usage

x-y
«Subtract value y from value x.»

Example(s)

244 - 112
«Returns 132.»
{file.SALES} - {file.COGS} - {file.S&A}
«Returns 214,972, where Sales = 455,031, COGS = 188,213, and S&A = 51,846.»
{file.ONHAND} - 877
«Returns 114, where OnHand = 991.»
Date(1991, 04, 15) - 12
«Returns Apr 03 91.»

To Currency (\$)

Usage

\$x
«Convert x from number to currency.»

Example(s)

The following examples all assume the following format: Decimal Places = (1.00), Negative sign = (345.00-), Currency Symbol = (Float), and Thousands Separator = (1,000.00).

\$12345678

«Returns \$12,345,678.00.»

```
$(123 * 456)
```

«Returns \$56,088.00.»

```
${file.QUANTITY} * 3)
```

«Returns \$42.00, where Quantity = 14.»

```
${file.MILES} * {file.PLEDGE})
```

«Returns \$363.35, where Miles = 169 and Pledge = 2.15.»

NOTE: \$ * \$ = error. You cannot multiply a dollar by a dollar.

Variable Declarators

Usage

```
NumberVar x;  
CurrencyVar x;  
StringVar x;  
BooleanVar x;  
DateVar x;  
TimeVar x;  
DateTimeVar x;
```

«Declares a variable *x* that can hold data of a type corresponding to the variable declarator used.»

Example(s)

```
NumberVar Amount;
```

«Declares a variable named *Amount* that can hold any value of an integer or decimal data type.»

```
BooleanVar Outstanding;
```

«Declares a variable named *Outstanding* that can hold a TRUE or FALSE value.»

```
StringVar LastName := "Adams";
```

«Declares a variable named *LastName* that can hold a string value and assigns the string "Adams" to that variable.»

26

Application Development with Crystal Reports

What you will find in this chapter...

- Introduction To The Crystal Report Engine 632
- Before using the Report Engine in your application 633
- Using The Crystal Report Engine 634
- The Crystal Custom Control 636
- The Crystal ActiveX Control (OCX) 639
- The Crystal Visual Component Library for Delphi 642
- The Report Engine Class Library 645
- The Report Engine API 647
- Distributing Report Engine Applications 669
- Additional Sources of Information 670

Introduction To The Crystal Report Engine

Besides acting as a powerful stand-alone report creating application, Crystal Reports provides a report writing module that you can add to your own applications. As a developer using C, C++, Visual Basic, ObjectVision, Turbo Pascal, Visual dBASE, Delphi, or any programming language that can access a DLL, you can add sophisticated report generating and printing capabilities to your applications without the time-consuming task of writing your own code.

The Crystal Report Engine is a Dynamic Link Library (DLL) that allows your applications to access the same powerful report printing features that are available in Crystal Reports. As a licensed user of Crystal Reports, you receive royalty-free rights to ship the Report Engine DLL (CRPE.DLL or CRPE32.DLL) and all of its support files with any application you create.

NOTE: For more information regarding current runtime file requirements, please see the Runtime File Requirements Help file.

From your application, you access the Report Engine through any of several Report Engine development tools:

- The Crystal Custom Control, CRYSTAL.VBX,
- The Crystal ActiveX Control, CRYSTAL.OCX or CRYSTL32.OCX,
- The Crystal Visual Component Library for Delphi, UCRPE.DCU or UCRPE32.DCU,
- The Report Engine Class Library, PEPLUS.H and PEPLUS.CPP, or
- The Report Engine API, CRPE.DLL or CRPE32.DLL.

When your application runs, it links with the Report Engine to access report writing functionality. Reporting can be simple, producing only a single report that is sent to a printer or Preview window with no options available to the user, or it can be complex, allowing the user to change such things as record selection, sorting, grouping or export options.

NOTE: All references to CRPE.DLL are for the 16-bit version. If you plan on using the 32-bit version, it is called CRPE32.DLL.

Special Features Of The Report Engine

SQL AND ODBC

The Crystal Report Engine is fully compatible with most popular SQL DBMS applications, including Sybase SQL Server, Oracle, Gupta SQLBase, and Microsoft SQL Server. The Report Engine includes options for logging on and off of SQL servers and other ODBC data sources and also includes the ability to edit the SQL statement passed through to an SQL or ODBC database. See *How to set up an ODBC data source*, Page 453 and *The SQL Language*, Page 441.

EXPORTING

The Report Engine enables you to print to a printer or a Preview window with simple function calls. In addition, you can export a file:

- through e-mail to another person on your network,
- directly to disk,
- to a Microsoft Exchange folder,
- to a Lotus Notes folder, or
- to an ODBC data source.

The report can be exported in any of several word processing, spreadsheet, database file, or data exchange formats including HTML. See *How to export reports*, Page 110.

Before using the Report Engine in your application

Before you add the Crystal Report Engine to your application, you should be familiar with some key features of the Report Engine. Review the following points, and make sure you understand each before attempting to make calls to the Report Engine from your application.

- The Report Engine outputs existing reports. You can not create report files using the functionality of the Report Engine. Reports must be created using the Crystal Reports application described throughout this manual. Make sure you understand the report creation process before trying to generate reports with the Report Engine.

- The Report Engine provides a convenient add-on to your existing application development project. With just a few lines of code, you produce a powerful report writing and distribution tool that would take thousands of lines of code and weeks to produce otherwise.
- The Report Engine does not require the use of a fixed user interface. The Report Engine is designed to work with your existing development project and allows you to define the user interface your customers and users are familiar with and expect from your application.

Using The Crystal Report Engine

Any development project that incorporates the Crystal Report Engine requires three steps:

- Create the reports your users will access,
- Design the user interface that will drive the Report Engine, and
- Add the Report Engine to your application.

Creating reports

Creating reports to include with your applications is identical to creating reports for your own use; there are no restrictions. Using the procedures outlined in the Crystal Reports User's Guide and online Help, create as many kinds of reports as you want to make available to your users. You can make the reports as simple or as sophisticated as your needs dictate.

Creating the interface for printing the reports

The interface you develop to allow users to print reports is limited only by your needs and your imagination. The kind of user interface you select is unimportant to the Report Engine.

Common methods of using the Report Engine include a single menu command that produces a single report, a dialog box allowing several options for printing reports, or a completely separate front-end application that is called by your application. All are acceptable techniques, and each has its advantages. How you design your user interface can depend on any or all of the following:

- The purpose of your application,

- The types of reports your application will use,
- The printing options you want to make available with those reports, and
- Whether your application will offer only one report or a choice of several reports.

Consider your application and your reporting needs carefully, and design the User Interface that will use the Report Engine most efficiently.

Adding the Report Engine to your application

As described earlier in this chapter, there are five different Report Engine development tools that can be used to add the Crystal Report Engine to your application:

- The Crystal Custom Control,
- The Crystal ActiveX Control,
- The Crystal Visual Component Library for Delphi,
- The Report Engine Class Library, and
- The Report Engine API.

Be aware that you can not use two or more of these tools in the same application. For example, you can not create a Visual Basic application that contains the Crystal Custom Control and also makes calls to the functions in the Report Engine API. You must choose one tool to implement the Report Engine in your project and stick with that tool.

When choosing a Report Engine tool, consider the following:

- What is your development environment?
- What is your programming ability?
- Do you need to implement the entire Report Engine or just a few features of it?

For example, the Crystal Visual Component Library for Delphi only works in the Borland Delphi programming environment. Therefore, if you are programming in Visual Basic, the Crystal VCL is not an option. The Report Engine Class Library, on the other hand, is based on the Microsoft Foundation Class Library for C++. To use the Report Engine Class Library, you must be

using a C++ development tool, and you must be using the MFC library.

If you are an experienced programmer, you might consider the Report Engine API or the Report Engine Class Library. Novice programmers, on the other hand, may want to take advantage of the easy-to-use features of the Crystal Custom Control, the ActiveX Control, or the Visual Component Library.

The rest of this chapter examines the steps needed with each tool to add the Report Engine to your application. Remember, you can only use one tool at a time in an application. Review each section and decide which Report Engine tool is best for your development project.

The Crystal Custom Control

The Crystal Custom Control is a VBX file first supported by Visual Basic 3.0. Although this type of add-on control has been replaced by the more powerful ActiveX control, many development applications, including Visual Basic 4.0, continue to support VBX controls.

NOTE: The Crystal Custom Control, and any VBX, is only available for 16-bit development environments.

Adding the Custom Control to your project

This section demonstrates how to add the Crystal Custom Control to an application project being designed in Visual Basic 4.0. If you wish to use the Custom Control in a different development environment, please refer to the documentation that came with your development tools for information on adding a VBX control to your project.

The Crystal Custom Control is installed in the \WINDOWS\SYSTEM directory when you install Crystal Reports. You add the Crystal Custom Control to your Visual Basic project using the Custom Controls command on the Visual Basic Tools menu.

To add the Crystal Custom Control:

1. Open Visual Basic.
2. Open the project to which you want to add the Custom Control.

3. Choose the CUSTOM CONTROLS command from the Tools menu. The Custom Controls dialog box appears.
4. If Crystal Custom Control appears in the *Available Controls* list, click the check box next to it, click OK, and skip to Step 8.
5. If Crystal Custom Control does not appear in the *Available Controls* list, click *Browse*, and the Add Custom Control dialog box will appear.

NOTE: Do not confuse Crystal Custom Control with Crystal Report Control. Crystal Custom Control is the Crystal VBX. Crystal Report Control is the Crystal ActiveX control.

6. Use the controls in the Add Custom Control dialog box to locate and select the CRYSTAL.VBX file. This file is installed in your \WINDOWS\SYSTEM directory by default. Once you locate and select the file, click *Open*.
7. Crystal Custom Control will now appear in the *Available Controls* list box. Click the check box next to the name of the control, and click OK.
8. Visual Basic adds the Crystal Custom Control to your toolbox. The tool looks like this:



9. When you want to add the Custom Control to a form, double-click the tool and the program installs it on the active form.

NOTE: CRYSTAL.VBX can be added to AUTOLOAD.MAK to automatically load the Custom Control to your project.

NOTE: For instructions on how to add CRYSTAL.VBX to development applications other than Visual Basic, refer to the documentation that came with the development application you are using.

Using the Custom Control

This section illustrates the basic features of the Crystal Custom Control and how to use the Custom Control in a Visual Basic 4.0 project to add Report Engine functionality.

Once you have the Crystal Custom Control object on your form, you build the connection between your application and Crystal Reports by setting the object's properties via the *Properties* list. Using the properties list you specify:

- the name of the report you want to print in response to an application event,
- the destination for that report (window, file, or printer),
- the number of copies you want to print (if your report is going to the printer),
- print file information (if your report is going to a file),
- print window sizing and positioning information (if your report is going to a window),
- selection formula information (if you want to limit the records in your report), and
- sorting information.

Crystal Custom Control properties can be changed either at design time or at runtime. Note, however, some properties are available only at runtime. These properties do not appear on the *Properties* list in Visual Basic.

NOTE: For a complete description of each property in the Crystal Custom Control, refer to the Crystal Reports Developer's online Help.

CHANGING PROPERTIES ON THE PROPERTIES LIST

To change the value for a property, highlight the property and then do the following:

- If the settings box displays an inactive (grayed-out) arrow, enter your new value in the settings box.
- If the settings box displays an active (black) arrow, click the arrow to reveal your alternatives and select a value from the drop down list that appears. Alternately you can double-click the property itself to cycle through the list of values. Move on to the next property when the value you want is displayed.
- If an ellipsis (...) appears at the right of the settings box, click the ellipsis to reveal a dialog box you can use to define your setting. Alternately you can double-click the property itself to

call up the dialog box. Define your setting in the dialog box and click *OK* when finished.

CHANGING PROPERTIES AT RUNTIME

You can set most of the Custom Control properties at runtime by adding simple entries to your procedure code. Runtime property settings replace settings you make via the Properties list at design time.

The Action property is used to actually process the report at runtime. This property can only be set at runtime, and it is the only means by which a report can actually be generated by the Custom Control.

For information on how to set Visual Basic control properties at runtime, refer to your Visual Basic documentation. The Crystal Reports Developer's online Help contains code examples for each of the Crystal Custom Control properties. Search for the property you are interested in for examples.

The Crystal ActiveX Control (OCX)

ActiveX is Microsoft's new Internet technology designed to bring more powerful applications to desktops and networks. ActiveX moves beyond applications that produce static documents to a Windows environment that provides active controls, documents, and client applications that can operate and interact not only with each other, but also with network intranets and the global Internet.

ActiveX controls, formerly known as OLE controls, provide plug-in capabilities that let you add application components, and even entire applications, to your own development projects without writing a line of code. Crystal Reports supports the ActiveX concept with the Crystal ActiveX Control. Use the ActiveX Control to easily add all of the report processing power of Crystal Reports to your own Visual Basic 4.0, Visual C++, Borland C++, Delphi, and other applications.

NOTE: Your development tools may refer to an ActiveX Control by any of the following names: OLE Control, OCX Control, Custom Control, or ActiveX Control. As long as the term used refers to a control with an .OCX filename extension, it is synonymous with the term ActiveX Control used here.

Adding the ActiveX Control to your project

This section demonstrates how to add the Crystal ActiveX Control to an application project being designed in Visual Basic 4.0. If you wish to use the ActiveX Control in a different development environment, please refer to the documentation that came with your development tools for information on adding an ActiveX or OLE control (OCX) to your project.

The Crystal ActiveX Control is installed in the \WINDOWS\SYSTEM directory when you install Crystal Reports. You add the ActiveX Control to your Visual Basic project using the Custom Controls command on the Visual Basic Tools menu.

To add the Crystal ActiveX Control:

1. Open Visual Basic.
2. Open the project to which you want to add the ActiveX Control.
3. Choose the CUSTOM CONTROLS command from the Tools menu. The Custom Controls dialog box appears.
4. If Crystal Report Control appears in the *Available Controls* list, click the check box next to it, click OK, and skip to step 8.
5. If Crystal Report Control does not appear in the *Available Controls* list, click *Browse*, and the Add Custom Control dialog box will appear.

NOTE: *Crystal Report Control is the name of the Crystal ActiveX Control when it is added to a development project. The term ActiveX Control refers to a type of control, while Crystal Report Control is the name of the ActiveX Control provided by Crystal Reports.*

6. Use the controls in the Add Custom Control dialog box to locate and select the CRYSTL16.OCX (16-bit) or CRYSTL32.OCX (32-bit) file. This file is installed in your \WINDOWS\SYSTEM directory by default. Once you locate and select the file, click *Open*.
7. Crystal Report Control will now appear in the *Available Controls* list box. Click the check box next to the name of the control, and click OK.
8. Visual Basic adds the Crystal ActiveX Control to your toolbox. The tool looks like this:



9. When you want to add the ActiveX Control to a form, double-click the tool and the program installs it on the active form.

NOTE: *The ActiveX Control can be added to AUTOLOAD.MAK to automatically load the Control to your project.*

NOTE: *For instructions on how to add an ActiveX Control or OLE control to development applications other than Visual Basic, refer to the documentation that came with the development application you are using.*

Using the ActiveX Control

Once you have the ActiveX Control object on your form, you build the connection between your application and Crystal Reports by setting the object's properties at design time or changing properties at runtime. The ActiveX properties let you specify:

- the name of the report you want to print in response to an application event,
- the destination for that report (window, file, or printer),
- the number of copies you want to print (if your report is going to the printer),
- print file information (if your report is going to a file),
- print window sizing and positioning information (if your report is going to a window),
- selection formula information (if you want to limit the records in your report),
- sorting information, and
- other related properties.

Crystal ActiveX Control properties can be changed either at design time or at runtime. Note, however, some properties are available only at runtime. These properties do not appear at design time.

NOTE: *For a complete description of each property in the Crystal ActiveX Control, refer to the Crystal Reports Developer's online Help.*

CHANGING PROPERTIES FOR THE ACTIVEX CONTROL

To change ActiveX properties:

1. Click the ActiveX control on your form to select it.
2. Right-click and choose the **PROPERTIES** command from the shortcut menu that appears. The Crystal Report Control Properties dialog box appears.
3. Use the tabs and controls in this dialog box to change the ActiveX Control properties at design time.

NOTE: ActiveX Control properties also appear in the Visual Basic Properties box. Properties can be changed here much like they are changed for the Crystal Custom Control (see Page 638).

CHANGING PROPERTIES AT RUNTIME

You can set most of the ActiveX Control properties at runtime by adding simple entries to your procedure code. Runtime property settings replace settings you make via the Properties list at design time.

Use the Action property or the PrintReport method to actually process the report at runtime. The Action property and the PrintReport method can only be used at runtime, and are the only means by which a report can actually be generated by the ActiveX Control.

For information on how to set ActiveX Control properties at runtime, refer to the Crystal Reports Developer's online Help. Online Help also provides examples of how to set each property at runtime.

The Crystal Visual Component Library for Delphi

The Crystal Visual Component Library (VCL) has been designed specially for the Borland Delphi development environment. The VCL can be easily added to any Delphi project and compiled into your final executable application. Like the Custom Control (VBX) and ActiveX Control (OCX), the VCL provides all of the report processing power available in Crystal Reports for your own Delphi projects.

NOTE: Delphi 2.0 can also use ActiveX Controls. Refer to your Delphi 2.0 documentation on how to use an OLE Control (OCX) in a Delphi project.

Adding the VCL to your project

The following instructions demonstrate how to add the Crystal VCL to a Delphi 2.0 project using the `INSTALL` command on the Component menu. If you are using Delphi 1.x, use the `INSTALL COMPONENTS` command on the Options menu and refer to your Delphi 1.0 documentation for complete instructions.

1. With your project open in Delphi, choose the `INSTALL` command from the Component menu. The Install Components dialog box appears.
2. Click *Add* to add a new component to your project. The Add Module dialog box appears.
3. Click *Browse*, and the Add Module dialog box changes to allow you to browse through your directories and files to find a component module.
4. Select Unit file (*.DCU) from the *Files of type* drop down list box. The Crystal VCL is a Delphi Unit file.
5. Use the controls in the Add Module dialog box to locate and select the Crystal VCL. This file is installed in the Crystal Reports VCL directory (\CRW\VCL by default). The file is named UCRPE.DCU (16-bit) or UCRPE32.DCU (32-bit).
6. Once you select the VCL file, click *Open* and the Add Module dialog box closes.
7. In the Install Components dialog box, the name of the Crystal VCL (UCRPE or UCRPE32) will appear in the *Installed units* list box. Select this VCL, and the *TCrpe* class will appear in the *Component classes* list box. Click *OK* to close the Install Components dialog box when finished. Delphi will recompile the component library, adding in the Crystal VCL.
8. Click the Data Access Tab on the Component Palette. The Crystal VCL appears as the last component on the Data Access Tab. When you hold the cursor over the component, a Tool Tip appears indicating the name of the component as *Crpe*.
9. Double-click the *Crpe* component to add it to the active form in your project.

Using the Crystal VCL

Once you add the Crpe component to a form in your project, you build the connection between your Delphi application and the Crystal Report Engine by setting the component's properties via the Object Inspector. Using the Object Inspector, you specify:

- the name of the report you want to print in response to an application event,
- the destination for that report (window, file, or printer),
- the number of copies you want to print (if your report is going to the printer),
- print file information (if your report is going to a file),
- print window sizing and positioning information (if your report is going to a window),
- selection formula information (if you want to limit the records in your report),
- sorting information, and
- other related properties.

Crpe component properties can be changed either at design time or at runtime. Note, however, some properties are available only at runtime. These properties do not appear on the *Properties* list in the Object Inspector.

NOTE: For a complete description of each property in the Crystal VCL, refer to the Crystal Reports Developer's online Help.

CHANGING PROPERTIES IN THE OBJECT INSPECTOR

To change the value for a property, click the property and then do the following:

- If a text box appears next to the property name, type in a value for the property.
- If a drop down list box appears next to the property name, click the arrow to reveal your alternatives and select a value from the list.
- If a text box with an ellipsis (...) button appears next to the property name, click the button to reveal a dialog box you can use to define your setting for the property.

CHANGING PROPERTIES AT RUNTIME

You can set most of the properties for the Crpe component at runtime by adding simple entries to your procedure code. Runtime property settings replace settings you make via the Object Inspector at design time.

The *Execute* property is used to actually process the report at runtime. This property can only be set at runtime, and it is the only means by which a report can actually be generated by the Crpe component.

For information on how to set component properties at runtime, refer to your Delphi documentation. The Crystal Reports Developer's online Help contains code examples for each of the Crpe component properties. Search for the property you are interested in for examples.

The Report Engine Class Library

The Report Engine Class (REC) Library provides object oriented programming of the Crystal Report Engine in C++. The REC Library is based on the Microsoft Foundation Class (MFC) Library. The two Report Engine Class definitions in the REC Library are derived from the MFC CObject class.

MFC is a widely available, and highly powerful class library originally designed for use with Microsoft's C++ development environment (now known as Visual C++). However, MFC is a complete class library designed for programming Windows applications and can be used with most C++ development tools including Borland C++ Development Suite and Symantec C++.

The REC Library is comprised of two primary classes and several supporting structures. These classes are wrapped around the Crystal Report Engine API, providing an object oriented approach to programming the Crystal Report Engine.

NOTE: The REC Library can be used with any available version of the MFC Library.

The REC Library was installed in the same directory as Crystal Reports (\CRW by default). You need to add two files to your C++ project to use the REC Library:

PEPLUS.H

This C++ header file contains the Report Engine class definitions along with several structure definitions used by the Report Engine classes. You must #include this file in any code module that will be using the Report Engine classes.

PEPLUS.CPP

This C++ source code file contains implementations of all of the class methods for the REC Library classes. This file must be added to your C++ project file in order to use the REC Library. This code is compiled directly into your Visual C++ project and calls the functions available in the Crystal Report Engine.

The Report Engine API

As mentioned above, the REC Library classes are wrapped around the Crystal Report Engine API, and the methods in these classes make direct calls to the functions in the Report Engine API. For this reason, your project must include the CRPE32M.LIB Report Engine library. Make sure CRPE32M.LIB has been added to the list of libraries included with your C++ project.

The Report Engine Class Library classes

The REC Library consists of two primary classes. The CRPEngine class controls the entire Report Engine. It is designed so that there should only be one CRPEngine object in the entire application. The CRPEngine object contains methods that are common to all print jobs (e.g. SQL connections, version information, etc.). More importantly, it is responsible for creating and managing all CRPEJob objects.

The CRPEJob class controls print jobs. A print job is a request for a report to be processed and printed, previewed, or exported. You do not construct a CRPEJob object directly; instead, you request a job instance from the CRPEngine class and receive a pointer to a CRPEJob object. It is the CRPEJob object that allows you access to the attributes of a print job.

Both classes are derived from the MFC CObject class and provide all of the functionality of that class, including runtime class information and object diagnostic output.

NOTE: For complete descriptions of each of the REC classes, their respective members, and all supporting structures, refer to the Crystal Reports Developer's online Help.

The Report Engine API

The Report Engine API (REAPI) is the most direct method of adding the Crystal Report Engine to your application project. The Report Engine itself is a Windows Dynamic Link Library (DLL), and, therefore, exports its functionality in the form of DLL functions. These functions make up the Report Engine API.

The Report Engine DLL, CRPE.DLL (16-bit) and CRPE32.DLL (32-bit) is installed in your \WINDOWS\SYSTEM directory when you install Crystal Reports. This assures that the DLL is available to any application on your system that uses the Report Engine.

NOTE: For complete information on distributing Report Engine and other runtime DLLs with your application, refer to the Crystal Reports Runtime Help.

The process for loading a DLL and calling DLL functions is a well documented aspect of the Windows API. If you are not familiar with working with DLLs, please refer to Windows API documentation before attempting to use the Report Engine API. You may also want to consider one of the other methods described in this chapter for adding the Report Engine to your application.

The rest of this section assumes an understanding of DLLs and how to use them in a Windows application. It also assumes a basic understanding of the C language. The examples here are written in C, and do not cover the LoadLibrary, GetProcAddress, or FreeLibrary calls.

Many Windows development environments support direct calls to DLL functions, Visual Basic, Visual dBASE, and Delphi, for example. Refer to the documentation for your development environment for complete instructions on using a DLL. Your documentation may also cover instructions on how to translate C function calls to the language you use. Study your documentation, then review the steps described here for using the Report Engine in an application via the REAPI.

Declarations for the Report Engine API

Crystal Reports provides several source code files that declare the functions in the REAPI for several popular development languages. These files can be found in the Crystal Reports directory (\CRW by default) and are ready to be immediately

added to your project. The following REAPI declaration files are available:

- CRPE.H declares all Report Engine functions for C/C++.
- GLOBAL.BAS and GLOBAL32.BAS declare all Report Engine functions for Visual Basic.
- CRPEDB.H declares several Report Engine functions for Visual dBASE. Because of limits in the dBASE language, not all Report Engine functions are available to dBASE programmers. Refer to the individual function in Crystal Reports Developer's online Help for information on dBASE availability.
- CRPE.PAS and CRPE32.PAS declare all Report Engine functions for Delphi.

NOTE: You can declare functions yourself on an individual basis, but unless you will only be using a few of the Report Engine functions in your code, it is easiest to simply copy one of the previously mentioned code files into your project directory and add it to your project.

Using the Report Engine API

The REAPI provides two options for processing and producing reports from within an application:

- A Print-Only link, and
- A Custom-Print link.

The Print-Only link is the fastest, easiest method for producing a report with the REAPI. A Print-Only link, however, provides a very limited functionality. It allows a report to be printed on a default printer or previewed in a window on-screen. It does not allow you to customize a report in any way before printing it, though.

A Custom-Print link, on the other hand, provides all the report processing power of Crystal Reports itself. By coding a Custom-Print link into your application, you can change record selection, record sorting, group creation, group selecting, group sorting, exporting to disk files, e-mail, Exchange and Lotus Notes folders, or ODBC data sources, selecting specific printers for printing, logging on to SQL servers and ODBC data sources, editing formulas, formatting report sections, and much more. A Custom-Print link is, however, a more complex process to code than a Print-Only link.

The first time you use the REAPI in your application project, you may want to start by coding a simple Print-Only link to produce basic reporting functionality. As your project develops and you become more familiar with the REAPI, you can expand the reporting functionality with a Custom-Print link.

Establishing A Print-Only Link

A Print-Only link is performed using the PEPrintReport function. The PEPrintReport function provides basic report printing functionality and demonstrates basic techniques for calling Report Engine functions from your application.

PEPrintReport enables your application to print a report, to select the output device, either a default printer or a Preview window, and to specify the size and location of the Preview window if the report is printed to a window. This function does not enable you to customize the report (select the records to print, set the sort order, etc.) at print time. You can set those parameters at report design time (using the Crystal Reports Report Designer), but you can not change them at print time through a Print-Only link.

NOTE: If the report is sent to a Preview window, you should also use the PEOpenEngine and PECloseEngine functions with your Print-Only link. PEOpenEngine and PECloseEngine allow you to control how long the Preview window remains open. The window will remain open until the PECloseEngine function is called or the user clicks Close in the window. If PEOpenEngine and PECloseEngine are not used, and the report is sent to a Preview window, the window will automatically close as soon as the report finishes processing.

NOTE: You may also want to get in the habit of using PEOpenEngine and PECloseEngine in a Print-Only link, as they are required steps to coding a Custom-Print link. The rest of this section demonstrates these functions with a Print-Only link.

PeprintReport Arguments

PEPrintReport is declared in CRPE.H as follows:

```
short FAR PASCAL PEPrintReport (  
    char FAR *reportFilePath,  
    BOOL toDefaultPrinter,  
    BOOL toWindow, char FAR *title,
```

```
int left, int top,
int width, int height,
DWORD style, HWND parentWindow);
```

The following table describes each argument:

Parameter	Description
reportFilePath	The name of the report to be printed. Include the path if the report is not in the current directory. The report name can be hard-coded and unchangeable at runtime, or you can pass a string variable or character array as the result of a user choice.
toDefaultPrinter	If toPrinter is set to TRUE (1), the report is sent to a printer. The toWindow argument should be set to FALSE.
toWindow	If toWindow is set to TRUE (1), the report is sent to a Preview window. The toPrinter argument should be set to FALSE.
title	The title that you want to appear in the window title bar. This argument can receive a string variable or a character array at runtime.
left	The position, in screen coordinates, at which you want the left edge of the Preview window to appear if the report is being printed to a window.
top	The position, in screen coordinates, at which you want the top edge of the Preview window to appear if the report is being printed to a window.
width	The width, in screen coordinates, of your Preview window, in pixels, if the report is being printed to a window.
height	The height, in screen coordinates, of your Preview window, in pixels, if the report is being printed to a window.

<i>Parameter</i>	<i>Description</i>
style	The style setting (as defined in WINDOWS.H). Style settings can be combined using the bitwise OR operator. These are standard Windows styles. Refer to Windows API documentation for complete information on window styles. Use 0 for default style settings.
parentWindow	Specifies the window handle for the parent window to be used for this Preview window.

NOTE: If `toDefaultPrinter = True`, and if you have specified a printer in the report using the `PRINTER SETUP` command, `PEPrintReport` prints to the specified printer. Otherwise it prints to the Windows default printer. If you wish to override both the printer specified in the report and the Windows default printer, you will need to establish a Custom-Print link and specify the printer using the `PESelectPrinter` function.

NOTE: If `toPrinter = True`, you may enter null values for all of the remaining parameters except `reportFilePath` because they apply to printing to a Preview window only. The title parameter requires a null string, i.e., `""`, while the rest of the parameters will accept 0 (zero).

NOTE: If `parentWindow` is null, Crystal Reports creates a top level window. The top left corner specified is relative to the origin of the screen.

- If `parentWindow` is the handle of an MDI frame window, Crystal Reports creates a Preview window that is an MDI child window with the top left corner relative to the origin of the frame window's client area.
- If `parentWindow` is the handle of some other window, Crystal Reports creates a Preview window that is a child of that window with the top left corner specified relative to the origin of the parent window's client area.

NOTE: You can use the Windows constant `CW_USEDEFAULT` as the value of left, top, width, and height to put in the default settings.

NOTE: If the Preview window is a top-level window and the window style is defined as 0 (i.e., the final two parameters in the PEPrintReport call are 0,0) or, if the Preview window is an MDI child window and the window style is defined as 0, Crystal Reports uses the following default style:

```
(WS_VISIBLE | WS_THICKFRAME | WS_SYSMENU |  
WS_MAXIMIZEBOX | WS_MINIMIZEBOX)
```

That is, the default window is a visible window with a thick frame that can be used for sizing the window. The window includes a system menu box, and a maximize and minimize box.

Example code for a Print-Only link

The first step in actually accessing the Report Engine is to load it into memory. This can be done just before PEPrintReport is called, when a dialog box that allows printing opens, or even when your application first starts.

Once the Report Engine is open, PEPrintReport can be called as a result of some user action, such as clicking a button on screen, or some internal application procedure.

Finally, close the Report Engine by calling PECloseEngine when you are all finished with it. If you have several print jobs, do not close the Report Engine until all print jobs are finished. Opening and closing the Report Engine uses processor time and should only be performed when necessary.

The following C code demonstrates a possible message handler for an application that provides Print-Only link functionality through a button in a dialog box. Use this code as an example of how to perform a Print-Only link.

```
short result;  
  
switch (message)  
{  
  
    case WM_INITDIALOG:  
        if (!PEOpenEngine())  
            ; // Handle error  
        return TRUE;  
  
    case WM_DESTROY:
```

```

        PECloseEngine();
        return TRUE;
    case WM_COMMAND:
        switch (wParam)
        {
            case IDC_PRINTBUTTON:
                result = PEPrintReport (
                    "boxoffic.rpt",
                    FALSE, TRUE,
                    "My Report",
                    CW_USEDEFAULT,
                    CW_USEDEFAULT,
                    CW_USEDEFAULT,
                    CW_USEDEFAULT,
                    CW_USEDEFAULT,
                    hwndParent);
                if (result != 0)
                    return FALSE;
                return TRUE;
        }
        break;
    }
}

```

Establishing A Custom-Print Link

A more advanced, and more powerful, method of using the Crystal Report Engine is through a Custom-Print link. Establishing a Custom-Print link gives you a great deal of control over your reports at runtime. For example, you can:

- set or modify the report sort order,
- set or modify the record selection and/or group selection formulas,
- modify existing report formulas,
- set or modify the database location,
- capture and evaluate Report Engine errors,
- export a report to a file, e-mail, Exchange or Lotus Notes folder, or ODBC data source,
- log on to SQL servers and ODBC data sources,

- format report sections,
- and much more.

NOTE: The Report Engine allows you to add a selection formula and sort fields to a report at runtime, even if none existed in the report when it was designed. Report formulas created in the Crystal Reports Formula Editor, however, must be added when the report is created in Crystal Reports. A formula can be edited with the Report Engine, but can not be added to an existing report from the Report Engine. Design your reports carefully, and keep this in mind when you create your application.

Coding a custom-print link

There are six required steps to coding a Custom-Print link in your application. Each uses a different REAPI function. The steps are:

1. Open the Report Engine (PEOpenEngine)
2. Open a print job (PEOpenPrintJob)
3. Set the output destination (PEOutputToPrinter, PEOutputToWindow, or PEEExportTo)
4. Start the print job (PEStartPrintJob)
5. Close the print job (PEClosePrintJob)
6. Close the Report Engine (PECloseEngine)

In addition to these six steps, you can add several optional tasks any time after Step 2, opening the print job, and before Step 4, starting the print job. These optional tasks include changing selection formulas, editing report formulas, selecting export options, and sorting report fields.

Some REAPI functions can be called at special times to retrieve information about the print job or Report Engine. For example, PEGetVersion retrieves the current version of the Report Engine being used and can be called at any time, even without the Report Engine being open. Another example, PEGetJobStatus, can be called after Step 4 to obtain information about the current status of a job being printed. For more information on all REAPI functions, refer to the Crystal Reports Developer's online Help.

NOTE: The steps described here apply to a single print job. It is possible to have more than one print job open at once.

1. Open the Report Engine

Example

```
PEOpenEngine ();
```

Description

This step starts the Crystal Report Engine and prepares it to accept a print job. The Report Engine must be open before a print job can be established. You should open the Report Engine before the user has a chance to try to print a report. For example, if your application uses a dialog box as the User Interface to the Report Engine, open the Report Engine immediately after the dialog box is created at runtime. Your dialog box can allow the user to establish a print job and make changes to the report while the Report Engine is already open.

Every time the Report Engine is opened, it should be closed once your application is finished accessing it (see *Close the Report Engine*, Page 658). For example, if you open the Report Engine when a dialog is created, close the Report Engine when that dialog is destroyed.

2. Open a print job

Example

```
job = PEOpenPrintJob("BOXOFFIC.RPT");
```

Description

When you open a print job, the Report Engine returns a Job Handle for the print job. This handle is important to identifying the print job in the rest of your code.

To establish a print job, PEOpenPrintJob requires the path and name of the report that is to be printed. This argument can be hard-coded into the function call, as in the example above, or you can prompt the user to choose a report for printing and pass a variable argument to the function.

To close a print job, refer to *Close the print job* below. In most cases, you should open the print job immediately before printing and

close the print job as soon as the job is finished and the Preview window is closed or printing is complete.

3. Set the output destination

Example

```
PEOutputToWindow (job, ReportTitle,  
CW_USEDEFAULT, CW_USEDEFAULT,  
CW_USEDEFAULT, CW_USEDEFAULT, 0, NULL);
```

Description

The Report Engine must know where to send the final report. The report can be printed to a printer, displayed in a Preview window, exported to a disk file, or exported. The example above sends the report to the Preview window.

Although you can choose any of the several destinations for report output, you must establish a destination for the report to print. You can, however, write code in your application that allows your users to decide on a destination themselves.

NOTE: This step does not actually print the report, it only establishes a destination for the report when printed. The report is actually printed in Step 4 using the PESTartPrintJob function.

The following functions are available to establish a print destination:

- PEOutputToWindow
 - Printing a report to a window requires no other print destination code other than the function itself.
- PEOutputToPrinter
 - Printing a report to a printer requires no other print destination code other than the function itself. However, PESelectPrinter can be used to select a printer other than the default printer at runtime. The PESelectPrinter function uses the Windows structure DEVMODE. For more information on this structure, refer to the Windows SDK.
- PEExportTo
 - The PEExportTo function works with the PEExportOptions structure and several DLLs that control a report's export

destination and format. The information required by `PEExportTo` can be set in your code at design time or it can work with options in your application to allow a user to specify export destination and format. If you would like to allow your users to set the destination and format of a report file, but you do not wish to program the interface to do this, use the `PEGetExportOptions` function to have the Report Engine provide dialog boxes that query the user for export information at runtime. See *How to export reports*, Page 110.

4. Start the print job

Example

```
PEStartPrintJob(job, TRUE);
```

Description

This function actually sends the report to the output device indicated in Step 3. Once `PEStartPrintJob` is called, the Report Engine begins generating the report. The Report Engine displays a dialog that indicates the status of the report being generated. If the report is sent to the Preview window, the window will appear as soon as `PEStartPrintJob` is called. The Preview window can be closed by a call to `PECloseWindow`, by closing the Report Engine (as in Step 6), or by the user clicking the *Close* button. No changes can be made to a print job until it finishes printing.

5. Close the print job

Example

```
PEClosePrintJob(job);
```

Description

Once the print job has completed, it can be closed using `PEClosePrintJob`. If you wish to make more changes to the report and print it again, you can do so before closing the job. However, once your application is finished with a report, it should close the print job to free up memory in the user's system.

6. Close the Report Engine

Example

```
PECloseEngine();
```

Description

This function closes the Report Engine entirely. No other Report Engine functions relating to print jobs may be called once the Report Engine is closed. Therefore, you should keep the Report Engine open until it is no longer needed in your application. For example, if the Report Engine is accessed through a dialog box in your application, you should wait to close the Report Engine until the dialog box is exited and destroyed.

A sample Custom-Print link

The sample code below has been designed to demonstrate four of the six basic steps in establishing a custom-print link using the C programming language. This example is based on the following scenario:

- Using Crystal Reports, you have created a report called ORDER.RPT and saved it to the C:\CRW directory. This report is a listing of customer orders, and it is the only report your application will need to print.
- In your application, you have created a PRINT REPORT MENU command that opens a Print Report modal dialog box. The dialog box allows the user to select whether the report is printed to the printer or printed to a Preview window. If the report is to be printed to the Preview window, a Boolean variable called *ToWindow*, declared and initialized in another section of code not seen here, is given the value of TRUE. If the report is to just be sent straight to the printer, *ToWindow* is given the value FALSE.
- In the Print Report dialog box, there is also a *Print* button that initializes the event procedure to generate and print the report. The *Event code* section below demonstrates how the custom-print link can be coded in the *Print* button event procedure of your application.
- PEOpenEngine is called when the dialog box is created, and PECloseEngine is called when the dialog box is destroyed. For

this reason, these two steps are not included in the custom-print link that appears below.

The section titled *Event code* demonstrates the basic custom-print event procedure. This code includes *If* statements that check to see if an error has occurred during the call to the Report Engine. When an error occurs, you can easily handle the error in a separate routine or function. The event code below calls the function `ReportError` whenever an error occurs. `ReportError` is not a Report Engine function but is meant simply as an example of how to handle Report Engine errors. The code for `ReportError` appears in the section *Error code*.

Event code

```
short hJob;          // print job handle
BOOL bResult;

hJob = PEOpenPrintJob("C:\\CRW\\ORDER.RPT");

if (!hJob)
{
    ReportError(hJob);
    return;
}

if (ToWindow)
{
    bResult = PEOutputToWindow( hJob,
                                "My Report", CW_USEDEFAULT,
                                CW_USEDEFAULT, CW_USEDEFAULT,
                                CW_USEDEFAULT, 0, NULL );
}
else
{
    bResult = PEOutputToPrinter(hJob, 1);
}

if (!bResult)
{
    ReportError(hJob);
    PEClosePrintJob(hJob);
    return;
}
```

```

if (!PEStartPrintJob(hJob, TRUE))
    {
        ReportError(hJob);
    }

PEClosePrintJob(hJob);

return;

```

Error code

```

void ReportError(short printJob)
    {
        short        errorCode;
        HANDLE        textHandle;
        short        textLength;
        char          *errorText;

        errorCode = PEGetErrorCode(printJob);
        PEGetErrorText ( printJob,
                        &textHandle,
                        &textLength );

        errorText = (char*)malloc(textLength);
        PEGetHandleString(textHandle,
                        errorText,
                        textLength);

        MessageBox( hWnd, errorText,
                    "Print Job Failed",
                    MB_OK | MB_ICONEXCLAMATION );

        return;
    }

```

Code Evaluation

Event code

The following is an evaluation of the sample event code that appears above.

```

short hJob;        // print job handle
BOOL bResult;

```

This section declares two local variables that are important to the remainder of the code. The variable *hJob* will receive the handle to the print job that results from a `PEOpenPrintJob` call. This handle is required by most Report Engine functions. *bResult* will be given a TRUE or FALSE value as the result of several Report Engine calls. Any time *bResult* receives a FALSE value, an error has occurred.

```
hJob = PEOpenPrintJob("C:\\CRW\\ORDER.RPT");
```

This Report Engine call opens the new print job according to the path and filename of the report that is to be printed. In this example, the report name is hard-coded in the Report Engine call. A user would have no choice as to which report is printed. This function could also accept a character array or a pointer to a character array as an argument, allowing you to give your users the opportunity to choose a specific report for printing. `PEOpenPrintJob` returns a handle to the new print job, *hJob*. This handle will be used in all of the subsequent Report Engine calls shown here.

```
if (!hJob)
{
    ReportError(hJob);
    return;
}
```

This *if* statement verifies whether a valid print job handle was received in the previous line of code. If `PEOpenPrintJob` returned a value of 0, the print job is invalid and an error is reported. For more information on processing Report Engine errors, see the Error code section that appears below.

```
if (ToWindow)
{
    bResult = PEOutputToWindow( hJob,
                               "My Report", CW_USEDEFAULT,
                               CW_USEDEFAULT, CW_USEDEFAULT,
                               CW_USEDEFAULT, 0, NULL );
}
else
{
    bResult = PEOutputToPrinter(hJob, 1);
}
```

ToWindow acts as a Boolean variable that provides information from the user's decision as to whether this report will be printed to a Preview window or to a printer. If *ToWindow* holds a TRUE value, then the user has decided to print the report to a Preview window.

The *if else* code determines an output destination for the report based on the user's earlier decision. The *PEOutputToWindow* function prepares the Report Engine to create a Preview window while *PEOutputToPrinter* directs the Report Engine to print the report to the default printer. (The printer used by the Report Engine can be changed with the *PESelectPrinter* function.) The variable *bResult* receives a FALSE value if an error occurs in either function call.

```
if (!bResult)
{
    ReportError(hJob);
    PEClosePrintJob(hJob);
    return;
}
```

Once the appropriate destination function is called, we must verify its success and report an error if *bResult* is FALSE. *ReportError* is our error handling routine. It is an internal function designed to process any errors that occur during a print job. The function is passed the current value of the *hJob* handle for use in analyzing errors. See the Error code section below for information on processing errors.

NOTE: ReportError is not a Report Engine function. It is specific to the code that appears here and is meant only as an example of how to handle Report Engine errors.

Since a print job has been opened, we must close it after the error is reported using *PEClosePrintJob*. See below for more information on this function. Finally, the *if* statement causes a return after the error has been reported, thus ending the print job session.

```
if (!PEStartPrintJob(hJob, TRUE))
{
    ReportError(hJob);
}
```


PEStartPrintJob actually sends the print job to the printer or a Preview window. If the report is printed to a window, PESTartPrintJob creates and opens the window according to the parameters set in the PEOutputToWindow function. If PESTartPrintJob fails (returns FALSE), an error is reported.

```
PEClosePrintJob(hJob);
```

Once the report has printed, this print job can be closed and another one can be started if needed. If the report has been printed to a Preview window, PEClosePrintJob does not close the window. The Preview window is closed when the Close button is clicked, the PECloseWindow function is called, or the PECloseEngine function is called.

```
return;
```

Now that the print job has finished, the event procedure can return, and the application can wait for the next user event to occur.

Error code

```
void ReportError(short printJob)
{
```

Report Engine error processing can be most efficiently handled by a separate internal function that is called during a print job. The Event code evaluated above calls the ReportError function. The code for that function appears here as an example of how to access and evaluate Report Engine errors. The error number returned by PEGetErrorCode can be used to control how your application reacts to different types of Report Engine errors.

NOTE: The REAPI functions described here, PEGetErrorCode and PEGetErrorText, are specific to REAPI error handling. For complete descriptions of these functions, refer to the Crystal Reports Developer's online Help. The function PEGetHandleString is used to retrieve variable length strings generated by different REAPI functions. For more information on handling REAPI variable length strings, see Page 665.

```
short      errorCode;
HANDLE    textHandle;
short     textLength;
char      *errorText;
```

Completely processing any Report Engine error requires at least four variables like those above. While only `errorCode` will be needed to retrieve the Report Engine error number, the other three variables will all be needed to retrieve the actual error text.

```
errorCode = PEGetErrorCode(printJob);
```

`PEGetErrorCode` returns a number associated with the error that has occurred. For a list of these error codes and their meanings, please see *Report Engine Error Codes* in the *Crystal Reports Developer's online Help*.

```
PEGetErrorText ( printJob,
                 &textHandle,
                 &textLength );

errorText = (char*)malloc(textLength);

PEGetHandleString(textHandle,
                 errorText,
                 textLength);
```

The error text must be returned in the form of a handle to a variable length string. The handle is used, along with the `PEGetHandleString` function to obtain the actual error text and store it in a character array. This is a complicated process, and it should be examined carefully if your code is to work. See *Report Engine API Variable Length Strings*, Page 665.

```
MessageBox( hWnd, errorText,
            "Print Job Failed",
            MB_OK | MB_ICONEXCLAMATION);
```

Once the error has been obtained, we can display error information to the user. This example simply opens a warning message box to alert the user of the problem. Using the error code and the error text, however, you can control Report Engine error messages any way that you find appropriate for your application.

```
return;
}
```

Once error processing is finished, we can return to processing the print job. If an error has occurred during the print job, however, then the print job should be terminated immediately after the error is processed. Review the evaluation of the event code above for ideas on how to terminate a print job after an error.

Report Engine API Variable Length Strings

Several REAPI functions provide information in the form of a variable length string value or character array. When your program calls an REAPI function that produces a variable-length string, the Report Engine saves the string, creates a string handle which refers to the string, and returns that handle along with a value indicating the length of the string. To retrieve the contents of the string, you must call *PEGetHandleString*. This approach allows you to allocate a buffer of the exact size needed to hold the string before obtaining the actual string.

If your development language cannot allocate a buffer at runtime, you should declare a reasonably large buffer. Field names and error messages will generally be less than 100 bytes, but formulas may be 1000 bytes or longer. You can control how much data is copied to the buffer when you call *PEGetHandleString*.

Here is the procedure to follow when obtaining a variable length string:

1. Call the function which produces the string. This returns the string handle and length. The length includes all characters in the string plus a terminating null byte.
2. If necessary, allocate the string buffer.
3. Call *PEGetHandleString* to copy the string from the handle into the buffer.

NOTE: *PEGetHandleString* frees the memory occupied by the string handle, so you can only call this function once for a given handle.

NOTE: For experienced Windows programmers: text and name handles are Global Memory Handles for memory segments on the global heap. If you prefer, you can access these segments using the Windows *GlobalLock*, *GlobalUnlock*, and *GlobalFree* functions. Contents of name and text handles are null terminated ASCII strings. You must free the text handle with *GlobalFree* when you are done with it. (*PEGetHandleString* does this for you, if you use it.)

Sample Code

Use the following C code as an example of how to call a function that returns a variable length string. The code uses the

PEGetNthSortField function which obtains the name of a field being used to sort the report and the direction of the sort. There are several other functions that return variable length strings, but all are handled in a similar fashion.

Examine this code carefully and try to incorporate it into your own application without modifying the basic procedure. Only experienced programmers should try making changes to this technique since small mistakes here can cause major errors in your application. If you expect to use several REAPI functions that return variable length strings, you may want to set this code up in a separate function to avoid repetition and errors.

```
HANDLE    nameHandle;
short     nameLength;
short     direction;
char      *fieldName;

PEGetNthSortField ( printJob, sortFieldN,
                    &nameHandle, &nameLength,
                    &direction );

/*allocate fieldName buffer*/
fieldName = (char*)malloc(nameLength);

PEGetHandleString ( nameHandle,
                    fieldName,
                    nameLength );

/*
** fieldName now contains name
** of field and nameHandle is no
** longer valid.
*/
```

NOTE: If you retrieve a string handle but don't retrieve the string itself (i.e., you do not use `PEGetHandleString`), you should free up the string memory by calling `GlobalFree (nameHandle)`.

Code Evaluation

```
HANDLE    nameHandle;
short     nameLength;
short     direction;
char      *fieldName;
```

Any time you evaluate a function that returns a variable length string, you will need at least three variables: a handle to the string, a short integer to hold the length of the string, and a character array or pointer to a character array. The direction variable in this example will hold the sort direction and is specific to the PEGetNthSortField function.

It is important to note that although the PEGetNthSortField function is defined in the Report Engine as accepting a pointer to a handle (HANDLE*) and a pointer to a short (short*), we do not define nameHandle and nameLength as pointer variables. Instead, they are defined simply as a HANDLE and a short integer, then passed to PEGetNthSortField with the & operator. This technique automatically initializes the variables with the address of the variable itself. Since the PEGetNthSortField function requires the address in memory to place the information, this is the most convenient method to define and pass the variables.

```
PEGetNthSortField ( printJob, sortFieldN,  
                  &nameHandle, &nameLength,  
                  &direction );
```

The PEGetNthSortField function places a handle to the sort field name in the nameHandle location and the length of the field name (all characters in the name plus a terminating null byte) in the nameLength location. These values will be used to extract the actual field name.

```
/*allocate fieldName buffer*/  
fieldName = (char*)malloc(nameLength);
```

Now that we know the actual length of the field name we are trying to obtain, we can allocate exactly the right amount of memory to store that name. The malloc function does this for us.

NOTE: malloc is defined in the C runtime library *stdlib.h*.

```
PEGetHandleString ( nameHandle,  
                  fieldName,  
                  nameLength );
```

PEGetHandleString uses the string handle to retrieve the field name and store it in *fieldName*. At the same time, *nameHandle* is invalidated. Now, the text can be used like any other character string.

NOTE: This code is meant as a basis for your own code. Although these elements shown here are necessary for extracting a variable length string from certain Report Engine functions, experienced programmers may wish to expand the code to trap errors or handle the string text differently.

The following is a list of all of the REAPI functions that return variable length strings:

- PEGetNthSortField
- PEGetNthGroupSortField
- PEGetFormula
- PEGetSelectionFormula
- PEGetGroupSelectionFormula
- PEGetNthParam
- PEGetSQLQuery
- PEGetErrorText

Sample Code

There are many topics in the Crystal Reports Developer's online Help that provide sample code for using the Report Engine API in C, Visual Basic, Visual dBASE, and Delphi.

- All topics for individual REAPI functions contain sample calls for Visual Basic, C, Delphi, and Visual dBASE. (Locate by function name in the Crystal Reports Developer's online Help.)

Other topics that contain useful example code are:

- *Establishing a print-only link using PEPrintReport* contains sample print-only link calls for C.
- *Coding a custom-print link* contains C code which can be modified for a custom-print link Report Engine call.
- *Functions that return variable length strings* contains a block of C code that you can modify and use for calling a function that returns a variable length string.

Report Engine API Structures

Several REAPI functions require a structure or user-defined variable type to be passed as one or more arguments. Some of these functions require that you assign values to all members of

the structure before calling the function so that the information can be used to make various settings in the Report Engine. Other functions require only the size of the structure be assigned to the `StructSize` member. These functions fill in the rest of the structure members for you, providing you with valuable information about a print job.

NOTE: The term structure is used here to mean both C structures and other user-defined types or records in languages such as Visual Basic and Delphi. If you are unfamiliar with this type of data, refer to the documentation for the programming language you are using.

Each structure used by REAPI is defined and explained in the Crystal Reports Developer's online Help with link to the function that uses it. Functions that use structures also have hypertext links to the structure definitions.

Some of the structures, `PEExportOptions` for example, are complex, requiring other structures be passed as member values. Not all programming languages support this feature. If you are using a programming language that does not allow the use of a structure variable as a member variable defined inside other structures, declare the member variable as another data type, such as an integer or a variant data type, and assign it a value of 0 (zero) at runtime. The Report Engine will automatically provide default values or will request information from the user.

NOTE: Structure variables cannot be created using Visual dBASE. Report Engine functions requiring structures as parameters are not available to dBASE.

Distributing Report Engine Applications

Crystal Reports comes with a free runtime license for any application that uses the Crystal Report Engine through any of the methods described in this chapter. When distributing a Report Engine application, you must also distribute several runtime files required by the Report Engine. These files are listed in the Runtime Files Help. Be sure to carefully examine this Help file and distribute the appropriate runtime files with your application. All runtime files are included under the runtime license agreement unless otherwise stated.

Additional Sources of Information

In addition to the information provided in this chapter, you will find a wide-variety of developer topics in Crystal Reports Developer's online Help. Many of these topics contain sample code in C, Visual dBASE, Delphi, and Visual Basic that you can copy directly into your application. To get a listing of all developer topics, see Crystal Reports online Developer's Help.

A

Report Processing Model

What you will find in this appendix...

Multi-pass Reporting 672

Multi-pass Reporting

What is a "Pass"?

A pass is a process that Crystal Reports uses each time the data is "read" and manipulated. Therefore, if a report reads and manipulates the data twice it is considered to be a 2-pass report. This feature is one that is very powerful in the data access and reporting industry because it allows for complex reporting and formula manipulation. Therefore, percent of subtotal calculations are possible due to the 2-pass reporting capabilities of Crystal Reports. Some reports can be 1-pass yet in most cases 2-passes will be issued. The following section offers a brief overview of the 2-pass reporting concept. Following the written description is a visual representation of this process.

Pre - Pass #1

When previewing a report the first elements to be evaluated are "flat" formulas. Flat formulas are those that do not contain database fields. For example, $100 * 30$ would be a flat formula. Flat formulas are evaluated at the beginning of the Print Preview process and are never evaluated again. This process is known as "BeforeReadingRecords". If you were to drop a flat formula field (i.e. $100 * 30$) in the details section, the result would be 3000 for each record displayed.

Pass #1

Once the "BeforeReadingRecords" process has taken place, Crystal Reports will begin reading the database records. During the record reading process the following will occur:

1. Record Retrieval.
2. Evaluate "recurring" formulas. These formulas are those that contain database fields but do not contain references to subtotals or summary information that would require data manipulation in the second pass. This evaluation time is known as "WhileReadingRecords".
3. Apply the record selection criteria. If the selection criteria is based on a database field that is indexed (i.e. `{company.LASTNAME} = "SMITH"`) then Crystal Reports rejects records not equal to SMITH immediately after evaluating the recurring formulas. The reason for this is that

the selection criteria may include recurring formulas. (i.e. {table.FIELD} = {@Formula}).

4. **Totaling.** A typical report usually contains groups, sorts and subtotals and Crystal Reports tries to process as many of these in the first pass as possible. As the records are processed they are divided into groups based on the group field specified in the report. Each of these groups is sent to an internal "totaler". This counting mechanism stores a subtotal of each group, in memory, which is then used later in the report process.
5. **Store the saved records.** After the totaling process is complete all of the records and totals are stored into a "saved records" object. This object stores data in memory and on disk in the form of temporary files. Saved records are used during the report's second pass for 2-pass calculations, group sorting, etc. Therefore, the second pass of the report does not actually read the database again, instead it uses the saved records object.

Steps 1 to 5 is an iterative process that will repeat for each record being read.

TopN/ BottomN/ Group Sorting

TopN/BottomN is a process that allows a user to select either the TopN or BottomN groups. N is the number you specify. Before you can use the TopN feature, your data must be subtotaled or summarized. The Group sorting allows you to specify the order in which your "groups" are printed. This sort order is based on the subtotaled or summarized field, not the grouping field.

This process, an intermediate step between passes, actually occurs in between the first and second pass of the report process and does not actually require the records to be read. Instead it only looks at the grouping information stored in the Saved Records Object and orders the groups as specified.

Pass #2

After completing the TopN/Group sort process, Crystal Reports enters into a second pass of the data. This means that Crystal Reports will look at the Saved Records Object for the current information and continue with the following elements of Pass #2.

1. **Read records that are contained in the Saved Records Object.** These records are read one at a time along with their respective subtotals.

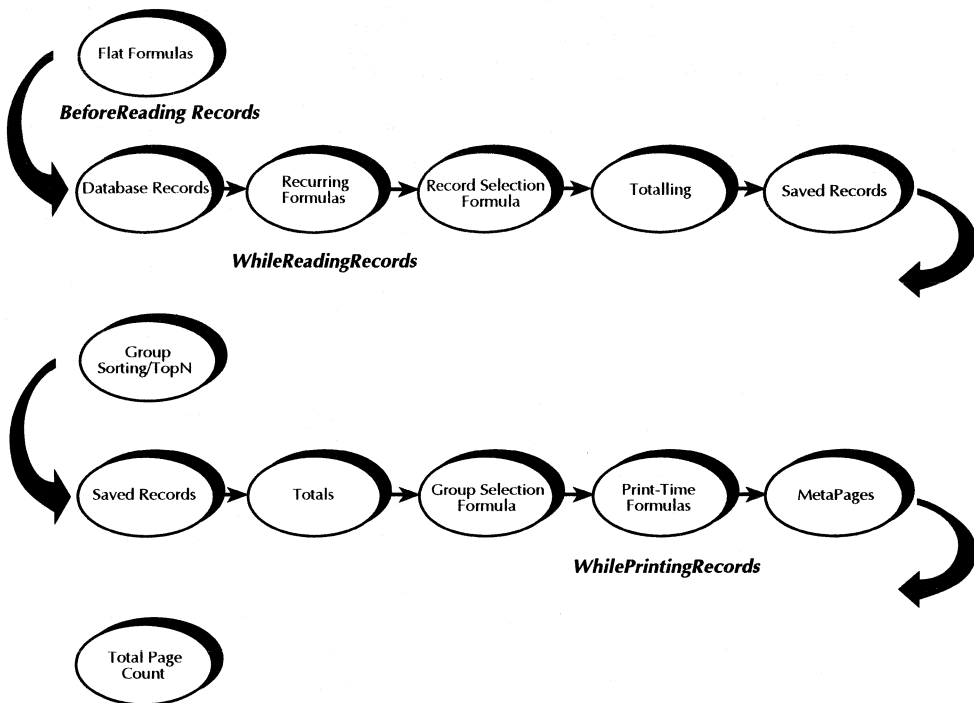
2. Once the subtotal or summary information is available the Group Selection Formula can now be applied. The group selection formula allows the user to select groups based on the subtotaled or summary field not the grouping field. For example, you may only want the groups that have a subtotal greater than X.

HINT: The Group Selection Formula dialog can be used for Record Selection formulas, however it is not recommended. If, for example, you used a typical selection such as:

`{company.LASTNAME} = "SMITH"`

in the Edit Group Selection dialog box, the records that are displayed on your report may be correct, however, the subtotals, summaries and grand totals will most likely be incorrect. The reason is that all of the subtotal/grand total information is calculated in Pass #1 and therefore, if you decide to filter out records in Pass #2 the subtotals will not be modified accordingly.

3. Evaluation of Print Time Formulas. This process is known as "WhilePrintingRecords". This would include formulas that have been explicitly defined as "WhilePrintingRecords" in the formula itself as well as formulas that refer to subtotals or summary fields. Examples of print-time formulas are:
 - % of Subtotals
 - Running Totals or Running Averages
 - Formulas explicitly marked "WhilePrintingRecords"
4. MetaPage Generation. These pages are generated to display your report to screen. MetaPages are similar to a standard Windows Meta File. Essentially each page is a "recording" of the individual report pages. Therefore, all of the lines, boxes, fields, etc., are stored in the MetaPages. This method of storing report pages is much more efficient than storing the pages as bitmaps. For example, the following diagram is a flow-chart of the multi-pass report process:



B

Product Support, Upgrades, and Licensing

What you will find in this chapter...

Product Support 678

Fax On Demand 678

Mail-in Support - FREE 679

Fax Support - FREE 679

Telephone Support 680

Extended Technical Support Policy 681

Product Registration 681

Crystal Reports Upgrade Plan 682

Product Return Policy 682

Product Replacement Policy 683

License Agreement And Limited Warranty 683

Restricted Rights Legend 688

Product Support

Crystal Reports is a very powerful report writer for the Windows environment. Seagate Software IMG, Inc. is proud of the quality of the product and has spent a great deal of time trying to make it intuitive to use. If, however, there is something you cannot figure out how to do, we suggest you consult the extensive Crystal Reports online Help System. The Help System can be accessed by clicking on any Help button, by pressing the F1 key, or by choosing the CONTENTS command from the main Crystal Reports Help menu. The help system contains most of the information from your User's Guide and includes numerous examples. If you still cannot find an answer to your question, we suggest you consult your User's Guide. That guide includes an extensive index to help you find the topic of interest. Should you have questions that cannot be answered through the use of help or the User's Guide, the following alternatives are available for contacting Seagate Software IMG, Inc. directly.

NOTE: Product support plans vary from region to region. Contact your local distributor for a list of product support plans available in your region. See README.HLP for a list of International distributors.

Fax On Demand

If you have a fax machine, you can get technical and marketing documents from Seagate Software IMG, Inc. by calling (604) 681-3450. Listen to the menu, select the documents you would like to receive, and enter your fax number. Seagate Software IMG, Inc. will fax the documents to you automatically.

NOTE: Product support plans vary from region to region. Contact your local distributor for a list of product support plans available in your region. See README.HLP for a list of International distributors.

Seagate Fax On Demand (604) 681-3450

Fax Support - FREE

Another efficient way to receive support on Crystal Reports is to fax in your technical support request. To do this, first fill in the bottom part of the Technical Support Request form in the product. This form appears when you choose the TECHNICAL SUPPORT REQUEST command from the Crystal Reports Help menu. When you have completed the form, click the *Print* button to print it.

Fax the completed form to us 24 hours a day, Monday through Saturday. After a technical support representative has had a chance to review your fax, one of our support representatives will respond to you by return fax.

NOTE: Product support plans vary from region to region. Contact your local distributor for a list of product support plans available in your region. See README.HLP for a list of International distributors.

Seagate Fax Support (604) 681-7163

Mail-in Support - FREE

If you would prefer, you can also contact us by mail with written questions or comments, and we will respond by return mail.

If you believe that your questions are such that we should review all of your related files (such as databases, etc.) to help solve your problem, then copy all of the related files and the Crystal Reports report file (FILENAME.RPT) onto a diskette of any PC compatible format and send it to us. We will investigate the problem and mail back a response to you as soon as possible with your original diskette.

NOTE: Product support plans vary from region to region. Contact your local distributor for a list of product support plans available in your region. See README.HLP for a list of International distributors.

Our mailing address is:

Seagate Software IMG, Inc.
4th Floor - 1095 West Pender Street
Vancouver, BC, Canada V6E 2M6

Telephone Support

Registered users of Crystal Reports are entitled to free telephone support (subject to availability) for 60 days from the time of purchase. Telephone support is available from 8 to 5, P.S.T. If you are calling from out of the Vancouver, B.C. area, long distance charges apply.

Before you call technical support, make sure you do the following:

- check the manual
- check help
- check FOD (fax on demand)
- check the Internet

PLEASE HAVE THE FOLLOWING INFORMATION AVAILABLE:

- serial number,
 - If you are registered, you can find the serial number by choosing the ABOUT CRYSTAL REPORTS command on the Help menu.
 - If you are not registered, you will need to register first to obtain your serial number.
- product name and version number,
- operating system you are using (i.e., Windows 95, Windows NT or Windows 3.1) and whether you have a 16-bit or 32-bit version of Crystal Reports,
- version of database and other software you are using with Crystal Reports (if required) - Btrieve, Sybase, Paradox, etc.,
- Technical Support Request form from the product,

- network information if you are on a network,
- contents of autoexec.bat and config.sys files,
- a list of steps necessary to recreate the problem,
- the database type you are using, and
- the programming environment (if applicable).

NOTE: Product support plans vary from region to region. Contact your local distributor for a list of product support plans available in your region. See README.HLP for a list of International distributors.

Telephone Number (604) 669-8379

Extended Technical Support Policy

Registered users may purchase an extended support policy which entitles the user to unlimited technical support for a one year period. Support plan members will receive a special phone number which they can use to get directly through to the Technical Support team. Call for detailed information about this plan.

NOTE: The extended technical support policy is available in the United States and Canada only.

Product Registration

When you use Crystal Reports for the first time, it will ask for your name, address, and related information. Then the product will suggest that you register Crystal Reports with us to receive your product Serial Number. The program will ask for the Serial Number the next time you run Crystal Reports. When you enter the Serial Number, you have completed the registration process.

You can register Crystal Reports using three different methods:

1. Fill out the Registration form that is built into Crystal Reports and then register by modem using the Crystal Reports communications program. The procedure will register your copy of the program, assign a serial number, and enter that number automatically into your system.

2. Print the Registration Form that is built into Crystal Reports and then fax it to us at (604) 681-5147. We will then fax back to you a Serial Number that can be entered into the product the next time it asks for it.
3. Fill out the enclosed Registration Form and mail it to us. We will mail a Serial Number back to you so you can enter it into the product.

Registering Crystal Reports will ensure that you are kept up to date with all product advancements, and it will allow us to provide quality technical support to people that are properly registered with Crystal Services.

Registration FAX (604) 681-5147

Crystal Reports Upgrade Plan

If you would like to automatically receive upgrades to Crystal Reports, you can purchase Crystal Reports Upgrade plan. This will entitle you to receive free upgrades for one year for only \$150.00 per year. You will automatically get upgrades shipped by UPS ground. (You can receive them by air, if you wish, at an additional expense.) This upgrade plan includes all product upgrades as well as any documentation changes during the year. If you want worry free software maintenance, then this is the plan for you.

NOTE: The upgrade plan is available in the United States and Canada only.

Order Phone Line (604) 681-3435

Order FAX line (604) 681-2934

Product Return Policy

If you are not satisfied with Crystal Reports for any reason, you can return it to the original place of purchase for a refund within 30 days of the purchase date.

Product Replacement Policy

If the Crystal Reports diskettes or documentation are defective, then please contact Seagate Software Information Management Group, Inc. within 30 days of the purchase date. Fax the description of the problem and we will solve it as quickly as possible. Please fax the description of the defect to (604) 681-2934.

NOTE: Product support plans vary from region to region. Contact your local distributor for a list of product support plans available in your region. See README.HLP for a list of International distributors.

License Agreement And Limited Warranty

IMPORTANT - READ CAREFULLY

This License Agreement and Limited Warranty (herein referred to as "Agreement") is a legal agreement between you (either an individual or a single entity) and Seagate Software Information Management Group, Inc. (herein referred to as "Seagate Software") for the Crystal Reports software product, which includes computer software and associated media, printed materials, and "on-line" or electronic documentation (herein referred to as "Software Product"). By opening the sealed software package and / or by installing, copying or otherwise using the Software Product, you agree to be bound by the terms of this Agreement. If you do not agree to the terms of this Agreement, promptly return the Software Product and accompanying items in resalable condition to the place from which you obtained them for a full refund.

1. LIMITED LICENSE.

- 1(a). You are purchasing only the right to use a single copy of the Software Product, and are acquiring no rights to the software itself, or the enclosed copy, except the Limited License granted herein.
- 1(b). Except as stated in 1(c), Seagate Software grants you a Limited License to use one copy of the Software Product on a single computer. You may access the Software

Product from a hard disk, over a network (subject to 1(c)) or by any other method, strictly in accordance to the terms hereof.

- 1(c). For additional users to use the Software Product on a network, you must acquire "Multiple User Packs", which allow additional designated users in numbers and on terms specified by the documentation accompanying the Software Product. You may have as many copies of the Software Product in use on a network as is specifically authorized in the documentation.
2. COPYRIGHT. All title and copyrights in and to the Software Product (including but not limited to, any images, photographs, animations, video, audio, music, and text incorporated into the Software Product), the accompanying printed materials, and any copies of the Software Product are owned by Seagate Software or its suppliers. The Software Product is protected by copyright laws and international treaty provisions. Therefore, you may not copy the Software Product except that you may either (a) make one copy of the Software Product solely for backup or archival purposes, or (b) install the Software Product on a single computer provided you keep the original solely for backup or archival purposes. You may not copy the printed materials accompanying the Software Product.
3. OTHER RESTRICTIONS.
 - 3(a). Rental. You may not rent or lease the Software Product.
 - 3(b). Software Transfer. Unless this is a "Not for Resale" or "Evaluation" product, you may permanently transfer all of your rights under this Agreement, provided that you retain no copies, you transfer all of the Software Product (including all component parts, the media and printed materials, any upgrades, and this Agreement.), and the recipient agrees to the terms of this Agreement. If the Software Product is an upgrade, any transfer must include all prior versions of the Software Product.
 - 3(c). Export. Any export of the Software Product or its technical data from the United States may be restricted by U.S. export control laws, and accordingly you must comply with such laws before engaging in any such

export.

- 3(d). Limitations on Modifications or Reverse Engineering. You may not alter, disassemble, decompile, or reverse-engineer the Software Product.
- 3(e). No Separation of Components. The Software Product is licensed as a single product, and neither the software programs comprising the Software Product or any update or upgrade may be separated for use by more than one user at a time.
- 3(f). Termination. Without prejudice to any other rights, Seagate Software may terminate this Agreement if you fail to comply with the terms and conditions of this Agreement. In such event, you must destroy all copies of the Software Product.

4. LIMITED WARRANTY.

- 4(a). Seagate Software warrants the physical disk(s) and physical documentation to be free of defects in materials and workmanship for a period of 60 days from the purchase date. If Seagate Software receives notification within this warranty period of any such defects and such notification is determined to be correct, Seagate Software will replace the disk(s) or documentation.
- 4(b). Seagate Software's entire liability for any breach of this warranty is limited to replacement of defective disk(s) or documentation, and does not include or extend to claims for any other damages.
- 4(c). SEAGATE SOFTWARE SPECIFICALLY DISCLAIMS ALL OTHER CONDITIONS AND WARRANTIES, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED CONDITION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

5. LIMITATION OF LIABILITY. IN NO EVENT SHALL SEAGATE SOFTWARE BE LIABLE FOR COSTS OF SUBSTITUTE PRODUCTS, LOSS OF PROFITS, DATA OR USE OF THE SOFTWARE PRODUCT, OR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OR

OTHER SIMILAR CLAIMS, HOWEVER CAUSED, AND ON ANY THEORY OF LIABILITY, EVEN IF SEAGATE SOFTWARE HAS BEEN SPECIFICALLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL SEAGATE SOFTWARE'S LIABILITY FOR ANY DAMAGES TO YOU OR ANY OTHER PERSON EVER EXCEED THE LOWER OF THE SUGGESTED LIST PRICE OR THE ACTUAL PURCHASE PRICE PAID FOR THE SOFTWARE PRODUCT, REGARDLESS OF THE FORM OF THE CLAIM. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY. ALL THIRD PARTY SUPPLIERS TO SEAGATE SOFTWARE OF ANY PORTION OF THE SOFTWARE PRODUCT MAKE NO WARRANTY AND SHALL HAVE NO LIABILITY TO YOU.

6. **RUNTIME SOFTWARE.** The software files listed in the software file **runtime.hlp**, provided on your software disk(s), individually, collectively or in any combination, constitute Runtime Software. Unless this is a "Not for Resale" or "Evaluation" product, Seagate Software grants you a Limited Runtime License to reproduce and distribute exact copies of the Runtime Software, and to modify, reproduce and distribute the sample applications made from the original disks in this package (or a set of backup copies made directly from them), if and only if all of the following conditions are satisfied:
 - 6(a). you may distribute such copies solely for the purpose of executing specific-purpose application programs written using an authorized copy of Software Product;
 - 6(b). you remain solely responsible for support, service, upgrades, and technical or other assistance, required or requested by anyone receiving such copies;
 - 6(c). you may not use the name, logo, or trademark of Seagate Software or the Software Product without written permission from Seagate Software;
 - 6(d). you may not alter, disassemble, decompile, or reverse-engineer the Runtime Software;
 - 6(e). Seagate Software makes no warranty other than the Limited License warranty provided in this agreement to you; and

- 6(f). you will indemnify Seagate Software against any claims or liabilities arising out of its use, reproduction or distribution, and the use by the recipients or copies made thereunder.

You may not include the Runtime Software or sample applications in any general-purpose report writing and/or database querying product, or any other product that is generally competitive with Seagate Software product offerings. If you wish to distribute copies of the Runtime Software or sample applications on other terms, you must obtain written permission from Seagate Software.

"Not for Resale" and "Evaluation" products do not include a Limited Runtime License, and you may not, therefore, distribute Runtime Software or sample applications.

The Software Product contains software components from Seagate Software suppliers. This Agreement does not grant you the rights to distribute software components from these suppliers, including but not limited to, Microsoft Corporation's "JET Files", unless you have proper authorization from the appropriate supplier.

7. **GOVERNING LAW.** If you acquired this product in Canada, this Agreement is governed by the laws of the Province of British Columbia, Canada, without regard, and each party irrevocably attorns to the exclusive jurisdiction of the courts of the Province or British Columbia. If you acquired this product in the United States or elsewhere other than Canada, this Agreement is governed by the laws of the State of California, without regard, and each party irrevocably submits to the exclusive jurisdiction of the courts of the State of California.
8. **MISCELLANEOUS.** This document constitutes the entire agreement of the parties regarding the subject matter herein. All rights not expressly granted to you are retained by Seagate Software.

Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. Use, duplication, reproduction, or transfer of this commercial product and accompanying documentation is restricted in accordance with FAR 12.212 and DFARS 227.7202 and by this license agreement. Contact Legal Department, Seagate Software, Inc. (408) 456-4646.

Glossary

Absolute Formatting	Formatting that is always applied to an object. See also <i>Conditional Formatting</i> .
Access	To access data means to retrieve data.
Acrobat Reader	Application used to review and print the Online User's Guide and Developer's Guide.
ActiveX Control	Custom Control for Visual Basic 4.0 that incorporates the Object Linking and Embedding (OLE) technology. Formerly known as an OLE Control (OCX).
Active database	An active database is a database that has been selected for use in a report. You activate databases via the New command in the File menu and the Add Database to Report command from the Database menu.
Aggregate functions	An operation that summarizes data (sums it, calculates an average, identifies a maximum value, and so forth). The term "Aggregate functions" is often associated with SQL data sources.
Alias	In Crystal Reports, an alias is an alternative name assigned to a database. If a database is called customer.db, you could assign the alias customer, cust, company, DB1, or any other name that suits your needs. Aliases make it easier for you to use a report created with a database whose name and/or location has changed since the report was created.
Area	An area is a group of like sections (i.e., Details A and Details B) that all share the same characteristics but can be formatted differently.

Argument

An argument is an item, or one of a group of items, that receives the action of a function. It provides information that the function needs in order to operate. The Truncate function, for example, cannot operate by itself. It needs an argument that identifies the item to be truncated. Thus, in the formula:

```
Truncate ({orders.ORDER AMOUNT})
```

where Truncate is the function and {orders.ORDER AMOUNT} is the argument, it is the value of the {orders.ORDER AMOUNT} field that is the item to be truncated.

Array

An array is a group of values, separated by commas. Arrays are used with a variety of Crystal Reports functions: Average([array]), Maximum([array]), etc. In these functions, the array is the argument for the function. The function works on the items in the array. Items in an array can be constants, data fields, or formula results.

Arrowhead

A symbol to show that a field is indexed.

Attribute

An attribute is a quality applied to an object (i.e., font size, color, and so forth).

Auto arrange

When you place a field on your report, Crystal Reports allocates a space equal to the field width as specified in your database. Often that field width is far larger than the values that actually appear in each of the fields. For neat looking reports, it is often necessary to resize the fields so the space allotted more closely matches the size of the field values. Once you have resized the fields you often need to reposition them for proper balance.

Auto Arrange will do this for you. Simply choose the AUTO ARRANGE REPORT command from the Format menu.

Bitmap

A graphic file that can be added to a report.

BLOB field

A BLOB field is a field containing BLOB data. A BLOB (Binary Large Object) is simply a bitmapped graphic that has been entered into a database. Placing a BLOB field on your report allows you to access these graphics as you would other data types.

Boolean expression

A Boolean expression is an expression that defines a logical relationship between two or more items. A Boolean expression is either TRUE or FALSE. $A > 5$ and $B < 10$ is a Boolean expression that uses the Boolean operator And. For the expression to be TRUE, both conditions (joined with the And operator) must be true. The value of A must be greater than 5 and the value of B must be less than 10. If the values do not fall into those ranges, then the expression is FALSE. Boolean expressions are useful in If-Then-Else formulas. For example, If $A > 5$ and $B < 10$ then "In Range" Else "" is a formula that says, if the Boolean expression $A > 5$ and $B < 10$ is TRUE, print "In Range" otherwise if the Boolean expression is FALSE print nothing (as designated by the empty string "").

Boolean formulas

Boolean formulas are formulas that return a Yes/No (TRUE/FALSE) value. For example, the Boolean formula $\{orders\ detail.QUANTITY\} > 6$ compares the value in the $\{orders\ detail.QUANTITY\}$ field to 6. If the value is greater than 6 it returns a Yes; if it is 6 or less, it returns a No. Contrast this with a non-Boolean formula like $\{orders\ detail.QUANTITY\} * 6$. In this case Crystal Reports returns a number, the value of $\{orders\ detail.QUANTITY\}$ multiplied by 6.

All record selection formulas and group selection formulas must be Boolean.

Calculated data field

A calculated data field is a field that holds a value that comes from a calculation instead of coming directly from a database. For example, if the database you are using includes a $\{file.SALES\}$ field and a $\{file.COST\}$ field but no Gross Profit field, you can still show gross profit on your report, if you wish, using a calculated data field. To create a calculated data field, you simply create a formula that subtracts $\{file.COST\}$ from $\{file.SALES\}$. The formula calculates a Gross Profit value for each row and prints it wherever you place the formula.

Case sensitive

Case sensitive means that a program differentiates between UPPERCASE and lowercase letters when evaluating a text string. A case sensitive search for the word "house" will return only the value "house," but a non-case sensitive search will return "house," "House," "HOUSE," "HoUsE," and similar mixed-case responses. Crystal Reports operators (Equal, In string, etc.) are case sensitive.

Column	In Crystal Reports, a column is the display of data from a single field or formula. Columns run up and down the page. The words column and field are sometimes used interchangeably in this User's Guide. Contrast with Row definition.
Comments	Comments are blocks of text that accompany formulas to describe their functionality. Crystal Reports ignores comments when it runs the formula.
Concatenate	Concatenate means to join two or more text strings together to form a single contiguous string.
Condition	In an If-Then-Else formula, the condition is the If part of the formula, the set of circumstances that must take place (be true) to trigger the Then (or consequence) part of the formula. In the formula <code>If x<5 Then x Else 5</code> , the expression <code>x<5</code> is the condition.
Conditional Formatting	The ability to apply formatting to objects and section only in certain situations. For example, you can conditionally format numeric database fields to display in red when negative.
Conditional Formatting Formulas	Conditional Formatting Formulas are formulas that apply formatting to objects or sections only if certain conditions are met.
Conditional properties	Performed on an object only if a comparison statement returns True.
Consequence	In an If-Then-Else formula, the consequence is the Then part of the formula, the action that takes place if the If condition is met. In the formula <code>If x<5 Then x Else 5</code> , the expression <code>Then x</code> is the consequence.
Constant	<p>A constant is a value that is fixed and unchanging as opposed to a variable value which can take on different values depending on the circumstances.</p> <p>The value 5 is a constant; the value of the Quantity field (which sometimes may be 5, sometimes may be a different number) is a variable value. In the formula for converting pounds to ounces (<code>Ounces = Pounds * 16</code>) for example, 16 is a constant while Ounces</p>

and Pounds are variables. In the formula Today - January 1, 1900, January 1, 1900, is a constant, while Today is a variable that changes whenever the current date changes. In Crystal Reports, constants can be numbers, text strings, dates, dollar amounts, time, date/time, or the result of a formula that itself contains no variables, e.g., 14-9.

Container document

A file that contains an embedded or linked OLE object.

Cross-Tab

A Cross-Tab is a report that summarizes data and then presents the summaries in a compact row and column format that makes it easy to make comparisons and identify trends.

Data field

A data field (or field) is the basic building block of a record. Each record is made up of one or more data fields, and each data field can hold one piece of data (known as a value). A customer record in a typical customer mailing list database might contain data fields similar to these: Name, Address, City, State, Zip, Phone, Fax. A data field can be empty or contain a value. Data field data is generally displayed or printed in columns in the Details section of a Crystal Reports report.

Data Source

A Data Source is a database, table, query, dictionary or stored procedure result set that provides the data for a report.

Data types

A data type is a classification of the data that appears in a field or formula. Each piece of data used in a Crystal Reports report or formula has one of the following data types: text, dollar amount, number, date, or Boolean (TRUE/FALSE). It is important to understand data types because each function and operator works with only a limited number of data types (often as few as one). For some operators (+ and - for example), Crystal Reports uses a different set of calculation rules for one type of data than it uses for another.

Database

A database is a bank of related data. Each unit (record) of the database is typically organized in a fixed format to make it easier to retrieve selected portions of the data on demand. Each record is made up of one or more data fields, and each data field can hold one piece of data (known as a value).

Debug	Eliminating errors that occur when you run a formula.
Default	A default is a pre-loaded response to a software request for data. It is the response the computer accepts automatically if you do not enter different data.
Details area	A collection of one or more Details sections (i.e., Details A, Details B, etc.)
Details section	The Details section of a Crystal Reports report is the core section of the report. You structure the report in this section by inserting data fields, formulas, and other report elements.
Dictionary	A one-stop, ready-to-use source of data that is usually created for end users by computer professionals within the organization. The Dictionary takes away the need for the end user to search multiple databases, struggle with links, build formulas, and decode cryptic field names. The user just selects the data he or she needs from the Dictionary and builds the report.
Divide by zero protection	PCs will not allow you to divide a number by zero. If you attempt such a division, you will get a system error message. To protect you from a system error, Crystal Reports refuses to print a report which contains a formula that divides a value by zero.
Drag	<p>Drag means different things, depending on the context in which the word is used:</p> <ul style="list-style-type: none"> • When referring to moving a field, drag means to click on the field box and, while keeping the button pressed, to move it to a new position using the mouse. You release the mouse button when the field is in the position you want it. • When referring to resizing a field, drag means to click on one of the field box handles and, while keeping the button pressed, to make the field bigger or smaller using the mouse. You release the button when the field is the size you want it. • When referring to formatting text, drag means to highlight the text of interest by moving the I-beam cursor across it while the button is pressed. You release the button when you have finished highlighting.

Dynamic Link Library (DLL)

A Dynamic Link Library (DLL) is a special kind of file that contains Windows functions. DLLs are used by developers to extend the capabilities of Windows applications. The library is activated whenever a program or another DLL calls a function in the library. DLLs link on the fly, at runtime, whenever an included function is called. DLL functions are available on an as-needed basis to any program that can call DLLs; they do not need to be linked to the program via the compiler. The Crystal Reports Engine can be called as a DLL by developers for use with applications they are developing.

Element

The word element is used at times to describe individual report components such as database fields, formulas, group fields, and text. The Design Tab uses rectangular boxes to represent fields; text entered directly into the Design Tab appears as text in Crystal Reports.

Embed, Embedded object

An embedded object contains a presentation of the object, all of the data that pertains to the object, and information about the application used to create it. When you modify the original object in the server document, nothing happens to the embedded object unless you specifically update that object.

Empty date

An empty date [designated as Date(0,0,0)] is a date that contains no month, day, or year, and thus does not print. In Crystal Reports you use an empty date in If-Then-Else formulas that either return a date or not. For example, the formula:

```
If PageNumber = 1 Then
    PrintDate
Else
    Date(0,0,0)
```

prints the print date on the first page and prints nothing on every other page. Since the Then part of the formula is a date (PrintDate), the Else part of the formula must be a date as well, but a non-printing date. To create such a non-printing (empty) date you use the Date function and the arguments (0,0,0).

Empty number

An empty number (designated as zero [0]) is a field value that is printed typically when a value does not meet a specific condition in a numeric If-Then-Else formula. In Crystal Reports, you use an

empty number to specify that 0 be printed. For example, in the formula:

```
If {file.FIELD} = 3.5, Then
    {file.FIELD}
Else
    0
```

you are specifying that the numeric Gradepoint be printed (Then) if the grade point is 3.5 or higher. You are using the empty number 0 to indicate that 0 is to be printed (Else) if the grade point is below 3.5. Often a user will format the field that contains this formula to be suppressed if 0. In other words, nothing gets printed in the case of a zero value

Empty string

An empty string (designated as " ") is a string that contains no characters. In Crystal Reports, you use an empty string to specify that nothing be printed. For example, in the formula:

```
If {file.FIELD} = 3.5, Then
    "Cum Laude"
Else
    " "
```

you are specifying that the words Cum Laude be printed (Then) if the grade point is 3.5 or higher. You are using the empty string " " to indicate that nothing is to be printed (Else) if the grade point is below 3.5.

Evaluation time

Evaluation time refers to the time in the reporting process that a formula gets evaluated. Three evaluation times functions are:

- BeforeReadingRecords
- WhileReadingRecords
- WhilePrintingRecords

For a complete discussion of the Evaluation Times, see Chapter 11, *Advanced Formulas* 269.

Expert

Crystal Reports offers you several Experts. Experts are tools that take you step-by-step through various aspects of report creation. In most cases Experts have a series of numbered tabs. Simply begin at step one and proceed to the last step. When you have

completed the last step, the Crystal Reports Expert will do the rest of the work. Its that easy!

NOTE: The term “expert” in Crystal Reports is the term “wizard” in Microsoft Access.

Export

Export means to distribute your report to a disk file or through e-mail. Crystal Reports enables you to export your reports in many popular spreadsheet, database, word processor, HTML, and data interchange formats.

Field

A field is the basic building block of a record. Each record is made up of one or more fields, and each field can hold one piece of data (known as a value). A customer record in a typical customer mailing list database might contain fields similar to these: Name, Address, City, State, Zip, Phone, Fax. A field can be empty or contain a value. Field data is generally displayed or printed in columns in the Details section of a Crystal Reports report.

Field placement frame

A field placement frame is a rectangular cursor that appears as an aid to placing database fields and formulas on your report. Once you have selected a field or created a formula, the field box appears. When you move the box to the place in the report you want the field or formula to appear and click the button, Crystal Reports inserts the item at the point specified.

Field Value

See Value.

Field width

Field width is the size of the field in the originating database. A field width is generally fixed, and values in the field may take up all or only a part of the allotted width. The program includes many *Trim* functions for removing excess white space from field values that don't fill their respected fields.

File

A file is a collection of related data stored together on a disk or tape under a single name. In Crystal Reports, each report is stored as a single file.

Fixed properties

Properties that will always be performed on the object.

Flag A flag is a character or group of characters used to highlight or identify items of interest to call them to the reader's attention. For example, in an accounts receivable report, the words "past due" might be printed as a flag beside every past due account.

Flat formula A formula that does not reference any database field. For example:

1+1

Footer A footer is text that appears at the bottom of a report page. Footer text often includes page numbers and sometimes other information that describes or identifies the report. Crystal Reports gives you the option of printing the footer on all pages or only on selected pages of your report.

Format bar The bar that displays buttons you can click to perform many common formatting tasks. It is located below the toolbar.

Form letter In Crystal Reports, a form letter is a letter that can be reproduced, personalized, and customized using Crystal Reports powerful text object capabilities. Form letters generally include both text and field values. You create the letter and the program runs it each time inserting values from a different record in the database.

Formula A formula is a symbolic statement of the manipulations you want performed on certain data before it is printed on your report.

If your report is to contain a {file.SALES} field and a {file.COST} field, for example, you may want to create a GrossProfit field and designate its value as {file.SALES} - {file.COST} - {file.SALES} - {file.COST} is a simple formula that tells Crystal Reports to subtract the value of the {file.COST} field from the value of the {file.SALES} field and then to print the result.

You can use formulas to calculate numeric values, compare one value to another and select alternative actions based on the comparison, join multiple text strings into a single string, and for a multitude of other purposes. Creating a formula in Crystal Reports is much like creating one in your favorite spreadsheet.

NOTE: *The term "formula" in Crystal Reports is the term "expression" in Microsoft Access.*

Formula Editor	Formula Editor is the dialog box used to create and edit formulas. It contains tools for inserting fields, functions and operators into the formula, for checking formula syntax and for typing in formula components and arguments. Modified versions of the Formula Editor are used for creating Record and Group Selection formulas.
Formula language	Crystal Reports formula language is a powerful, yet easy to use, programming language designed for creating formulas.
Formula Syntax	Formula syntax is the set of grammar rules you are required to follow when creating formulas using the Crystal Reports formula language.
Freeform	Freeform implies that placement of objects is not limited to grids (vertical or horizontal).
Function	<p>A function is a built-in procedure or subroutine used to evaluate, make calculations on, or transform data. When you specify a function, Crystal Reports performs the set of operations built into the function without you having to specify each operation separately. In this way, a function is a kind of shorthand that makes it easier and less time consuming for you to create reports.</p> <p>Crystal Reports comes with a wide range of functions, and it also includes tools that allow you to build and save additional functions for yourself.</p>
Grand total	A grand total is the summary of all values in a column for the entire report.
Grid	In Crystal Reports, the grid is an underlying network of "lines" that are similar to the lines on graph paper. You can use these lines to help align fields and graphics. If you have Snap to Grid selected as one of your default options in the File Options dialog box, Crystal Reports will automatically align any fields you insert or resize to the nearest grid coordinate.
Group	A group is a set of records that are related to each other in some way. In a customer list, for example, a group could consist of all those customers living in the same ZIP code, or in the same

Region. In a sales report, a group could consist of all the orders placed by the same customer, or all of the orders generated by a specific sales representative. Crystal Reports offers you a great deal of flexibility in the way you group the data on your report.

- Group footer** A Group Footer is a section created by the program whenever you insert a group, a summary, or a subtotal. The Group Footer section is typically used to display the summary or subtotal.
- Group header** A Group Header is a section created by the program whenever you insert a group, a summary, or a subtotal. The Group Header section is typically used to display the name of the group or some other identifying information.
- Group value** (See Summary)
- Guidelines** Guidelines are lines that are non-printing lines that you can use for aligning, moving, and resizing objects with precision. Guidelines enable you to work in a freeform environment (without a grid), yet still have absolute control over the placing of objects in your report.
- Header** A header is text that appears at the top of a report page, above the body of the report. While a header can contain virtually any information, it often contains such things as the report title, company name, date, range of dates covered by the report, etc. Crystal Reports gives you the option of printing the header on all pages or only on selected pages of your report.
- HTML** The language used by the World Wide Web to publish web pages on the Internet that contain links to other pages.
- Index** An index is a small file that identifies the location of each record in a database. Since a tiny index file can be searched or sorted much quicker than a large database, Crystal Reports uses index files to speed up the report generation process. In a search, for example, Crystal Reports searches the index for the correct field location. Once found, Crystal Reports goes directly to the database field. Such a search does away with the need for searching every field of every record in a database. A database may have several indexes, each based on a specific field (or fields).

Indexed fields

Fields in the database that are in a specific order to speed up the retrieval of particular records. Instead of searching through all the data in all the records, the program goes first to the index, and finds a pointer that direct it to the specific record it is looking for. Indexed fields are tagged with arrowheads in the Select Expert and Visual Linking dialog boxes

In-place editing

The ability to change an OLE object's properties while in Crystal Reports. The menu items change to provide the editing tools from the server application so that you can make the changes easily.

In-place ruler

The ruler that appears when you are editing a text object. This ruler enables you to set tabs, and position objects with precision.

Insertion point

The insertion point is a vertical line that indicates the point at which Crystal Reports will insert any text that you type in. You set the insertion point by moving the I-beam cursor to the position you want to insert text and click. When typing text for the first time in a Design Tab section, Crystal Reports sets the insertion point flush left in the section, regardless of where you click the I-beam cursor.

Integer

An integer is a positive or negative whole number or zero. Integers have no decimal places. Crystal Reports Truncate function cuts the decimal places off a value, thus converting the value into an integer.

Link

A link is a field that is common to two or more databases and that serves as a connecting point between those databases. Crystal Reports uses the link to match up records from one database with those from the other(s). For example, if the databases each contain a customer number field (even though the fields might have different names), Crystal Reports can use those fields to electronically connect all records in one database with corresponding records in the other(s). When you create a single report based on multiple databases, the link assures that all the data in each row on that report refers to the same customer (transaction, invoice, etc.).

NOTE: The term "link" in Crystal Reports is the term "relationship" in Microsoft Access.

Live Header

A live header is a header that changes dynamically with the content of a field. If you group your data by region, for example, a typical live group header would print the name of the region at the beginning of each group.

Linked object

A linked object contains a presentation of the object, and a pointer to a defined part of the server document. When you modify the original object in the server document, the links assure that the object in your report is modified automatically as well. Conversely, if you modify the object in the container document, the original object file is modified as well.

Microsoft Foundation Class (MFC)

An object-oriented programming interface that encapsulates many related function calls into one object.

Nesting

In Crystal Reports, nesting means to use one If-Then-Else expression inside another. For example, If employees degree is not Ph.D. Then (if employee's sex is male, use the salutation Dear Mr. Else use the salutation Dear Ms.) Else use the salutation Dear Dr. In this example, the nested If-Then-Else statement is surrounded by parentheses. The example says, check the degree field on the employee record to verify that the employee is not a Ph.D. If that condition is true [the employee is not a Ph.D.], then use a letter salutation based on the sex indicated on the employee record. (If the sex is male, then use a male salutation. Else [if the sex is female] use a female salutation.) Else [that is, if the employee is a Ph.D.], use a Dr. salutation. By using this type of formula construction, you can create a wider set of conditions and a wider set of consequences easier than you could without nesting.

Null

Null means there is no value within a database field for a given record. It does not mean zero because zero is a value.

Null string

A null string is an empty string. It contains no characters. If you were to use the Count function to count the string, it would return a length of zero. "" is used to designate a null string.

Numeric

Numeric data is data on which you can perform arithmetic. The designation numeric refers to the way the data is treated by Crystal Reports and database programs, not to the way the data looks to you.

For example, a serial number 12345 looks numeric, that is, every character is a number. But a serial number is not the kind of data on which you would want to perform arithmetic so you would probably store a serial number as text instead of as numeric data.

Numeric is one of several data types. Database programs require you to designate a data type when you create a field for use in a database. The data type you select determines the rules the program follows when dealing with the values stored in that field.

Object

An object is one of several kinds of report elements that generally contain data and have specific properties that define their behavior or appearance. Crystal Reports uses the following kinds of objects:

- field objects
- text objects
- cross-tab objects
- graph objects
- subreport objects
- picture objects
- OLE objects

Each of these objects can be formatted individually, moved, resized, duplicated, and so forth.

ODBC

ODBC stands for Open Database Connectivity. It is an interface that gives applications the ability to retrieve data in data management systems using SQL for accessing the data. Such an interface allows a developer to develop, compile, and ship applications without targeting specific database management systems. Also called interoperability.

OLE

OLE is an acronym for Object Linking and Embedding. It refers to the ability to create compound reports, that is, reports that contain elements from other applications and that can be edited using the original application.

OLE container application

An OLE container application is an application that can contain and process OLE objects created elsewhere (like Paint or Paintbrush, etc.). Crystal Reports is a container application.

OLE server application	An OLE server application is an application that can create OLE objects that can then be placed in documents created by container applications. Crystal Reports is a container application; whereas, Microsoft Word and Excel are examples of server applications.
One-to-Many	One-To-Many refers to a situation occurring in linked databases in which one record in one database can be matched with many records in another database. An example of a One-To-Many link would occur when linking a customer table to an orders table. In such a case, for every one customer in the primary database, there would typically be many orders in the second (lookup) database.
Operators	<p>Operators are special symbols that describe an operation or an action to take place between two or more values.</p> <p>The symbol / for example, is an operator that means divide. A/B means Divide A by B. Crystal Reports reads the operators in a formula and performs the actions specified. Crystal Reports contains arithmetic, string, comparison, Boolean, conversion, date, and range operators.</p>
Order of Precedence	The order of precedence is a set of rules that determines the order in which arithmetic operations take place in a formula that involves multiple arithmetic operations. Multiplication (*) and division (/) are performed first (first tier operations), followed by addition (+) and subtraction (-) (second tier operations). When there are multiple operations involving the same tier, the order of precedence dictates that the operations are performed from left to right. You can use parentheses, if you wish, to alter the normal order.
Page footer	A section that prints at the bottom of each page. Page footers are typically used for page numbers, chapter names, and other identifying information.
Page header	A section that prints at the top of each page. Page headers are typically used for titles and other identifying information.
Parameter field	A special kind of field Crystal Reports that prompts the user for a value. You can use parameter fields for report titles, record selection, sorting, and a variety of other uses. Using parameter

fields enables you to create a single report that you can modify quickly to fit a variety of needs.

NOTE: The term “parameter field” in Crystal Reports is the term “parameter queries” in Microsoft Access.

Paste

In Crystal Reports, Paste means to retrieve and place data from the Windows clipboard into a report or formula. The data may have been Cut from the same report or formula or from a different one.

Perspective Editor (PG Editor)

The third party tool to change graphs that are plotted off the summaries in a report.

Population

A population is the entire set of values that might be tested statistically as opposed to a sample which is a subset of the population. A population does not necessarily refer to a group of people; it can refer to the number of automobiles produced on an assembly line or the number of construction companies bidding on a project.

For example, a Real Estate Agent might sell twenty houses in one year. The population of houses sold by that Agent in that year is twenty.

Population standard deviation

Population standard deviation is a statistical test of how the values in an entire population (all values) deviate from the mean or average value for that population. Population standard deviation is most often used when all values are being evaluated as opposed to just a sample of those values (StdDev).

NOTE: This comparison simply suggests typical usage. In practice, some users prefer a calculation based on N values (PopulationStdDev) while others prefer a calculation based on N-1 values (StdDev). Both forms of standard deviation are provided by Crystal Reports.

Population variance

Population variance is the square of the population standard deviation. It is a measure of the amount by which the values in an entire population vary from the mean (average) value for that population.

Population variance is typically used when all values are being evaluated as opposed to just a sample of those values (Variance).

NOTE: This comparison simply suggests typical usage. In practice, some users prefer a calculation based on N values (Population Variance) while others prefer a calculation based on N-1 values (Variance). Both forms of variance are provided by Crystal Reports.

Properties

Properties are qualities that define the appearance or action of an object or a section. There are two kinds of properties in Crystal Reports:

- On/Off Properties (i.e., *Suppress*)
 - A property that can only be toggled on or off.
- Attribute property (i.e., *color*)
 - A property for which you have to supply a value.

Range

A range is a set of values that fall between and include a defined upper and lower limit. For example, the range 10 to 20 includes 10, 20, and all the numbers that fall between. Also, the range January 1, 1991 to January 30, 1991, includes January 1, January 30, and all the dates that fall between. In Crystal Reports, a range can consist of numbers, dollar amounts, or dates.

Record

In a database, a record is a complete unit of related information, an electronic file folder that holds all of the data on a given entity. Each record contains one or more fields that contain the specific pieces of data of interest. In a customer database, for example, a record would store all of the data on a single customer. In an inventory database, a record would store all of the data on a single inventory item. Data from an individual record is displayed or printed as a row of data on a columnar report.

Report

A report is simply an organized presentation of data. As a management tool, a report is used to provide management with the insight it needs to run an organization effectively. Crystal Reports allows you to create comprehensive, customized, attractive management reports quickly and easily. But report in Crystal Reports also refers to invoices, form letters, mailing labels, and other related items that require the organization and output of data.

Report footer section

The Report footer section is the last section of your report in the Design Tab. You can place a summary in this section that you want to appear only on the last page of your report.

Report Gallery

Report Gallery is a special dialog box that appears when you click the NEW button or you choose NEW from the File menu. The Report Gallery serves as a gateway to all the Crystal Reports' report creation experts and to the graphical interface for selecting the report and data type for creating custom reports.

Report Header section

The Report Header section is the first section of your report in the Design Tab. You can place a title in this section, or any data you want to appear only on the first page of your report.

Request

Request is a set of criteria that specify the subset of data that you want to use for your report. For example, if you want your report to contain only California data, you create a record selection request that retrieves only California records for your report. You create record and group selection requests using the Select Expert.

Returns

The word returns refers to the result of a function, an operation, or a formula.

- When you use a function, it performs a calculation or manipulation that results in a data change of some kind. The data that results is what the function returns. For example, `Average(1,2,3,4,5)` returns the average of the array 1,2,3,4,5. `Truncate(1.2345)` returns the integer (whole number) portion of the number 1,2,3,4,5.
- When you use an operator, the result of the operation using that operation is what the operation returns. For example, $5*6$ equals 30. You can say that the operation $5*6$ returns 30. Also, the operation $100<200$ compares the two values and returns True; $200<100$ compares the two values and returns False.
- When you use a formula that contains functions or operators, each function or operation within the formula returns a result, but the formula taken as a whole returns a result too. When talking about a formula, it is the result of the formula that is of interest, not the result of individual functions or operations. For example, in the formula `If {file.QTY} < {file.REORDERAMOUNT} Then "Reorder "Else " "`, an internal

operation compares the value of the {file.QTY} field with the value of the {file.REORDERAMOUNT} field. If {file.QTY} is less than {file.REORDERAMOUNT}, that individual operation returns the value True. but that is not what the formula taken as a whole returns. The formula, taken as a whole, returns the flag "Reorder" when the operation internally returns the value True.

Row

In Crystal Reports, a row is the display of data from a single record. Rows run across the page. The words row and record are sometimes used interchangeably in this manual. Contrast with column.

Ruler

The Ruler is visible in both the Design Tab and Preview Tab when their respective check boxes are toggled on in the File Options dialog box.

The ruler provides a visual reference for positioning and resizing fields, graphs, lines, boxes, and bitmaps. The increments on the ruler are based on your measurement settings in the International section of the Windows control panel. The ruler also enables you to change page margins while immediately seeing the results of your changes on the report itself.

Running Totals

Running totals are totals that are displayed generally on a record by record basis. They total all records (in the report, in the group, and so forth) up to and including the current record. For example, if your first three records have values of 2, 4, and 6, the running total for each of the three records would be 2, 6 and 12, respectively.

Sample

A sample, as used in statistics, is a subset of a population used to represent the entire population. Researchers frequently do not have the option of testing an entire population before forming conclusions based on their tests. In such cases, they use a sample to represent the whole.

For example, political polling before elections is often based on questioning only four or five hundred people. From the answers given by this sample, predictions can be made on how an entire nation will vote.

Scroll bars, scrolling

Sometimes a window can display only a portion of a document. In such a case, the window includes scroll bars that you can use to move other parts of the document into the window for your review.

Scroll bars also appear with lists that are longer than the available window. The scroll bars allow you to move back and forth through the list. The process of moving through a list or document using scroll bars is called scrolling. In Crystal Reports, the screen automatically scrolls whenever you move the cursor outside the window and press and hold down the button.

Section

A section is a part of the report design environment. The program divides the design environment into several sections, each of which has different printing characteristics. You place objects in the various sections to build your report.

Select

- With regard to a report element (data field, formula, etc.), select means to point to the element and then click button to choose the element as the object of the next menu selection.
- With regard to text, select means to highlight the text by dragging the I-beam cursor over it.
- With regard to records, select means to identify and choose those records of interest while disregarding all others.
- With regard to groups, select means to identify and choose those groups of interest while disregarding all others.

Selection formula

A selection formula is a formula that specifies the records, or groups of records, you want included in your report.

Server document

A file that stores the original OLE object.

Shortcut Menu

A dynamic menu available in the Design and Preview Tabs. Access the shortcut menu by highlighting an object(s) and right-clicking.

Snap property

Snap is a "magnetic" property that attracts nearby objects. Crystal Reports uses two facilities that have the snap property: Guidelines and the Grid. Whenever an object is moved close to a guideline or a grid coordinate, the program snaps it into position for accurate placement and alignment.

Sort direction

Sort direction describes the way records or groups are printed in your report. They are printed either in ascending (A to Z, 0 to 9), or descending (Z to A, 9 to 0) order.

Sort and group by field

A sort and group by field is a field that triggers the printing of a subtotal (or a group field value) whenever its own value changes.

On a customer order report, for example, two fields are {customer.CUSTOMER ID} and {orders.ORDER AMOUNT}. If you want to subtotal by customer (total the orders for each customer), you click the {orders.Order AMOUNT} field as the field to subtotal and the {customer.CUSTOMER ID} field as the sort and group by field. Crystal Reports sorts the data by customer, so that all orders from the same customer are grouped together. Then, whenever the value in the {customer.CUSTOMER ID} field changes (when it changes from one customer to a different customer), Crystal Reports prints a subtotal of the values in the {orders.ORDER AMOUNT} field (a total of orders for the individual customer.) You also select sort and group by fields to trigger summaries.

Sort field

A sort field is a data field on which the sort procedure is based. A mailing list, for example, could be sorted, in ascending order, on the {customer.POSTAL CODE} code field; that is, the customers would be sorted so that those with the lowest postal codes would appear first and those with the highest Postal codes would appear last. It could also be sorted in ascending alphabetical order, on the {customer.CONTACT LAST NAME} field; that is, customers with last names beginning with A would appear first and those with last names beginning with Z would appear last. In these examples, {customer.POSTAL CODE} and {customer.CONTACT LAST NAME} are the sort fields.

Sort order

Sort order is an indicator of the direction in which you want your data to be presented, once sorted. Data is typically printed in one of two sort orders: ascending (lowest to highest, earliest to latest, first to last, a to z, etc.) or descending (highest to lowest, latest to earliest, last to first, z to a, etc.).

Sorting

Sorting is a method of organizing the order in which data appears on your report. Crystal Reports provides you with powerful tools for sorting your report data.

SQL SQL stands for Structured Query Language; a system for managing organizing and retrieving data stored on a computer database. Structured Query Language is a computer language that enables you to interact with a specific type of database called a relational database.

SQL Pass-through The ability to get the SQL Server to process the data retrieval criteria in order to pass the smallest possible result set back to Crystal Reports for final processing. When processing can be passed-through to the server, it makes the reporting process more efficient and it minimizes network traffic.

Standard deviation Standard deviation is the square root of the variance. It is a statistical test of how various values in a set of values deviate from the mean or average value for that set. You can use standard deviation, for example, for assessing the relative difficulty of tests given to students, for evaluating and projecting customer purchase patterns, or for comparing the results delivered by two or more products under evaluation (laboratory blood tests, smoke detectors, radar detectors, etc.) The uses are endless.

Standard deviation (as opposed to population standard deviation) is typically used to project the standard deviation for an entire population (all values) based on testing only a small sample of that population. For example, a company producing batteries with a new manufacturing process might want to test the batteries to determine how long they will last before they go dead. If the company tested all of its batteries, it would have no product left to sell. As an alternative, the company might test thirty batteries selected at random and project the mean burn out time and standard deviation for all batteries based on the results from that thirty battery sample.

NOTE: This comparison simply suggests typical usage. In practice, some users prefer a calculation based on N values (PopulationStdDev) while others prefer a calculation based on N-1 values (StdDev). Both forms of standard deviation are provided by Crystal Reports.

Static OLE object A static OLE object is a picture of an object that is stored in a document when it is saved. The picture can be displayed or printed by a user who does not have the application in which the

original object was created. The object cannot be edited in place, however, without first converting it to an editable type of object. Static OLE objects offer better online and print performance than do standard bitmaps.

String

A string is a series of connected characters (letters, numbers, symbols, spaces) stored and used as text. The word "hello" is a text string as is the phrase "Order # 2453" and the customer number "B30-124-777." Strings are sometimes referred to as text strings or character strings.

Subreports

A subreport is a report within a report. It has all of the characteristics of a report with one exception: it can not itself include a subreport. Subreports can be free-standing or they can be linked to the data in the primary report. Crystal Reports enables you to insert as many subreports as you wish.

Substring

A substring is simply a part of a larger string. "Columbia" is a substring of the string "British Columbia," "1040" is a substring of the customer number "B-1040-0032456," and "G" is a substring of the string "President George Bush."

Subtotal

A subtotal is a partial total, a total of a specific, limited group of data in a field. For example, given the following data:

1, 2, 3, 4, 5, 6, 7

a subtotal after the 3 produces the value 6 (1 + 2 + 3). A second subtotal after the 6 produces the value 15 (4 + 5 + 6).

Summary

A Summary is the value generated as the result of an evaluation, a tally, or a calculation performed on data from a single group.

A subtotal is the sum of all values from a single field, from all the records in a group. In a sales report, for example, if you subtotal the amount ordered by sales representative, Crystal Reports gathers all the records that belong to the sales representative and totals the amounts ordered from all the records.

In a group average, Crystal Reports averages the values in a group of records; in a group count, it counts the values in a group of records, and so forth. Summary values are important tools for creating powerful reports.

Summary field	A summary field is a field that determines the sum of the values, the average value, the maximum value, the minimum value, or count of values in a group of values in a given field. Much like a subtotal, a summary field groups data to your specifications and then performs the requested calculation/determination.
Syntax	Syntax, in Crystal Reports, is a set of rules that specifies the proper way to use functions and operators in formulas.
Tabs	Tabs are used in many dialog boxes and Experts in Crystal Reports. Tabs resemble the tabs on common file folders. Tabs always have text on them to indicate what you will find on the Tab.
Template	A template is a copy of a report used as the starting point for creating a new report. When you use a template, your original report remains unchanged.
Text box	A text box is a section of a dialog box that you can use for typing in text. For example, the large white box in the Formula Editor is a text box that you can use for typing in formula text.
Text object	A text object is a specialized Crystal Reports object that can contain text, database fields, and formula fields. It contains its own mini word processor that can be used for anything from adding a label to creating an entire document.
Text string	A text string is text that is entered directly onto the report itself instead of being entered via a data field or formula.
Toolbar	A bar at the top of the window that contains a number of buttons that you can click to activate the most frequently used Crystal Reports commands.
Total	A total is a sum of values. Subtotals and grand totals are different varieties of totals.
Truncate	Truncate means to cut off or eliminate all data that comes after the decimal point. Thus, if you truncate 1.2345, you get the value 1. If you truncate the value 1.9999 you also get the value 1. Truncate does not round data, it simply cuts off unwanted data.

**Two pass formula/
function**

A two pass formula is a formula that requires two passes through the data for completion. The first pass performs some calculation or selection and the second pass performs a calculation or selection that uses the result generated by the first pass.

An example of a two pass formula is one that calculates the sales for each sales representative as a percent of total company sales. The first pass sums the sales for each representative to arrive at total company sales. The second pass divides the sales per representative by total company sales to calculate the percent of total sales.

Underlay

The ability of an object (a bitmap, a graph and so forth) to print beneath multiple sections which follow the section in which it was placed. For example, you can place a bitmap in one section, format the section to underlay the following sections and then expand the bitmap so it appears as a background for the entire page of your report.

Value

A value is the data found in a field. For a field called {customer.CONTACT FIRST NAME}, for example, John or Mary might be the value. For a field called {orders.ORDER AMOUNT}, 1234.55 or \$200 might be the value.

Variance

Variance is the square of the standard deviation. It is a measure of the amount by which all values in a group vary from the mean (average) value in the group. It is a statistical test that can be used to evaluate the variability in a group of values (for example, the amount bid by each of the bidders on a construction project).

Variance (as opposed to PopulationVariance) is most often used to project the variance for an entire population (all values) based on testing only a small sample of that population. For example, with a limited number of bids in on a construction project, you might want to project the variance for all bids based on the sample already in. Or, based on sales figures for the first three months of the year, you might want to project the variance for orders for the entire year (including the nine months yet to come).

NOTE: These comparisons simply suggest typical usage. In practice, some users prefer a calculation based on N values (PopulationVariance) while others prefer a calculation based on N-1 values (Variance). Both forms of variance are provided by

Crystal Reports. For a more thorough discussion on the use of variance, consult any reliable statistics text.

Verify

In Crystal terms, verify does not mean to repair and compact the database (MDB file) in Access. It means to let the report understand the changes made to the database structure (fields and tables NOT records).

Wildcard

A wildcard is a character that represents any character (?) or any group of characters (*) in a search string. For example, if you are searching for Dan*, the search string will return strings like Danny and Daniel.

Word Wrap

Word wrap is a word processor-type property of a text object that automatically moves a word to the following line when the word is too long to fit the remaining space on the current line.

Symbols

\$ operator	629
% operator	626
() in formulas.....	262
() operator	626
-() operator	623
(Format menu).....	553
* operator.....	622
+ operator	610
+ operator (numbers).....	610
+ operator (string).....	613
- operator	629
/ operator	614
// operator	613
:= operator	611
; operator	628
< operator	619
<= operator.....	620
<> operator.....	624
= operator	615
> operator	616
>= operator.....	616
[,] operator.....	621
[] in formulas	262
[] operator.....	628
{ } in formulas	262

Numerics

16-bit installation requirements..	12
32-bit installation requirements..	12

A

A to B link	
setting up.....	469
A to B, A to C reports	
subtotaling.....	301
A to C link	
setting up.....	469
About Crystal Reports command	
(Help menu).....	567
About More Crystal Products	
command (Help menu)	567
Abs function	
examples using.....	570
explained	570
absolute formatting	185
absolute value function	
examples using.....	570
explained	570
Access database.....	484, 504
Access parameter queries	
opening.....	448
Access queries	
opening through DAO.....	444
opening through ODBC.....	446

Access via ODBC.....	504
Access via the DAO engine	485
Access via the Jet engine	484
accessing	
Access data.....	484, 504
ACT! data	492
Btrieve data.....	487
dBASE, FoxPro, Clipper	
data	480
dictionaries	516
direct access data.....	478
Essbase data.....	496
Excel data	505
Exchange data.....	490
INFORMIX data	507
Lotus Notes data	513
NT Event Log data.....	497
ODBC data sources.....	498
Oracle 7 data.....	493
Paradox data	482
query files	514
SQL data	508, 511
SQL Server data	494
Sybase data	495
Visual FoxPro data	513
ACT! database	492
using with Crystal Reports ..	471
activating	
databases	123
ActiveX Control	639
adding to project.....	640
changing properties	642
using	641
Add Database to Report command	
(Database menu).....	558
add operator	
examples using	610
explained	610
adding	
blank lines	181
guidelines	74
section	83
title page.....	108
adding a graphic to your	
dictionary	397
adding a new field heading	
(dictionary)	396
adding an ODBC database table to	
a report.....	457
adding blank lines	
conditionally.....	181
adding custom control to	
project.....	636
adding formulas to your	
dictionary	392
adding multiple tables	92

Adding Report Engine to	
application.....	635
adding the ActiveX Control to	
project	640
adding VCL to project.....	643
address	
CompuServe.....	8
Internet.....	8
ad-hoc reporting	26
advanced totaling	
introduction to.....	292
Aged0To30Days function	
examples using.....	570
explained	570
Aged31To60Days function	
examples using.....	570
explained	570
Aged61To90Days function	
examples using.....	570
explained	570
aggregate functions	
summarizing data with	370
AllDatesFromToday function	
examples using.....	571
explained	571
AllDatesFromTomorrow function	
examples using.....	571
explained	571
AllDatesToToday function	
examples using.....	571
explained	571
AllDatesToYesterday function	
examples using.....	572
explained	572
amount spelled out function	
examples using.....	603
explained	603
and operator	
example using	611
explained	611
API	
Report Engine	647
applications	
Report Engine	669
Arbor Essbase.....	496
architecture	
client/server.....	437
area	
printing characteristics.....	62
areas	
identifying	64
working with	64
areas of the Design Tab.....	61
Arrange Icons command	
(Window menu).....	565

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

array
 formula example.....283
 using in a formula.....282
arrays
 using with summary
 functions.....283
Arrow cursor.....58
Asc function
 examples using.....572
 explained.....572
ascending sort order.....204
assigning values to multiple
 variables.....280
assigning values to variables.....278
 conditionally.....281
assignment expression
 combining with a variable
 declaration.....280
assignment operator.....278
 examples using.....611
 explained.....611
assignment statements
 example.....279
attribute properties.....187
 conditional.....189
Auto Arrange Report command
 (Format menu).....553
Average functions
 examples using.....572
 explained.....572

B

basic report design.....148
before using the Report Engine.633
BeforeReadingRecords function
 examples using.....573
 explained.....573
beginner formulas.....250
blank lines
 adding.....181
 adding conditionally.....181
 deleting.....180
Border and Colors command
 (Format menu).....554
borders
 adding.....190
bottom N groups
 selecting.....333
bottom N sorting.....673
bound subreports vs. unbound
 subreports.....336
Box command (Insert menu).....551
boxes
 adding.....191
 editing.....191

brackets
 in formulas.....262
Browse Field Data command
 (Edit menu).....541
Btrieve database.....487
button conventions.....3

C

C
 Report Engine API.....647
 sample REAPI code.....668
Calendar1stHalf function
 examples using.....573
 explained.....573
Calendar1stQtr function
 examples using.....574
 explained.....574
Calendar2ndHalf function
 examples using.....573
 explained.....573
Calendar2ndQtr function
 examples using.....574
 explained.....574
Calendar3rdQtr function
 examples using.....574
 explained.....574
Calendar4thQtr function
 examples using.....574
 explained.....574
Cascade command
 (Window menu).....565
Change Group Expert command
 (Report menu).....562
changing properties for ActiveX
 Control.....642
changing properties in the
 VCL.....644, 645
changing properties on the
 properties list.....638
changing the ODBC data source
 accessed by a report.....461
characters
 extracting a range of.....285
chart objects.....69
Chr function
 examples using.....574
 explained.....574
Class Library
 Report Engine.....645
classes
 REC Library.....645, 646
clause
 DISTINCT.....369
 FROM.....365
 GROUP BY.....370, 374

HAVING.....374
ORDER BY.....372
SELECT.....368
WHERE.....367, 373
client/server architecture.....437
Clipper database.....480
Close All command
 (View menu).....546
Close All command
 (Window menu).....565
Close command (File menu).....534
Close command (View menu).....546
coding a custom print link.....654
color
 adding.....190
column coordinates.....68
combinations
 key.....3
combining sorting and
 grouping.....204
combining three or more unrelated
 reports.....346
combining two unrelated
 reports.....345
command (Format menu).....553
command conventions.....3
commands
 About Crystal Reports (Help
 menu).....567
 About More Crystal Products
 (Help menu).....567
 Add Database To Report
 (Database menu).....558
 Arrange Icons
 (Window menu).....565
 Auto Arrange Report
 (Format menu).....553
 Border and Colors
 (Format menu).....554
 Box (Insert menu).....551
 Browse Field Data
 (Edit menu).....541
 Cascade (Window menu).....565
 Change Group Expert
 (Report menu).....562
 Close (File menu).....534
 Close (View menu).....546
 Close All (View menu).....546
 Close All (Window menu).....565
 Compile Report
 (Report menu).....563
 Contents (Help menu).....565
 Context Help (Help menu).....566
 Convert (Edit|Object menu).....543
 Copy (Edit menu).....540
 Cross-Tab (Format menu).....555

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

Cross-Tab (Insert menu)	550
Cut (Edit menu).....	539
Data Date Field (Insert Special Field menu).....	548
Data Time Field (Insert Special Field menu).....	548
Database Field (Insert menu).....	546
Database menu	557
Delete Section (Edit menu)	542
Edit (Edit Object menu)	543
Edit menu.....	539
Edit Selection Formula (Report menu).....	561
Exit (File menu).....	539
Export (File Print menu)	536
File menu	534
Font (Format menu)	554
Format Bar (View menu)....	544
Format menu.....	553
Formula (Edit menu)	540
Formula Field (Insert menu)	547
Grand Total (Insert menu)...	549
Graph/Chart (Format menu)	555
Graph/Chart Expert (Insert menu).....	552
Grid in Design (View menu)	545
Grid in Preview (View menu)	545
Group (Insert menu)	550
Group (Report Edit Selection Formula menu)	561
Group Name Field (Insert menu).....	550
Group Number Field (Insert Special Field menu).....	548
Guidelines in Design (View menu)	545
Guidelines in Preview (View menu)	545
Help menu	565
Insert menu	546
Last Modification Date (Insert Special Field menu).....	548
Last Modification Time (Insert Special Field menu).....	548
Line (Insert menu).....	551
Links (Edit menu).....	543
Log Off Server (Database menu).....	560

Log On Server (Database menu)	560
Mail (File Print menu)	537
Merge Section (Edit menu)	542
Move Backward (Format menu).....	556
Move Forward (Format menu).....	557
Move Section (Edit menu)	541
Move To Back (Format menu).....	556
Move To Front (Format menu).....	557
New (File menu)	534
Object (Edit menu)	542
Object (Insert menu)	552
Open (Edit Object menu)...	543
Open (File menu).....	534
Options (File menu)	538
Page Margins (File menu)...	537
Page Number Field (Insert Special Field menu).....	547
Paragraph (Format menu).....	555
Parameter Field (Insert menu)	547
Paste (Edit menu)	540
Paste Special (Edit menu)	540
Picture (Insert menu)	552
Preview Sample (File Print menu).....	536
Print (File menu)	536
Print Date Field (Insert Special Field menu)	548
Print Preview (File menu).....	536
Print Time Field (Insert Special Field menu)	548
Printer (File Print menu)	536
Printer Setup (File Print menu).....	537
Record (Report Edit Selection Formula menu)	561
Record Number Field (Insert Special Field menu)	548
Redo (Edit menu)	539
Refresh Report Data (Report menu)	562
Register/Change Address (Help menu)	566
Remove from Report (Database menu)	558
Report Comments (Insert Special Field menu)	549

Report Distribution Expert (Report menu).....	564
Report Expert (Report menu).....	560
Report menu.....	560
Report Options (File menu).....	538
Report Style Expert (Format menu)	553
Report Title (Insert Special Field menu).....	549
Rulers in Design (View menu)	544
Rulers in Preview (View menu)	544
Save (File menu)	535
Save As (File menu)	535
Save Data with Report (File menu).....	535
Search (Help menu).....	566
Section (Format menu).....	556
Select Records Expert (Report menu)	561
Set Alias (Database menu)	558
Set Label Layout (File menu).....	537
Set Location (Database menu).....	558
Set Print Date (Report menu)	563
Show SQL Query (Database menu).....	560
Show/Hide Sections (Edit menu)	541
Sort Records (Report menu)	562
Special Field (Insert menu).....	547
Status Bar (View menu).....	544
Stored Procedure Parameters (Database menu).....	560
Subreport (Insert menu)	551
Subreport Links (Edit menu)	541
Subsection (Insert menu)....	550
Subtotal (Insert menu).....	549
Summary (Insert menu).....	549
Summary Field (Edit menu)	541
Summary Info (File menu)...	538
Tabs (Format menu).....	555
Technical Support Request (Help menu).....	566
Text Object (Edit menu)....	540
Text Object (Insert menu)	546
Tile Horizontally (Window menu).....	564
Tile Vertically (Window menu).....	564
Toolbar (View menu).....	544

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

Top N/Sort Group Expert (Report menu)	562	query files	514	using.....	637
Total Page Count Field (Insert Special Field menu)	547	SQL data	508, 511	Crystal Dictionaries	516
Undo (Edit menu)	539	SQL Server data.....	494	why use?.....	384
Using Help (Help menu)....	566	Sybase data	495	Crystal Dictionaries overview...	384
Verify Database (Database menu)	559	Visual FoxPro data	513	Crystal Dictionary	
Verify on Every Print (Database menu).....	559	considerations		creating	385
View menu	544	OLE.....	530	selecting for a report.....	401
Visual Linking Expert (Database menu).....	557	considerations when		Crystal Query	
Window menu.....	564	commenting	260	selecting for a report.....	381
Zoom (View menu)	545	content (report)		Crystal Query data sets.....	514
comment considerations	260	deciding on	148	Crystal Query Designer	
comment operator		Contents command (Help menu)	565	overview.....	360
examples using	613	Context Help command (Help menu)	566	Crystal Report	
explained.....	613	control conventions.....	3	status bar	54
commenting your formula.....	259	controls		Crystal Report Engine	
comments in formulas.....	259	dialog box.....	3	adding to application.....	635
Compile Report command (Report menu)	563	conventions		coding a custom print link	654
CompuServe		button	3	custom print link.....	653
address	8	command.....	3	declarations	647
support	8	control	3	distributing applications.....	669
concatenate operator		formula	258	error handling.....	663
examples using	613	Convert command (Edit Object menu)	543	introduction to.....	632
explained.....	613	converting a Crystal Reports		PEPrintReport.....	649
concepts		3.x or 4.x dictionary file.....	400	print only link	649
formatting	184	coordinates.....	68	sample custom print link....	658
reporting.....	157	Copy command (Edit menu)	540	structures	668
conditional attribute properties.....	189	Count functions		using.....	634, 648
conditional formatting 26, 178, 188		examples using	574	Crystal Reports	
conditional messages	177	explained	574	cursors.....	58
conditional on or off properties 188		counter		Design Tab	60
conditionally assigning values		record	67	documentation, using	5
to variables.....	281	creating		format bar	53
connecting to		custom print link to Report		formula language.....	321
Access data.....	484, 504	Engine	653	grid.....	68
ACT! data	492	interface for Report Engine 634		installing.....	12
Btrieve data	487	query from another query.. 379		installing on a network	
dBASE, FoxPro, Clipper		reports for Report Engine... 634		workstation	13
data	480	creating a new Dictionary.....	385	interface	24
dictionaries	516	creating a report	89	learning	34
direct access data	478	creating custom groups.....	207	license agreement.....	683
Essbase data.....	496	creating sample data for users		limited warranty	683
Excel data	505	to browse (dictionary).....	399	main window	48
Exchange data	490	Cross-Tab capabilities.....	29, 357	menu bar	48
INFORMIX data	507	Cross-Tab command (Format menu)	555	Preview Tab.....	65
Lotus Notes data	513	Cross-Tab command (Insert menu).....	550	Query data sets.....	361
NT Event Log data	497	Cross-Tab objects	70	Quick Start guide.....	15
ODBC data sources	498	Cross-Tab overview	352	registering.....	681
Oracle 7 data.....	493	Crystal ActiveX Control.....	639	replacement policy	683
Paradox data.....	482	Crystal Custom Control.....	636	Report Editor.....	124
		adding to project.....	636	return policy	682
		changing properties....	638, 639	SQL	440

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

warranty.....	683	formatting.....	161	database fields	
Crystal VCL.....	642	grouping.....	215	inserting.....	125
adding to project.....	643	grouping in intervals.....	235	placing on your report.....	161
changing properties... 644, 645		how to group.....	215	setting spacing between.....	138
using.....	644	placing on report.....	160	using in an object.....	96
CurrentDate function		selecting.....	89	database files	
examples using.....	575	selecting for your report.....	158	direct access.....	478
explained.....	575	summarizing data with		database links vs.	
CurrentDateTime function		aggregate functions.....	370	subreport links.....	339
examples using.....	575	Data Access Objects (DAO).....	485	Database Management Systems.....	439
explained.....	575	data analysis with queries.....	361	Database menu.....	50
CurrentTime function		Data Date Field command		Database menu commands.....	557
examples using.....	575	(Insert Special Field menu)...548		database support, list of.....	32
explained.....	575	data dictionaries.....	159	database table	
cursor		data fields		updating the location.....	394
Arrow.....	58	inserting.....	125	databases	
Double-arrow.....	58	data file.....	158	Access.....	484
Drag.....	58	opening (for your		ACT!.....	492
Drag and Drop, multi-unit... 58		dictionary).....	386	Btrieve.....	487
Drag and Drop, single-unit.. 58		data sets		dBASE, FoxPro, Clipper.....	480
Drill Down.....	59	Crystal Query.....	514	Essbase.....	496
Hourglass.....	59	data source, ODBC		INFORMIX.....	507
I-beam.....	58	changing.....	461	Lotus Notes.....	513
Insertion Point.....	59	checking settings for.....	455	Microsoft Access.....	504
Pencil.....	58	exporting to.....	115	Microsoft Excel.....	505
Rectangle.....	59	logging off.....	461	Microsoft Exchange.....	490
Section Sizing.....	59	logging on to.....	457	Microsoft SQL Server.....	494
Section Splitting.....	59	opening for your dictionary.....	388	NT Event Log.....	497
Stop.....	58	setting up.....	453	Oracle 7.....	493
Tiny Hand.....	59	data source, SQL		Paradox.....	482
cursors.....	58	opening for your dictionary.....	388	SQL.....	508, 511
custom control.....	636	data sources.....	477	Sybase System 10/11.....	495
adding to project.....	636	introduction to.....	476	Visual FoxPro.....	513
changing properties... 638, 639		ODBC.....	498	databases, SQL	
using.....	637	Data Tab of the		using.....	435
custom groups		Graph/Chart Expert.....	523	DataDirect Library.....	511
creating.....	207	Data Time Field command		DataDirect ODBC drivers.....	508
custom print link.....	653	(Insert Special Field menu)..548		Date function	
coding.....	654	data type		examples using.....	576
errors.....	663	in footer.....	153	explained.....	576
sample.....	658	in header.....	152	date report prints function	
customized views		in report body.....	154	examples using.....	592
of data.....	159	variable.....	276	explained.....	592
Cut command (Edit menu).....	539	data you can graph on.....	524	DateTime function	
D		data, grouped		examples using.....	576
DAO		subtotaling.....	221	explained.....	576
opening Access queries		summarizing.....	219	dateTime values	
through.....	444	data, summarized		printing conditionally.....	271
DAO engine		drilling down.....	81	Day function	
Microsoft Access.....	485	database		examples using.....	577
data		activating.....	123	explained.....	577
customized views of.....	159	relational.....	404	DayOfWeek function	
different views of.....	349	selecting.....	123	examples using.....	577
flagging.....	199	database data types.....	477	explained.....	577
		Database Field command		dBASE database.....	480
		(Insert menu).....	546		

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

dBASE for Windows			
Report Engine API.....	647		
sample REAPI code.....	668		
DBMS			
SQL.....	439		
DCU.....	642		
adding to project.....	643		
changing properties ...	644, 645		
using.....	644		
debugging a formula.....	287		
decision making with queries ...	361		
declaration, variable			
combining with an			
assignment expression	280		
combining with assignment			
expression.....	280		
declarations			
REAPI.....	647		
declarators			
variable.....	630		
declaring multiple variables.....	280		
declaring variables ...	276, 277, 630		
Delete Section command			
(Edit menu).....	542		
deleting			
blank lines.....	180		
formulas from your report ..	264		
guidelines.....	75		
individual working copies of a			
formula.....	265		
sections.....	83		
Delphi			
adding VCL to project.....	643		
changing VCL			
properties.....	644, 645		
Report Engine API.....	647		
sample REAPI code.....	668		
using the Crystal VCL.....	644		
Delphi VCL.....	642		
descending sort order.....	204		
Design Tab.....	60		
differences from Preview			
Tab.....	66		
similarities to Preview Tab ...	66		
Design Tab areas.....	61		
design, report.....	148		
freeform.....	68		
designing a report prototype.....	156		
detail sections			
multiple.....	25		
multiple, creating.....	83		
Details area.....	61		
Details section.....	61		
developer features.....	32		
developers			
what you should know	633		
developing			
prototype, on paper.....	156		
development			
Report Engine API.....	647		
dialog box controls.....	3		
dictionaries.....	516		
data.....	159		
overview.....	384		
why use?.....	384		
dictionary			
adding a graphic to.....	397		
adding formulas to.....	392		
adding help text to.....	396		
converting from Crystal			
Reports 3.x or 4.x.....	400		
creating.....	385		
creating a report from.....	401		
creating sample data for			
users.....	399		
editing.....	400		
selecting for a report.....	401		
different kinds of tutorials.....	2		
different views of the same data	349		
direct access database files.....	478		
direction			
of sort.....	203		
DISTINCT clause.....	369, 441		
DistinctCount functions			
examples using.....	578		
explained.....	578		
distributing Report Engine			
applications.....	669		
divide operator			
examples using.....	614		
explained.....	614		
documentation			
using.....	5		
Double-arrow cursor.....	58		
Drag and Drop cursor			
multi-unit.....	58		
single-unit.....	58		
Drag cursor.....	58		
dragging objects between			
reports.....	30		
drill down.....	81		
on summarized data.....	81		
Drill Down cursor.....	59		
drill down with graphs.....	524		
DTSToDate function			
examples using.....	579		
explained.....	579		
DTSToSeconds function			
examples using.....	579		
explained.....	579		
DTSToTime function			
examples using.....	579		
explained.....	579		
E			
Edit command (Edit Object			
menu).....	543		
Edit menu.....	49		
Edit menu commands.....	539		
Edit Selection Formula command			
(Report menu).....	561		
editing			
in-place.....	30		
editing an existing dictionary ...	400		
editing an SQL query.....	470		
editing your graphs.....	525		
Editor			
Report.....	124		
effects			
greenbar paper.....	179		
overlay.....	29		
element (report)			
selecting.....	128		
spacing.....	138		
element fields			
spacing between.....	138		
element markers			
selecting.....	128		
eliminating blank lines.....	180		
e-mail			
exporting to.....	112		
embedded objects.....	529		
entering			
data fields.....	125		
entering comments in formulas	259		
entering fields in formulas.....	255		
entering functions in formulas ..	256		
entering operators in formulas ..	257		
equal join.....	427		
equal operator			
examples using.....	615		
explained.....	615		
errors in custom print link.....	663		
Essbase.....	496		
evaluation of formulas.....	264		
evaluation time functions			
BeforeReadingRecords.....	573		
WhilePrintingRecords.....	606		
WhileReadingRecords.....	606		
example assignment statements	279		
examples			
Abs function.....	570		
absolute value functions ...	570		
add operator.....	610		
Aged0To30Days function ..	570		
Aged31To60Days function	570		
Aged61To90Days function	570		

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

AllDatesFromToday function.....	571
AllDatesFromTomorrow function.....	571
AllDatesToToday function.....	571
AllDatesToYesterday function.....	572
amount spelled out function.....	603
and operator.....	611
array in a formula.....	283
Asc function.....	572
assignment operator.....	611
Average functions.....	572
BeforeReadingRecords function.....	573
Calendar1stHalf function.....	573
Calendar1stQtr function.....	574
Calendar2ndHalf function.....	573
Calendar2ndQtr function.....	574
Calendar3rdQtr function.....	574
Calendar4thQtr function.....	574
Chr function.....	574
comment operator.....	613
concatenate operator.....	613
Count functions.....	574
CurrentDate function.....	575
CurrentDateTime function.....	575
CurrentTime function.....	575
Date function.....	576
date report prints function.....	592
DateTime function.....	576
Day function.....	577
DayOfWeek function.....	577
declaring variables.....	630
DistinctCount function.....	578
divide operator.....	614
DTSToDate function.....	579
DTSToSeconds function.....	579
DTSToTime function.....	579
equal operator.....	615
greater than operator.....	616
greater than or equal operator.....	616
GroupNumber function.....	580
Hour function.....	580
If-Then-Else operator.....	617
in array operator.....	618
in range operator.....	618
in string operator.....	619
InRepeatedGroupHeader function.....	580
inserting database fields.....	125
InStr function.....	580
integer function.....	604
IsNull function.....	581

Last4WeeksToSun function.....	581
Last7Days function.....	582
LastFullMonth function.....	582
LastFullWeek function.....	582
LastYearMTD function.....	582
LastYearYTD function.....	583
Left function.....	583
Length function.....	583
less than operator.....	619
less than or equal operator.....	620
Like operator.....	621
LooksLike function.....	584
LowerCase function.....	584
make array operator.....	621
Maximum function.....	584
Mid functions.....	585
Minimum function.....	586
Minute function.....	586
modulo function.....	593
Month function.....	587
MonthToDate function.....	587
multiply operator.....	622
negate operator.....	623
Next function.....	587
Next30Days function.....	588
Next31To60Days function.....	588
Next61To90Days function.....	588
Next91To365Days function.....	588
NextIsNull function.....	588
not equal operator.....	624
not operator.....	623
number (group) function.....	580
number (of characters) function.....	583
number (page) function.....	590
number (record) function.....	593
number (rounding) function.....	594
number in text field function.....	589
number, day as function.....	577
number, month as function.....	587
NumberToCode39 function.....	588
NumberToPostnet function.....	589
NumericText function.....	589
OnFirstRecord function.....	589
OnLastRecord function.....	590
or operator.....	625
Over90Days function.....	590
PageNumber function.....	590
parentheses operator.....	626
percentage operator.....	626
Picture function.....	590

PopulationStdDev functions.....	591
PopulationVariance functions.....	591
Previous function.....	592
PreviousIsNull function.....	592
PrintDate function.....	592
RecordNumber function.....	593
Remainder function.....	593
ReplicateString function.....	593
Right function.....	594
Round functions.....	594
Second function.....	595
Soundex function.....	595
Space function.....	595
spaces, leading (removing) function.....	604
spaces, trailing (removing) function.....	604
Startswith operator.....	627
statement separator.....	628
StdDev function.....	596
StrCmp function.....	596
StringToCode39 function.....	597
StringToPostnet function.....	597
subscript operator (string).....	628
subtract operator.....	629
Sum function.....	597
Summary function.....	598
text to number function.....	600
Time functions.....	600
to currency operator.....	629
to operator.....	622
ToNumber function.....	600
ToText functions.....	601
ToWords functions.....	603
Trim function.....	603
TrimLeft function.....	604
TrimRight function.....	604
Truncate function.....	604
UpperCase function.....	605
Val function.....	605
variables, declaring.....	630
Variance function.....	605
WeekToDateFromSun function.....	606
WhilePrintingRecords function.....	606
WhileReadingRecords function.....	606
Year function.....	607
YearToDate function.....	607
Excel database.....	505
Excel via ODBC.....	505
Exchange database.....	490
Exit command (File menu).....	539

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

expert
 select records, using 316

Export command
 (File|Print menu)..... 536

exporting
 reports 110
 to a fax 117
 to an ODBC data source.... 115
 to e-mail 112
 to HTML 31, 114
 to Lotus Notes..... 113

expression, assignment
 combining with variable
 declaration 280

expression, SQL
 creating 377

extended technical support
 policy 681

extending prices
 how to 223

extracting a range of characters 285

extracting a range of values 284

F

fax
 exporting to 117
 on demand 8

fax on demand 8
 support 678

fax support 8, 679

features
 developer 32
 online help 6
 Report Engine 633

commands
 Format 553

field
 group number, inserting..... 94
 inserting special 94
 page number, inserting 94
 print date, inserting 94
 record number, inserting..... 94
 sort 203
 sorting 210
 spacing 138
 total page count, inserting.... 94

Format 553

field boxes
 selecting 128

field handles..... 128

field heading
 adding new (dictionary) 396

field headings
 renaming in your
 dictionary 393

field objects 69

field objects, parameter
 overview 306

field values
 group 259
 sorting records
 according to 372

field, database
 using in an object..... 96

field, formula
 linking from 347, 348
 linking to 347, 348

field, headings
 moving within the list
 (dictionary) 395

field, multiple
 sorting 213

field, parameter
 creating 308
 responding to prompts 310
 using in a formula 309

fields
 adding to a query 365, 368
 entering in formulas 255
 how to add to a query 368
 inserting 125
 inserting in formulas 256
 moving within the list
 (dictionary) 395
 parameter 27
 renaming in your
 dictionary 393
 selecting for users
 (dictionary) 390
 sorting a single 204
 sorting multiple 204

fields box of the Formula Editor 255

fields in formulas
 syntax 261

fields, database
 placing on your report..... 161

fields, parameter
 creating customizable
 formulas using 312
 setting record selection
 with 309
 setting report title with 313
 setting sort order with 314
 using wildcards with 311

fields, special
 placing on your report..... 161

File menu 48

File menu commands 534

File Save As dialog box 134

file, data
 opening (for your
 dictionary) 386

first letter
 grouping on 231

flagging values 199

flags
 determining types 155
 identifying needs 155

Font command (Format menu) .554

footer
 data types 153
 group 61
 page 61
 report 61

footer information
 identifying 152
 sources of 153

footers
 inserting 107

form letter
 creating using a text object 170

form letters 28

format bar 53
 moving 72

Format Bar command
 (View menu) 544

Format menu 49

Format menu commands 553

formatting
 absolute 185
 concepts 184
 conditional 26, 178, 188
 properties 186

formatting data 161

formatting text
 with formulas 273

form-style reports 28

formula
 inserting fields in 256
 selection templates 322
 using a range in 284
 using an array in 282

Formula command (Edit menu) 540

formula conventions 258

Formula Editor
 fields box 255
 functions box 256
 introduction to 254
 operators box 257
 text box 257

formula field
 linking from 347, 348
 linking to 347, 348

Formula Field command
 (Insert menu) 547

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

formula language.....	321	using a prompting variable in.....	309	LastFullMonth.....	582
using for record selection ..	317	using semicolons in	286	LastFullWeek.....	582
formula rules.....	261	using variables in	273	LastYearMTD.....	582
formula syntax	261	using variables to streamline	273	LastYearYTD.....	583
fields	261	FoxPro database	480	Left	583
functions	262	freeform report design	68	Length	583
numbers	261	guidelines.....	69	LooksLike.....	584
operators	262	French braces in formulas.....	262	LowerCase.....	584
text	261	FROM clause.....	365, 441	Maximum.....	584
formula templates	322	functions		Mid	585
formula variables		Abs	570	Minimum	586
special requirements for	275	absolute value.....	570	Minute.....	586
formulas.....	30	Aged0To30Days	570	modulo.....	593
adding to dictionary	392	Aged31To60Days	570	Month	587
brackets in.....	262	Aged61To90Days	570	MonthToDate.....	587
changing evaluation time ..	573	AllDatesFromToday	571	Next	587
comments in.....	259	AllDatesFromTomorrow.....	571	Next30Days	588
copying from one report to another	267	AllDatesToToday	571	Next31To60Days	588
copying from online help ..	266	AllDatesToYesterday	572	Next61To90Days	588
creating	254	amount spelled out	603	Next91To365Days	588
creating customizeable formulas using parameter fields.....	312	Asc	572	NextIsNull	588
debugging	287	Average	572	number (group).....	580
deleting from your report...	264	BeforeReadingRecords	573	number (of characters).....	583
deleting individual working copies of.....	265	Calendar1stHalf	573	number (page).....	590
deleting the specification...	265	Calendar1stQtr	574	number (record).....	593
description	250	Calendar2ndHalf	573	number (rounding).....	594
entering fields in.....	255	Calendar2ndQtr	574	number in text field.....	589
entering functions in.....	256	Calendar3rdQtr.....	574	number, day as	577
entering operators in.....	257	Calendar4thQtr.....	574	number, month as	587
extending capabilities with variables	275	Chr	574	NumberToCode39.....	588
formatting text with	273	Count.....	574	NumberToPostnet.....	589
French braces in	262	CurrentDate.....	575	NumericText	589
If-Then-Else	270	CurrentDateTime	575	OnFirstRecord	589
inserting comments in	259	CurrentTime.....	575	OnLastRecord.....	590
inserting functions in	256	Date.....	576	Over90Days	590
inserting group field values in	259	date report prints.....	592	overview	570
inserting in formulas	258	DateTime.....	576	PageNumber	590
inserting in other formulas ..	258	Day	577	Picture.....	590
inserting numbers in	258	DayOfWeek.....	577	PopulationStdDev.....	591
inserting operators in	257	DistinctCount.....	578	PopulationVariance	591
inserting text in.....	258	DTSToDate.....	579	Previous	592
introduction to.....	250	DTSToSeconds.....	579	PreviousIsNull	592
order of evaluation	264	DTSToTime.....	579	PrintDate	592
order of precedence	264	entering in formulas	256	RecordNumber.....	593
parentheses in	262	GroupNumber	580	Remainder.....	593
removing from your report.	264	Hour	580	ReplicateString	593
renaming in your dictionary	393	InRepeatedGroupHeader ..	580	Right.....	594
square brackets in.....	262	inserting in formulas	256	Round	594
using a parameter field in ..	309	InStr	580	Second	595
		integer value	604	Soundex	595
		IsNull	581	Space	595
		Last4WeeksToSun	581	spaces, leading (removing) ..	604
		Last7Days	582	spaces, trailing (removing) ..	604
				StdDev	596
				StrCmp	596
				string (repeat # times)	593

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

StringToCode39.....	597
StringToPostnet.....	597
Sum.....	597
Summary.....	598
text to number.....	600
Time.....	600
ToNumber.....	600
ToText.....	601
ToWords.....	603
Trim.....	603
TrimLeft.....	604
TrimRight.....	604
Truncate.....	604
UpperCase.....	605
Val.....	605
Variance.....	605
WeekToDateFromSun.....	606
WhilePrintingRecords.....	606
WhileReadingRecords.....	606
Year.....	607
YearToDate.....	607
functions box of the	
Formula Editor.....	256
functions in formulas	
syntax.....	262
functions, summary	
using with arrays.....	283

G

getting help.....	8
grand total	
calculating a percentage of	238
discussed.....	155
Grand Total command (Insert	
menu).....	549
grand totals.....	162
graph	
determining data.....	524
drill down.....	524
placing.....	524
graph objects.....	69
Graph/Chart command	
(Format menu).....	555
Graph/Chart Expert.....	522
Data Tab.....	523
Options Tab.....	524
Text Tab.....	523
Types Tab.....	523
Graph/Chart Expert command	
(Insert menu).....	552
graphic	
adding to your dictionary...397	
inserting as OLE.....	530
graphic object	
inserting.....	97

graphing overview.....	522
graphs	
editing.....	525
underlay.....	525
greater than join.....	430
greater than operator	
examples using.....	616
explained.....	616
greater than or equal operator	
examples using.....	616
explained.....	616
greater than or equal to join.....	432
greenbar paper effect.....	179
grid.....	68
turning off.....	78
turning on.....	78
Grid in Design command	
(View menu).....	545
Grid in Preview command	
(View menu).....	545
GROUP BY clause.....	370, 374, 443
Group command (Insert menu)	550
Group command (Report Edit	
Selection Formula menu)....	561
group field values.....	259
inserting in formulas.....	259
Group Footer section.....	61
Group Header section.....	61
group headers.....	241
custom, creating.....	246
live (based on a formula),	
creating.....	245
live, creating.....	243
standard, creating.....	242
Group Name Field command	
(Insert menu).....	550
group number field	
inserting.....	94
Group Number Field command	
(Insert Special Field menu)..	548
group selection.....	320
setting up using the Expert.	329
group selection formula	
creating.....	331
group selection templates	
using.....	332
group sorting.....	673
group value	
positions.....	154
group values.....	154
identifying needs.....	154
sorting based on.....	223
summarizing.....	208
group values, summarized	
sorting.....	209

grouped data	
subtotaling.....	221
summarizing.....	219
grouping.....	204
based on first letter.....	231
in intervals.....	235
grouping and sorting.....	204
grouping data	
how to.....	215
grouping operations	
results.....	205
grouping overview.....	202
grouping records.....	162
grouping results.....	205
grouping your data.....	162
GroupNumber function	
examples using.....	580
explained.....	580
groups	
custom, creating.....	207
identifying needs.....	154
planning which to use.....	154
selecting for report.....	316
selecting to be included in	
a query.....	374
sorting the records in.....	217
top N.....	321
groups, bottom N	
selecting.....	333
groups, top N	
selecting.....	333
guidelines	
adding.....	73, 74
deleting.....	73, 75
freeform report design with..	69
moving.....	73, 75
using to move objects.....	75
using to position objects.....	75
Guidelines in Design command	
(View menu).....	545
Guidelines in Preview command	
(View menu).....	545

H

handles	
on element markers.....	128
HAVING clause.....	374, 443
header	
data types in.....	152
header information.....	152
sources.....	152
headers	
creating custom.....	246
creating live.....	243

Can't Find It?
Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

creating live group based on a formula.....245	how to add summary information to your report109	how to create sample data for users to browse (dictionary) 399
creating standard242	how to add tables to a query365	how to debug a formula..... 287
group..... 61, 241	how to assign a value to a variable278	how to declare a variable.....277
inserting107	how to calculate a percentage of the grand total238	how to declare and assign values to multiple variables..... 280
page 61	how to change margins194	how to delete a formula specification.....265
report 61	how to change stored procedure parameters.....465	how to delete a section 83
headers for custom groups	how to change the ODBC data source accessed by a report.461	how to delete formulas from your report264
creating246	how to check settings for an ODBC data source455	how to delete individual working copies of a formula 265
heading, field	how to combine a variable declaration and assignment expression280	how to do a multiple field sort . 213
adding new (dictionary).....396	how to conditionally assign values to variables.....281	how to do a single field sort 210
headings, field	how to convert a Crystal Reports 3.x or 4.x Dictionary file.....400	how to edit an existing dictionary 400
moving within the list (dictionary) 395	how to copy formulas from Crystal Reports online Help266	how to edit an SQL query 470
renaming in your dictionary393	how to copy formulas from one report to another.....267	how to edit boxes 191
help	how to create a footer that appears on all pages but the first.....198	how to edit lines 191
copying formulas from.....266	how to create a form letter using a text object.....170	how to export reports..... 110
features..... 6	how to create a formula.....254	how to extend prices and subtotal the extensions..... 223
getting 8	how to create a multi-detail section report83, 166	how to flag values that meet certain conditions 199
Help menu 50	how to create a new dictionary 385	how to format text with formulas.....273
Help menu commands565	how to create a new query364	how to get help 8
help text	how to create a parameter field 308	how to group based on first letter231
adding to your dictionary ..396	how to create a prompting variable308	how to group data..... 215
hiding objects 106	how to create a query from another Crystal query379	how to group data in intervals.. 235
highlights	how to create a record or group selection formula331	how to hide objects 106
planning text 155	how to create an SQL expression377	how to hide parts of the report . 103
Hour function	how to create customizeable formulas using parameter fields312	how to identify unique values in a query 369
examples using580	how to create customizable variables.....312	how to insert a freestanding subreport.....340
explained580	how to create group headers241	how to insert a graphic object.... 97
Hourglass cursor59	how to create If-Then-Else formulas270	how to insert a graphic/picture as an OLE object..... 530
how formula are evaluated.....264	how to create multi-condition If- Then-Else formulas272	how to insert a Page n of N field 95
how objects print 63	how to create multiple levels of subtotals227	how to insert a prompting variable..... 306
how to add a graphic to your dictionary.....397		how to insert an object 93
how to add a new field heading (dictionary).....396		how to insert comments in formulas..... 259
how to add a section..... 83		how to insert fields in your formula 256
how to add a title page to your report 108		how to insert footers 107
how to add an ODBC database table to a report..... 457		how to insert functions in your formula 256
how to add and link multiple tables 92		how to insert group field values in formulas.....259
how to add borders..... 190		how to insert headers..... 107
how to add boxes..... 191		
how to add color..... 190		
how to add fields to a query 365, 368		
how to add formulas to your dictionary.....392		
how to add help text to your dictionary.....396		
how to add lines 191		
how to add shading..... 190		

Can't Find It?
Your subject may be in online Help. Use the Find Tab
to do a full text search of Crystal Reports online Help.

how to insert numbers in formulas.....	258	how to rename formulas in your dictionary	393	how to specify a join type in a query.....	367
how to insert operators in your formula	257	how to resize a section	83	how to specify records to be included in a query	373
how to insert other formulas in formulas	258	how to resize objects	98	how to split sections.....	86
how to insert page footers	107	how to resize sections.....	86	how to subtotal grouped data...	221
how to insert page headers.....	107	how to respond to parameter field prompts.....	310	how to subtotal running totals within groups	295
how to insert text in formulas	258	how to respond to prompting variables	310	how to subtotal true A to B, A to C reports.....	301
how to link a subreport to its parent report	342	how to select a Crystal Dictionary for a report.....	401	how to subtotal without grouping.....	298
how to link multiple tables in your dictionary.....	389	how to select a Crystal Query for a report.....	381	how to summarize data with aggregate functions	370
how to link subreports.....	337	how to select a stored procedure from an SQL database.....	465	how to summarize grouped data	219
how to link tables.....	160	how to select data and begin creating a report.....	89	how to turn the grid off	78
how to link tables in a query	367	how to select fields for users (dictionary)	390	how to turn the grid on	78
how to log off an ODBC data source	461	how to select groups to be included in a query.....	374	how to undo an action	80
how to log on to an ODBC data source	457	how to select multiple objects ...	98	how to update the location of a database table	394
how to log on to MS SQL Server via ODBC	459	how to select objects	98	how to use a database field in an object.....	96
how to maintain running totals in a list	292	how to select tables for users (dictionary)	390	how to use a parameter field in a formula	309
how to maintain running totals within groups	295	how to select the top or bottom N groups.....	333	how to use a prompting variable	306
how to move field headings within the list (dictionary)	395	how to set a report title using parameter fields	313	how to use a prompting variable in a formula	309
how to move fields within the list (dictionary)	395	how to set a report title using prompting variables	313	how to use a range in a formula	284
how to move objects	98	how to set page orientation.....	197	how to use a SQL query that you designed elsewhere.....	362
how to move the format bar	72	how to set paper size	197	how to use an ACT! database... ..	471
how to move the toolbar	72	how to set record selection using parameter fields.....	309	how to use an array in a formula.....	282
how to open a data file for your dictionary.....	386	how to set record selection using prompting variables	309	how to use multiple sections in a report	166
how to open Access parameter queries	448	how to set section properties ...	105	how to use record selection templates.....	332
how to open Access queries through DAO	444	how to set sort order using parameter fields	314	how to use semicolons in formulas	286
how to open Access queries through ODBC.....	446	how to set sort order using prompting variables	314	how to use subreports with "unlinkable" data	347
how to open an ODBC data source for your dictionary	388	how to set up an A to B, A to C link	469	how to use variables in formulas	273
how to open an SQL data source for your dictionary	388	how to set up an ODBC data source.....	453	how to use wildcards with parameter fields.....	311
how to print date/time values conditionally	271	how to set up group selection using the Expert	329	how to use wildcards with prompting variables.....	311
how to print time values conditionally	271	how to set up record selection using the Expert	328	how to work with text objects ..	168
how to redo an action	80	how to sort based on summarized group values	223	how to zoom your report in	79
how to remove formulas from your report	264	how to sort records according to field values	372	how to zoom your report out	79
how to rename field heading in your dictionary.....	393	how to sort records within groups.....	217	HTML exporting to	31, 114
how to rename fields in your dictionary.....	393				

Can't Find It?
 Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

I			
I-beam cursor.....	58	Windows 95	13
identifying		Windows NT	13
areas.....	64	installing Crystal Reports on a	
sections	64	network workstation	13
identifying trends	29	InStr functions	
if you need more help	8	examples using.....	580
If-Then-Else formulas		explained.....	580
creating	270	integer function	
creating multi-condition	272	examples using	604
If-Then-Else operator		explained.....	604
examples using.....	617	interface	24
explained	617	creating for Report Engine	634
in array operator		Internet address	8
examples using.....	618	Internet support	8
explained	618	INTER SOLV DataDirect Library	511
in range operator		INTER SOLV DataDirect	
examples using.....	618	ODBC drivers.....	508
explained	618	intervals	
in string operator		grouping in	235
examples using.....	619	introduction to advanced	
explained	619	totaling	292
indexed tables.....	409	introduction to data sources	476
linking.....	421	introduction to formulas	250
information		introduction to the	
statistical	539	Formula Editor.....	254
information, summary		IsNull function	
adding.....	109	examples using	581
INFORMIX database	507	explained.....	581
INFORMIX via ODBC	507	J	
in-place editing	30	Jet engine	
InRepeatedGroupHeader function		Microsoft Access.....	484
examples using.....	580	join type	
explained	580	specifying in a query	367
Insert menu.....	49	join types	
Insert menu commands.....	546	equal	427
Insert Picture command		greater than.....	430
and OLE.....	529	greater than or equal to	432
inserting		left outer	428
a graphic object.....	97	less than	431
database fields.....	125	less than or equal to.....	433
OLE object	530	not equal to	434
special fields.....	94	right outer	429
text.....	132	SQL	426
text object	95	K	
inserting an object.....	93	key combinations	3
inserting comments in formulas	259	keyboard shortcuts	3
inserting formulas in formulas	258	kinds of tutorials.....	2
inserting group field values in		L	
formulas.....	259	language	
Insertion Point cursor	59	SQL	441
installation	12	Last Modification Date command	
installation requirements	12	(Insert Special Field menu).....	548
installing Crystal Reports	12		
Windows 3.x.....	13	Last Modification Time command	
		(Insert Special Field menu)	548
		Last4WeeksToSun function	
		examples using.....	581
		explained	581
		Last7Days function	
		examples using.....	582
		explained	582
		LastFullMonth function	
		examples using.....	582
		explained	582
		LastFullWeek function	
		examples using.....	582
		explained	582
		LastYearMTD function	
		examples using.....	582
		explained	582
		LastYearYTD function	
		examples using.....	583
		explained	583
		learning Crystal Reports	34
		learning paths	43
		Left function	
		examples using.....	583
		explained	583
		left outer join	428
		Length function	
		examples using.....	583
		explained	583
		less than join	431
		less than operator	
		examples using.....	619
		explained	619
		less than or equal operator	
		examples using.....	620
		explained	620
		less than or equal to join	433
		letter	
		grouping.....	231
		letter, form	
		creating using a text object	170
		letters	
		form	28
		levels, multiple	
		subtotals	227
		Library	
		Report Engine Class.....	645
		license agreement	683
		Like operator	
		examples using.....	621
		explained	621
		limited warranty.....	683
		Line command (Insert menu).....	551
		lines	
		adding.....	191
		editing.....	191

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

lines, blank	
adding	181
adding conditionally	181
deleting	180
linked objects	529
linking	
from a formula field	347
indexed tables	421
subreport to parent report	342
subreports	337
tables	160, 411
to a formula field	347
two or more tables	160
unindexed tables	349
linking from a formula field	348
linking multiple tables	92
linking multiple tables in your dictionary	389
linking to a formula field	348
links	
database vs. subreport	339
Links command (Edit menu)	543
live group headers	
creating	243
live group headers based on a formula	
creating	245
location	
database table, updating	394
Log Off Server command (Database menu)	560
Log On Server command (Database menu)	560
logging off	
ODBC data source	461
source	457
logging on to MS SQL Server via ODBC	459
looking up tables	
methods	423
LooksLike function	
examples using	584
explained	584
Lotus Notes	
exporting to	113
Lotus Notes database	513
Lotus Notes via ODBC	513
LowerCase function	
examples using	584
explained	584

M

Mail command (File Print menu)	537
--------------------------------	-----

mail-in support	8, 679
make array operator	
examples using	621
explained	621
make range operator	
examples using	622
explained	622
margins	
changing	194
markers	
element, selecting	128
setting spacing between	138
Maximum function	
examples using	584
explained	584
menu bar	48
Database menu	50
Edit menu	49
File menu	48
Format menu	49
Help menu	50
Insert menu	49
Report menu	50
View menu	49
Window menu	50
menus	
shortcut	56
Merge Section command (Edit menu)	542
messages	
conditional	177
methods of looking up tables	423
Microsoft Access database	484, 504
Microsoft Excel database	505
Microsoft Exchange database	490
Microsoft SQL Server	494
Mid functions	
examples using	585
explained	585
Minimum functions	
examples using	586
explained	586
Minute function	
examples using	586
explained	586
modulo function	
examples using	593
explained	593
Month function	
examples using	587
explained	587
MonthToDate function	
examples using	587
explained	587
mouse	
right button capabilities	56

mouse actions	5
Move Backward command (Format menu)	556
Move Forward command (Format menu)	557
Move Section command (Edit menu)	541
Move To Back command (Format menu)	556
Move To Front command (Format menu)	557
moving	
format bar	72
guidelines	75
objects	75, 98
toolbar	72
moving field headings within the list (dictionary)	395
moving fields within the list (dictionary)	395
moving objects using guidelines	75
MS SQL Server	
logging on to via ODBC	459
multi detail section report	
creating	83
multi-condition If-Then-Else formulas	
creating	272
multi-pass reporting	672
multiple	
detail sections	25
multiple field sort	
how to	213
multiple field sorts	204
multiple sections in a report	166
multiple tables	
adding	92
linking	92
linking in your dictionary	389
multiply operator	
examples using	622
explained	622
N	
naming a variable	276
negate operator	
examples using	623
explained	623
network workstation	
installing Crystal Reports	13
networks	
client/server	437
New command (File menu)	534
Next function	
examples using	587

Can't Find It?
Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

explained	587
Next30Days function	
examples using	588
explained	588
Next31To60Days function	
examples using	588
explained	588
Next61To90Days function	
examples using	588
explained	588
Next91To365Days function	
examples using	588
explained	588
NextIsNull function	
examples using	588
explained	588
not equal operator	
examples using	624
explained	624
not equal to join	434
not operator	
examples using	623
explained	623
Notes database	513
NT Event Log	497
number (group) function	
examples using	580
explained	580
number (of characters) function	
examples using	583
explained	583
number (page) function	
examples using	590
explained	590
number (record) function	
examples using	593
explained	593
number (rounding) function	
examples using	594
explained	594
number in text field function	
examples using	589
explained	589
number, day as, function	
examples using	577
explained	577
number, month as, function	
examples using	587
explained	587
numbers	
inserting in formulas	258
numbers in formulas	
syntax	261
NumberToCode39 function	
examples using	588
explained	588

NumberToPostnet function	
examples using	589
explained	589
NumericText function	
examples using	589
explained	589
O	
Object command (Edit menu)	542
Object command (Insert menu)	552
object, text	
using to create a form letter	170
objects	69
chart	69
Cross-Tab	70
dragging between reports	30
embedded	529
field	69
graph	69
hiding	106
how they print	63
inserting	93
linked	529
moving	98
OLE	30, 70
picture	69
placement	25
resizing	98
selecting	98
selecting multiple	98
subreport	69
text	27, 69, 95
underlying	71
using a database field in	96
variable length	167
objects, OLE	
static	528
objects, text	
placing on your report	161
working with	168
OCX	639
adding to project	640
changing properties	642
using	641
ODBC	159
INFORMIX	507
INTERSOVLV DataDirect	508
INTERSOVLV DataDirect	
Library	511
Lotus Notes	513
Microsoft Access	504
Microsoft Excel	505
opening Access queries	
through	446
SQL databases	508, 511

Visual FoxPro	513
ODBC data source	503
changing	461
checking settings for	455
logging off	461
logging on to	457
setting up	453
ODBC data sources	498
exporting to	115
opening for your dictionary	388
ODBC database table	
adding to a report	457
OLE and the Insert Picture	
command	529
OLE considerations	530
OLE control	639
adding to project	640
changing properties	642
using	641
OLE object	
inserting	530
OLE objects	30, 70
static	528
OLE Objects overview	528
on or off properties	186
conditional	188
OnFirstRecord function	
examples using	589
explained	589
OnLastRecord function	
examples using	590
explained	590
online help	
copying formulas from	266
online help features	6
Open command	
(Edit Object menu)	543
Open command (File menu)	534
Open Database Connectivity	
(ODBC)	498
opening	
Access data	484, 504
ACT! data	492
Btrieve data	487
dBASE, FoxPro, Clipper	
data	480
dictionaries	516
direct access data	478
Essbase data	496
Excel data	505
Exchange data	490
INFORMIX data	507
Lotus Notes data	513
NT Event Log data	497
ODBC data sources	498
Oracle 7 data	493

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

Paradox data.....482
 query files.....514
 SQL data.....508, 511
 SQL Server data.....494
 Sybase data.....495
 Visual FoxPro data.....513
 opening a data file for your
 dictionary.....386
 opening Access parameter
 queries.....448
 opening Access queries
 through DAO.....444
 opening an ODBC data
 source for your dictionary..388
 opening an SQL data source for
 your dictionary.....388
 operator
 assignment.....278
 operators
 add.....610
 and.....611
 assignment.....611
 comment.....613
 concatenate.....613
 divide.....614
 entering in formulas.....257
 equal.....615
 greater than.....616
 greater than or equal.....616
 If-Then-Else.....617
 in array.....618
 in range.....618
 in string.....619
 inserting in formulas.....257
 less than.....619
 less than or equal.....620
 Like.....621
 make array.....621
 make range.....622
 multiply.....622
 negate.....623
 not.....623
 not equal.....624
 or.....625
 overview.....610
 parentheses.....626
 percentage.....626
 Startswith.....627
 statement separator.....628
 string.....628
 subscript (string).....628
 subtract.....629
 to currency.....629
 operators box of the Formula
 Editor.....257

operators in formulas
 syntax.....262
 Options command (File menu) 538
 Options Tab of the Graph/Chart
 Expert.....524
 or operator
 examples using.....625
 explained.....625
 Oracle 7.....493
 ORDER BY clause.....372, 442
 order of evaluation in formulas 264
 order of precedence.....264
 order, sort
 ascending.....204
 descending.....204
 original.....205
 setting using parameter
 fields.....314
 setting using prompting
 variables.....314
 specified.....205
 original sort order.....205
 Over90Days function
 examples using.....590
 explained.....590
 overlay effect.....29
 overlay sections.....29
 overview
 Cross-Tab.....352
 Crystal Dictionaries.....384
 functions.....570
 graphing.....522
 grouping.....202
 OLE Objects.....528
 operators.....610
 sorting.....202
 summarizing.....202
 overview, parameter
 field objects.....306

P

page count
 total.....95
 page footer
 inserting.....107
 Page Footer area.....61
 Page Footer section.....61
 Page Header area.....61
 Page Header section.....61
 page headers
 inserting.....107
 page margins
 changing.....194
 Page Margins command
 (File menu).....537

page n of N field
 inserting.....95
 page number field
 inserting.....94
 Page Number Field command
 (Insert|Special Field menu) ..547
 page orientation
 setting.....197
 PageNumber function
 examples using.....590
 explained.....590
 paper
 greenbar effect.....179
 paper size
 setting.....197
 Paradox database.....482
 Paragraph command
 (Format menu).....555
 parameter field.....27
 creating.....308
 prompts, responding to.....310
 using in a formula.....309
 Parameter Field command
 (Insert menu).....547
 parameter field objects
 overview.....306
 parameter fields
 creating customizable
 formulas using.....312
 setting record selection
 with.....309
 setting report title with.....313
 setting sort order with.....314
 using wildcards with.....311
 parameter queries, Access
 opening.....448
 parentheses in formulas.....262
 parentheses operator
 examples using.....626
 explained.....626
 Paste command (Edit menu).....540
 Paste Special command (Edit
 menu).....540
 Pencil cursor.....58
 PEPrintReport.....649
 percent
 calculating for grand total..238
 percentage operator
 examples using.....626
 explained.....626
 performance tips
 selection.....325
 PGEitor.....525
 picture
 inserting as OLE.....530
 Picture command (Insert menu)552

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

Picture function	
examples using.....	590
explained	590
picture objects	69
placing data on your report.....	160
placing graphs.....	524
placing objects.....	25
PopulationStdDev functions	
examples using.....	591
explained	591
PopulationVariance functions	
examples using.....	591
explained	591
positioning objects	75
positioning objects using	
guidelines	75
precedence	
order of	264
Preview Sample command	
(File Print menu)	536
Preview Tab	65
differences from Design Tab	66
similarities to Design Tab	66
Previous function	
examples using.....	592
explained	592
PreviousIsNull function	
examples using.....	592
explained	592
prices	
how to extend	223
print	
how objects print.....	63
Print command (File menu).....	536
Print date field	
inserting	94
Print Date Field command	
(Insert Special Field menu) ..	548
Print Preview command (File	
menu)	536
Print Time Field command	
(Insert Special Field menu) ..	548
Print to Printer.....	146
PrintDate function	
examples using.....	592
explained	592
Printer command	
(File Print menu)	536
Printer Setup command	
(File Print menu)	537
printing	
hard copy report.....	146
printing characteristics	
area	62
print-only link	649

procedures	
stored.....	439
product registration	681
product replacement policy.....	683
product return policy.....	682
product support	678
programming	
Report Engine API	647
programming a custom	
print link.....	654
prompting variable	
creating.....	308
prompts, responding to	310
using in a formula	309
prompting variables	
creating customizable	
formulas using.....	312
inserting.....	306
see parameter fields	305
setting record selection	
with.....	309
setting report title with	313
setting sort order with	314
using.....	306
using wildcards with	311
prompts, parameter field	
responding to	310
prompts, variable	
responding to	310
properties	
attribute	187
changing.....	638
changing for ActiveX	
Control	642
changing VCL	644, 645
conditional attribute.....	189
conditional on or off	188
formatting	186
on or off.....	186
properties list	
changing properties on	638
properties, setting	105
prototype	
designing	156
developing on paper	156
purpose of report.....	150
Q	
queries	
data analysis with	361
decision making with.....	361
query	
adding fields to	368
adding tables to	365
creating a new query.....	364

creating from another	
query	379
how to add fields to.....	368
identifying unique values... 369	
linking tables in	367
selecting for a report.....	381
selecting groups to be	
included	374
specifying a join type.....	367
specifying records to be	
included	373
why use?	360
query capabilities.....	26
Query data sets	361
query data sets	514
Query Designer	361
overview	360
query, SQL	
editing.....	470
Quick Start guide	15
R	
range	
using in a formula.....	284
range limit requests.....	317
range of characters	
extracting	285
range of data	
selecting.....	316
range of values	
extracting	284
RDBMS applications	404
readers	
report, targeting	151
reading	
Access data	484, 504
ACT! data	492
Btrieve data	487
dBASE, FoxPro,	
Clipper data.....	480
dictionaries.....	516
direct access data	478
Essbase data	496
Excel data	505
Exchange data	490
INFORMIX data.....	507
Lotus Notes data.....	513
NTEvent Log data	497
ODBC data sources	498
Oracle 7 data	493
Paradox data	482
query files.....	514
SQL data	508, 511
SQL Server data.....	494
Sybase data	495

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

Visual FoxPro data.....	513	redo		using multiple sections in ..	166
REAPI.....	647	how to	80	zooming in and out	79
coding a custom print link	654	Redo command (Edit menu).....	539	report areas	
custom print link.....	653	Refresh Report Data command		Details.....	61
declarations	647	(Report menu).....	562	Page Footer.....	61
error handling.....	663	Register/Change Address		Page Header.....	61
PEPrintReport.....	649	command (Help menu).....	566	Report Footer.....	61
print only link	649	registering		Report Header.....	61
REC Library.....	646	Crystal Reports	681	Report Comments command	
sample code	668	registration of product.....	681	(Insert Special Field menu) ..	549
sample custom print link....	658	relational database.....	404	report design	
structures	668	Remainder function		basic.....	148
using.....	648	examples using	593	freeform	68
REC Library	645	explained	593	freeform with guidelines	69
Record command (Report Edit		Remove from Report command		Report Distribution Expert	
Selection Formula menu).....	561	(Database menu).....	558	command (Report menu).....	564
record counter.....	67	removing		Report Editor	124
record number field		formulas from your report..	264	report element	
inserting.....	94	leading spaces (function) ...	604	selecting	128
Record Number Field command		trailing spaces (function) ...	604	spacing	138
(Insert Special Field menu) ..	548	renaming field heading in your		Report Engine	
record selection.....	162, 316	dictionary	393	adding to application.....	635
setting up.....	316	renaming fields in your		API	647
setting up using the Expert ..	328	dictionary	393	before using in your	
setting using parameter		renaming formulas in your		application	633
fields.....	309	dictionary	393	creating interface for.....	634
setting using prompting		replacement policy.....	683	creating reports for.....	634
variables.....	309	ReplicateString function		distributing applications....	669
using the formula language	317	examples using	593	features.....	633
using the Select Records		explained	593	introduction to.....	632
Expert.....	316	report		using.....	634
with a group selection		body data.....	153	Report Engine API	
formula.....	327	body data, calculated	153	coding a custom print link ..	654
record selection formula		body data, existing	153	custom print link.....	653
creating	331	body data, sources	153	declarations	647
Record Selection Formula Editor		body, data types	154	error handling.....	663
interaction with the Select		content, deciding on	148	PEPrintReport.....	649
Records Expert	319	creating.....	89	print-only link.....	649
record selection formula		deleting formulas in	264	REC Library.....	646
templates.....	322	hiding sections	103	sample code	668
record selection templates.....	322	placing data on	160	sample custom print link....	658
using.....	332	placing database fields on ..	161	structures	668
RecordNumber function		placing special fields on	161	using.....	648
examples using	593	placing text objects on	161	Report Engine Class Library.....	645
explained.....	593	printing	146	Report Expert command	
records		purpose.....	150	(Report menu)	560
grouping.....	162	readers, targeting.....	151	Report Footer area.....	61
planning which to use	154	removing formulas from	264	Report Footer section	61
selecting for report.....	316	selecting a Crystal		Report Header area	61
sorting	162, 202	Dictionary for	401	Report Header section.....	61
sorting according to field		selecting a query for.....	381	Report menu	50
values	372	selecting data for.....	158	Report menu commands	560
sorting within groups	217	spacing fields	138	report objects	69
specifying to be included		statistical information	539	Report Options command	
in a query.....	373	title	151	(File menu).....	538
Rectangle cursor.....	59	titles	132		

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

report prototype		section, properties	
designing.....	156	setting.....	105
report sections.....	69	sections	
Details.....	61	identifying.....	64
Group Footer.....	61	multiple.....	25
Group Header.....	61	multiple, in report.....	166
Page Footer.....	61	resizing.....	86
Page Header.....	61	splitting.....	86
Report Footer.....	61	working with.....	64
Report Header.....	61	sections of your report.....	69
Report Style Expert command		SELECT clause.....	368, 441
(Format menu).....	553	Select Expert.....	320
report title		setting up group selection	
setting using parameter		with.....	329
fields.....	313	setting up record selection	
setting using prompting		with.....	328
variables.....	313	Select Records Expert	
Report Title command		interaction with Record	
(Insert Special Field menu) ..	549	Selection Formula Editor ..	319
report, parent		using.....	316
linking to subreport.....	342	Select Records Expert command	
report, unrelated		(Report menu).....	561
combining.....	345, 346	selecting	
reporting		database.....	123
ad-hoc.....	26	element markers.....	128
multi-pass.....	672	groups for report.....	154, 316
two pass, what is it?.....	672	multiple objects.....	98
reporting alternatives.....	25	objects.....	98
reporting concepts.....	157	range of data.....	316
reports		records for report.....	154, 316
copying formulas between.....	267	selecting a Crystal Dictionary	
creating for Report Engine ..	634	for a report.....	401
dragging objects between.....	30	selecting bottom N groups.....	333
exporting.....	110	selecting data.....	89
form-style.....	28	selecting data for your report ...	158
how to link subreports.....	337	selecting fields for users	
multiple detail sections.....	25	(dictionary).....	390
sample.....	31	selecting records.....	162, 316
subreports.....	25	setting up.....	316
reports, A to B, A to C		using the formula language.....	317
subtotaling.....	301	using the Select Expert.....	316
requirements		selecting tables for users	
installation.....	12	(dictionary).....	390
requirements for using variables		selecting top N groups.....	333
in formulas.....	275	selection	
resizing		group.....	320
objects.....	98	selection formula	
sections.....	83, 86	creating.....	331
resizing the toolbar.....	72	selection formula templates.....	322
restricted rights legend.....	688	selection performance tips.....	325
results using different sorting		selection templates.....	322
and grouping operations.....	205	using.....	332
return policy.....	682	selection, record	
Right function		with a group selection	
examples using.....	594	formula.....	327
explained.....	594		
right mouse button capabilities...56			
right outer join.....429			
Round functions			
examples using.....594			
explained.....594			
row coordinates.....68			
Rulers in Design command			
(View menu).....544			
Rulers in Preview command			
(View menu).....544			
rules			
formula.....261			
running totals.....291			
maintaining.....292			
maintaining within groups.....295			
subtotaling within groups...295			
runtime			
changing custom control			
properties.....639			
changing VCL properties...645			
runtime properties			
ActiveX Control.....642			
S			
sample code for Report Engine			
API.....668			
sample custom print link.....658			
sample data			
creating for users			
(dictionary).....399			
sample reports.....31			
Save As command.....134			
Save As command (File menu).....535			
Save As dialog box.....134			
Save command.....134			
Save command (File menu).....535			
Save Data with Report			
command (File menu).....535			
Save dialog box.....134			
saving report			
first time.....134			
Search command (Help menu).....566			
Second function			
examples using.....595			
explained.....595			
section			
adding.....83			
deleting.....83			
hiding in report.....103			
overlay.....29			
resizing.....83			
Section command			
(Format menu).....556			
Section Sizing cursor.....59			
Section Splitting cursor.....59			

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

semicolons			
using in formulas	286		
Set Alias command (Database menu)	558		
Set Label Layout command (File menu)	537		
Set Location command (Database menu)	558		
Set Print Date command (Report menu)	563		
set value (operator)	611		
setting properties at runtime	639		
setting up an A to B link	469		
setting up an A to C link	469		
setting up an ODBC data source	453		
setting up record selection	316		
setting up record selection using the Expert	328		
settings, checking for ODBC data source	455		
shading			
adding	190		
shortcut menus	56		
shortcuts			
keyboard	3		
Show SQL Query command (Database menu)	560		
Show/Hide Sections command (Edit menu)	541		
single field sort			
how to	210		
single field sorts	204		
sort			
how to	210, 213		
sort direction	203		
sort field	203		
sort order			
ascending	204		
descending	204		
original	205		
setting using parameter fields	314		
setting using prompting variables	314		
specified	205		
Sort Records command (Report menu)	562		
sorting	204		
based on summarized			
group values	223		
bottom N	673		
by record	202		
discussed	155		
group	673		
multiple field	204		
single field	204		
top N	673		
sorting and grouping	204		
sorting overview	202		
sorting records	162		
sorting results	205		
sorting summarized			
group values	209		
sorting your data	162		
Soundex function			
examples using	595		
explained	595		
Space function			
examples using	595		
explained	595		
spaces			
between fields	138		
spaces, leading (removing)			
function			
examples using	604		
explained	604		
spaces, trailing (removing)			
function			
examples using	604		
explained	604		
Special Field command (Insert menu)	547		
special fields			
inserting	94		
placing on your report	161		
special requirements for variables	275		
specification, formula			
deleting	265		
specified sort order	205		
splitting			
sections	86		
SQL			
Crystal Reports	440		
stored procedures	439		
SQL data source			
opening for your dictionary	388		
SQL database	508, 511		
selecting stored procedures from	465		
SQL databases			
using	435		
SQL databases via ODBC	508, 511		
SQL DBMS	439		
SQL expression			
creating	377		
SQL join types	426		
SQL language	441		
SQL query			
editing	470		
using a query you designed elsewhere	362		
SQL Server	494		
SQL statement			
DISTINCT clause	441		
FROM clause	441		
GROUP BY clause	443		
HAVING clause	443		
ORDER BY clause	442		
SELECT clause	441		
WHERE clause	442		
SQL stored procedures	439		
SQL/ODBC	159		
square brackets in formulas	262		
standard group headers			
creating	242		
starting a report beginning a report	89		
Startswith operator			
examples using	627		
explained	627		
statement separator			
examples using	628		
explained	628		
statements, assignment			
example	279		
static OLE objects	528		
statistical information	539		
status bar	54		
Status Bar command (View menu)	544		
StdDev function			
examples using	596		
explained	596		
Stop cursor	58		
Stored Procedure Parameters			
command (Database menu)	560		
stored procedures	439		
changing	465		
selecting from an SQL database	465		
StrCmp function			
examples using	596		
explained	596		
string (repeat # times) function			
examples using	593		
explained	593		
StringToCode39 function			
examples using	597		
explained	597		
StringToPostnet function			
examples using	597		
explained	597		
structures			
Report Engine API	668		

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

subreport
 linking to parent report 342
 Subreport command
 (Insert menu) 551
 subreport linking 337
 Subreport Links command
 (Edit menu) 541
 subreport links vs. database
 links 339
 subreport objects 69
 subreport, freestanding
 inserting 340
 subreports 25
 bound vs. unbound 336
 overview 336
 using with unlinkable data. 347
 subscript operator (string)
 examples using 628
 explained 628
 Subsection command
 (Insert menu) 550
 Subtotal command
 (Insert menu) 549
 subtotalling
 grouped data 221
 price extension 223
 without grouping 298
 subtotalling A to B,
 A to C reports 301
 subtotals 162
 multiple levels of 227
 without groups 298
 subtract operator
 examples using 629
 explained 629
 suggested learning paths 43
 Sum functions
 examples using 597
 explained 597
 summaries 162
 summarized data
 drilling down on 81
 summarized group values
 sorting 209
 summarizing
 grouped data 219
 summarizing group values 208
 summarizing overview 202
 summarizing your data 162
 Summary command
 (Insert menu) 549
 Summary Field command
 (Edit menu) 541
 Summary functions
 examples using 598
 explained 598

summary functions
 using with arrays 283
 Summary Info command
 (File menu) 538
 summary information
 adding 109
 support
 CompuServe 8
 databases, list of 32
 fax 8, 679
 fax on demand 678
 Internet 8
 mail-in 8, 679
 product 678
 telephone 9, 680
 support, technical 8
 Sybase System 10/11 495
 syntax
 formula 261
T
 table, database
 updating location 394
 tables
 adding to a query 365
 indexed 409
 linking 160, 411
 linking in a query 367
 linking multiple in your
 dictionary 389
 methods of looking up 423
 selecting for users
 (dictionary) 390
 tables, multiple
 adding 92
 linking 92
 tables, unindexed
 linking 349
 Tabs command (Format menu) 555
 technical support 8, 678
 technical support policy
 extended 681
 Technical Support Request
 command (Help menu) 566
 telephone support 9, 680
 templates
 formula 322
 record selection 322
 records selection formula 322
 templates, selection
 using 332
 text
 formatting with formulas 273
 highlights planning 155
 inserting 132

inserting in formulas 258
 text box of the Formula Editor .. 257
 text in formulas
 syntax 261
 text object 95
 inserting 95
 using to create a form letter 170
 Text Object command
 (Edit menu) 540
 Text Object command
 (Insert menu) 546
 text objects 27, 69
 placing on your report 161
 working with 168
 Text Tab of the Graph/Chart
 Expert 523
 text to number function
 examples using 600
 explained 600
 text, help
 adding to your dictionary .. 396
 Tile Horizontally command
 (Window menu) 564
 Tile Vertically command
 (Window menu) 564
 Time functions
 examples using 600
 explained 600
 time values
 printing conditionally 271
 Tiny Hand cursor 59
 title (report) 151
 title page
 adding 108
 title, report
 setting using parameter
 fields 313
 setting using prompting
 variables 313
 titles
 report 132
 to currency operator
 examples using 629
 explained 629
 to operator 622
 examples using 622
 explained 622
 ToNumber function
 examples using 600
 explained 600
 toolbar 51
 moving 72
 resizing 72
 Toolbar command
 (View menu) 544
 top N groups 321

Can't Find It?
 Your subject may be in online Help. Use the Find Tab
 to do a full text search of Crystal Reports online Help.

selecting	333
top N sorting	673
Top N/Sort Group Expert command (Report menu).....	562
total page count field.....	95
inserting.....	94
Total Page Count Field command (Insert Special Field menu) ..	547
total, grand calculating a percentage of	238
totaling, advanced introduction to	292
totals, running	291
maintaining	292
ToText functions examples using	601
explained.....	601
ToWords functions examples using	603
explained.....	603
trends identifying	29
Trim function examples using	603
explained.....	603
TrimLeft function examples using	604
explained.....	604
TrimRight function examples using	604
explained.....	604
Truncate function examples using	604
explained.....	604
turning off the grid.....	78
turning on the grid.....	78
tutorial, types	2
two pass reporting what is it?	672
two tables linking	160
types of data.....	477
types of formatting properties ...	186
types of tutorials.....	2
Types Tab of the Graph/Chart Expert.....	523
U	
unbound subreports vs. bound subreports	336
underlay using with graphs.....	525
underlying objects.....	71
undo how to	80

Undo command (Edit menu)....	539
unindexed tables linking.....	349
unlinkable data subreports with	347
unrelated reports combining.....	345, 346
upgrade plan	682
upgrading from a previous version of Crystal Reports.....	15
UpperCase function examples using	605
explained	605
user defined types.....	668
user interface creating for Report Engine .	634
using Query Designer.....	361
using arrays with summary functions.....	283
using Crystal Reports documentation.....	5
Using Help command (Help menu).....	566
using SQL	435
using SQL databases	435
using the ActiveX Control	641
Using the Crystal Report Engine	634
using the custom control.....	637
using the formula language for record selection	317
using the Report Engine API.....	648
using the select records expert .	316
using the VCL	644
using variables to expand formula capabilities	275
using variables to streamline formulas.....	273

V

Val function examples using	605
explained	605
value = or - disregarded.....	570
assigning to a variable.....	278
operator for assigning to variable	611
variable.....	278
value, group position.....	154
values extracting a range of.....	284
group	154

identifying unique in a query	369
values, group sorting based on	223
sorting summarized	209
summarizing.....	208
variable assigning a value to	278
declaring	277
variable data types	276
variable declaration combining with an assignment expression.....	280
combining with assignment expression.....	280
variable declarators	630
variable length objects	167
variable name	278
variable value	278
variable, prompting creating	308
responding to prompts	310
using in a formula.....	309
variables conditionally assigning values to	281
declaring	276
naming	276
operator to assign value to .	611
using in formulas	273
using to expand formula capabilities.....	275
using to streamline formulas.....	273
variables in formulas requirements	275
variables, declaring.....	630
examples	630
variables, multiple assigning values to.....	280
declaring	280
variables, prompting creating customizable formulas using.....	312
inserting.....	306
setting record selection with	309
setting report title with	313
setting sort order with	314
using wildcards with.....	311
Variance functions examples using.....	605
explained.....	605
VBX	636
adding to project	636
changing properties ...	638, 639

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

using	637	Windows 3.x	
VCL.....	642	installing Crystal Reports on .13	
adding to project	643	Windows 95	
changing properties ...	644, 645	installing Crystal Reports on .13	
using	644	Windows NT	
Verify Database command		installing Crystal Reports on .13	
(Database menu)	559	working with areas	64
Verify on Every Print command		working with sections.....	64
(Database menu)	559	workstation, network	
View menu	49	installing Crystal Reports on .13	
View menu commands	544		
views		Y	
of data	159	Year function	
views, different		examples using	607
of the same data	349	explained.....	607
Visual Basic		YearToDate function	
Report Engine API.....	647	examples using	607
sample REAPI code.....	668	explained.....	607
Visual Component Library.....	642		
adding to project	643	Z	
changing properties ...	644, 645	zoom	
using	644	how to	79
Visual Component Library		Zoom command (View menu) ..	545
(VCL).....	642		
Visual dBASE			
Report Engine API.....	647		
sample REAPI code.....	668		
Visual FoxPro database	513		
Visual FoxPro via ODBC.....	513		
Visual Linking Expert			
command (Database menu)	557		
W			
warranty.....	683		
Web activities			
simplifying.....	31		
WeekToDateFromSun function			
examples using.....	606		
explained	606		
WHERE clause	367, 373, 442		
WhilePrintingRecords function			
examples using.....	606		
explained	606		
WhileReadingRecords function			
examples using.....	606		
explained	606		
wildcards			
using with parameter fields	311		
using with prompting			
variables	311		
window			
Crystal Reports	48		
Window menu	50		
Window menu commands	564		

Can't Find It?

Your subject may be in online Help. Use the Find Tab to do a full text search of Crystal Reports online Help.

Can't Find It?
Your subject may be in online Help. Use the Find Tab
to do a full text search of Crystal Reports online Help.