Crystal Reports™ 8.5
User's Guide

Seagate Software IMG Holdings, Inc.
915 Disc Drive
Scotts Valley
California, USA 95066

Documentation Version 1. Febuary 2001.

Contents

Chapter 1: Welcome to Crystal Reports

Chapter 2: What's New

Chapter 3: Installing Crystal Reports

Chapter 4: Quick Start

Chapter 5: Report Design Concepts

Chapter 6: Introduction to Reporting

Chapter 8: Record Selection

Chapter 9: Sorting, Grouping, and Totaling

Chapter 10: Running Totals

Chapter 11: Multiple Section Reports

Chapter 12: Formatting

Chapter 13: Charting

Chapter 14: Mapping

Chapter 15: OLE

Chapter 16: Cross-Tab Objects

Chapter 17: Creating and Updating OLAP Reports

Chapter 18: Distributing and Viewing Reports

Chapter 19: Report Alerts

Chapter 20: XML

Chapter 21: Creating Reports from Excel and Access

Chapter 22: Using Formulas

Chapter 24: Creating Formulas with Crystal Syntax

Chapter 25: Parameter Fields

Chapter 26: Subreports

Chapter 27: Understanding Databases

Chapter 28: Accessing Data Sources

Chapter 30: Dictionaries

Appendix A: Report Processing Model

Appendix B: International Office Directory

Welcome to Crystal Reports 1

Welcome! This chapter introduces you to Crystal Reports, the world standard for desktop and web reporting, and provides you with an overview of the contents of this User's Guide.

About Crystal Reports

Crystal Reports is designed to work with your database to help you analyze and interpret important information. Crystal Reports makes it easy to create simple reports, and, it also has the comprehensive tools you need to produce complex or specialized reports.

Create any report you can imagine

Crystal Reports is designed to produce the report you want from virtually any data source. Built-in report experts guide you step by step through building reports and completing common reporting tasks. Formulas, cross-tabs, subreports, and conditional formatting help make sense of data and uncover important relationships that might otherwise be hidden. Geographic maps and graphs communicate information visually when words and numbers are simply not enough.

Extend reporting to the Web

The flexibility of Crystal Reports doesn't end with creating reports — your reports can be published in a variety of formats including Microsoft® Word and Excel, E-mail and even over the Web. Advanced Web reporting lets other members of your workgroup view and update shared reports inside their web browser.

Incorporate reports into applications

Application and web developers can save time and meet their users needs by integrating the report processing power of Crystal Reports into their database applications. Support for most popular development languages makes it easy to add reporting to any application.

Whether it’s the web master in IT, the promotion manager in marketing, the database administrator in finance or the CEO, Crystal Reports is a powerful tool designed to help everyone analyze and interpret the information that's important to them.

About this guide

This guide includes procedures for typical reporting tasks such as placing fields, formatting reports, and sorting records. It also contains information on more specific areas of interest such as advanced formula creation and accessing different types of data. Use this guide as a reference for your basic reporting needs as well as an introduction to new concepts in report creation.

Chapter Contents

The following is a short description of each chapter in this guide.

Chapter 1: Welcome to Crystal Reports

Welcome! This chapter introduces you to Crystal Reports, the world standard for desktop and web reporting, and provides you with an overview of the contents of this User's Guide.

Chapter 2: What's New

Building on the proven stability of its version 8.5 release, Crystal Reports now brings you several new features and enhancements, especially in the area of web reporting.

This chapter introduces the new features and enhancements, which fall into three key categories: Web reporting, Report Designer features, and Developer features.

Chapter 3: Installing Crystal Reports

This chapter shows you how to install Crystal Reports locally from the product CD. It also describes how to install Crystal Reports to a network server, and how to install from a network server to a workstation machine. Additional topics include "Customizing your installation" and "Installing Crystal Enterprise."

Chapter 4: Quick Start

This chapter begins with an overview of the sample reports and data commonly used with Crystal Reports. It then presents an overview of the Report Experts—both new and advanced users can benefit from looking at the Standard Report Expert as it has many steps in common with the other Report Experts. Finally, the chapter offers two tutorials: one for new users and another for more advanced users.

Chapter 5: Report Design Concepts

This chapter describes the basic concepts of report design and helps you decide what information you want to include in your report.

Chapter 6: Introduction to Reporting

This chapter shows you the basics of creating a report in Crystal Reports. First you'll learn about the report design environment, then about how to select database tables, place objects on a report, and how to sort, group, and total your report data.

Chapter 7: Designing Optimized Web Reports

This chapter suggests ways to enhance reports so they take advantage of performance enhancements made to Crystal Reports. While the suggestions made here are especially important for optimizing the performance of reports distributed over the thin-wire environment of the Web, the majority of the guidelines and procedures are applicable to all of your reports.

Chapter 8: Record Selection

This chapter shows you how to filter the records you want included in a report. For example, using the record selection tools, you can limit the records in your report to include only records for a specific group of customers, a specific range of account numbers, or a particular date range.

Chapter 9: Sorting, Grouping, and Totaling

Sorting, grouping, and totaling are the steps that turn disorganized data into useful information on a report. This chapter describes the types of sorting, grouping, and totaling you can do within a report.

Chapter 10: Running Totals

Running totals are a flexible and powerful way to create specialized summaries and continually incrementing totals.

This chapter shows you how to add a basic running total and a running total within a group to your report. You will also learn how to create conditional running totals and running totals using formulas.

Chapter 11: Multiple Section Reports

This chapter introduces the various types of sophisticated reports you can create using the multiple section reporting capabilities in Crystal Reports. These capabilities enable you to create reports that treat individual values differently based on sets of criteria you establish. These concepts are then applied to the creation of Form Letters.

Chapter 12: Formatting

Formatting refers to changes in the layout and design of a report, as well as the appearance of text, objects, or entire report sections. This chapter details methods you can use to draw attention to data, change the presentation of dates, numbers, and other values, hide unwanted sections, and perform a variety of other formatting tasks to give a report a professional appearance.

Chapter 13: Charting

Crystal Reports enables you to present summarized data in colorful, easy-to-read charts. This chapter demonstrates how to create charts and how to use them in reports to make report data more meaningful and easier to understand. You can choose from a number of chart layouts and types, as well as drill down to see the details behind the graphical summaries and format chart objects.

Chapter 14: Mapping

Crystal Reports enables you to include maps with reports made up of geographic data. This chapter explains how to use maps in reports to make report data more meaningful and easier to interpret. You can customize and rearrange the appearance of a map and activate the drill-down mode to view the details behind the graphical summaries.

Chapter 15: OLE

This chapter explains how Object Linking and Embedding (OLE) can be used to edit graphics or other objects from within your report instead of opening an additional application.

Chapter 16: Cross-Tab Objects

A Cross-Tab object is a grid that displays values that are grouped and summarized in two directions. This chapter provides you with information about how you can use Cross-Tab objects in your report.

Chapter 17: Creating and Updating OLAP Reports

This chapter describes how to create OLAP reports with a variety of OLAP and Multi-Dimensional data sources and how to update them when the location of your data changes.

Chapter 18: Distributing and Viewing Reports

This chapter provides you with information about how to distribute finished reports using a variety of methods (printing, faxing, exporting). It also includes information about opening reports in Enterprise folders.

Chapter 19: Report Alerts

This chapter provides information about creating and using alerts in your Crystal reports.

Chapter 20: XML

This chapter describes the XML data access and export capabilities of Crystal Reports. It begins with a brief overview of XML technology and then provides procedures for setting up XML data access and using the XML exporting features.

Chapter 21: Creating Reports from Excel and Access

This chapter describes the Add-Ins for Microsoft Excel and Microsoft Access. It explains how you create reports from an Access table or query and an Excel spreadsheet with the Crystal Report Wizard.

Chapter 22: Using Formulas

This chapter explains the basics of formulas and introduces you to the Formula Editor in order for you to begin to create formulas.

Chapter 23: Creating Formulas with Basic Syntax

This chapter provides you with an overview of various parts of a formula. You learn about Basic syntax and the techniques you can use when creating a formula.

Chapter 24: Creating Formulas with Crystal Syntax

This chapter provides you with an overview of various parts of a formula. You learn about Crystal syntax and the techniques you can use when creating a formula.

Chapter 25: Parameter Fields

This chapter explains what parameter fields are and how they can be applied to create a single report that can be used to access different types of data depending on the user's needs.

Chapter 26: Subreports

A subreport is a report within a report. With subreports, unrelated reports can be combined into a single report. You can coordinate data that otherwise cannot be linked and present different views of the same data in a single report. This chapter shows you how to create and use subreports.

Chapter 27: Understanding Databases

This chapter describes the essentials you'll need to know to understand database design and use. After describing relational databases (including those from ODBC data sources), indexing, and table linking, the chapter introduces the concepts of server-side processing (pushing report processing to the server level) and field mapping (re-establishing report and database field mappings after a database changes). Finally, the chapter focuses on common database-related tasks such as changing a database driver and creating an ODBC data source.

Chapter 28: Accessing Data Sources

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

Chapter 29: The Crystal SQL Designer

A query is a selection statement used to obtain specific information from the database. This chapter describes how to use the Crystal SQL Designer to create, modify and optimize queries.

Chapter 30: Dictionaries

Dictionaries are structured and simplified views of data that can be created for some or all of the individuals in an organization. When working with Dictionaries, users see only the information they need and that you want them to see. They reduce support costs and time, and reduce data misuse, loss, and damage. This chapter shows how to set up and use Dictionaries to improve organizational efficiency and security.

Appendix A: Report Processing Model

This appendix provides you with in-depth information about the Report Processing model. This model determines the order in which data is accessed and manipulated during report generation.

Glossary

This guide comes with a comprehensive glossary explaining basic database and reporting concepts as well as terms specific to the program.

Online help

Crystal Reports online help includes all topics within the user's guide. It also provides additional reference material ranging from specific information on the function of a button to general information on how to create a report formula based on a sample business scenario.

Locate information quickly

Access online help from the Crystal Reports help menu.

Use the Contents tab to view all major sections in the online help and drill down to specific headings within each section, the Index tab to view topics in alphabetical order, and the Search tab to enter a keyword to view all the sections that relate to the keyword.

Sample Reports

Many topics in the Crystal Reports online help include a list of related sample reports. Use these reports to illustrate concepts the topic describes. The sample reports can be adapted to your own needs.

Sample reports are located in the Crystal Reports directory under `\Samples\En\Reports`.

For a list of sample reports see "Sample reports" on page 34.

Product registration

There are several ways you can register your product:

- Fill out the Product Registration form on the Seagate Software web site at: `http://www.seagatesoftware.com/register`
- Use the Registration Wizard accessible from the Crystal Enterprise program group.
- Print the Product Registration form and then fax it to the Registration fax number closest to you. Seagate Software will then fax you a registration number that can be entered into the product the next time you use it.

  Registration Fax Numbers

  USA/Canada +1 (604) 681-5147
  United Kingdom +44 (0) 20 8231 0601
  Australia +6 2 9955 7682
  Germany +49 (0) 69 9509 6182
  Hong Kong +852 2893 2727
  Singapore +65 777 8786

Registration is required to access online or telephone technical support. In addition, registering the product will ensure that you are kept up-to-date with product advancements.

Technical Support

To find out how to receive technical support services on the product you purchased:

- Consult the enclosed *Technical Support Guide*.
- Go to our support website at `http://support.seagatesoftware.com`.
- Contact your regional office. See the "International Office Directory" on page 661.

Command, button, and key conventions

This manual uses the following conventions:

Commands and buttons

For easy recognition, command names and button names are displayed in bold. For example: **New** command, **Print** 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).
- Enter means the Enter, Return, 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 (Ctrl-C is the Windows Copy command).

Other conventions

Letter Gothic font indicates data that you enter using the computer keyboard. It is also used to show example formulas: `{orders detail.UNIT PRICE}*.85`.

What's New 2

Building on the proven stability of its version 8 release, Crystal Reports now brings you several new features and enhancements, especially in the area of web reporting.

This chapter introduces the new features and enhancements, which fall into three key categories:

- Web reporting
- Report Designer features
- Developer features.

Web reporting

To better meet your web reporting needs, Crystal Reports now integrates fully with Crystal Enterprise, a web-based report management tool that works within your company's existing web infrastructure.

The Crystal Enterprise Standard CD (shipped with Crystal Reports 8.5) includes a temporary product license keycode that lets users install and operate the Crystal Enterprise system for a period of 30 days. To obtain a free, non-expiring product activation license keycode, users must register their copy of Crystal Enterprise Standard. Please use the Crystal Registration Wizard located in the Crystal Enterprise Program Group to complete your registration.

If you purchased the Professional or Developer edition of Crystal Reports, then you also received the Crystal Enterprise Standard CD. This additional CD is included in the box with Crystal Reports, along with five free concurrent access licenses of Crystal Enterprise Standard.

Increased value of the Crystal Enterprise Web Component Server

Use the newly re-designed and enhanced Web Component Server (WCS) from Crystal Enterprise to power your web-reporting solution and to replace the older WCS from Crystal Reports 8. Crystal Enterprise Standard provides significant improvements and enhancements over the version 8 WCS.

Run the Web Component Server on any machine

Install the Crystal Enterprise Standard CD on whichever machine you want to coordinate your web reports. With Crystal Enterprise Standard, you no longer need to install and run Crystal Reports and the WCS on your web server.

If you need a reporting solution with greater scalability, use Crystal Enterprise Professional, which can be installed on as many machines as are necessary (whereas Crystal Enterprise Standard is limited to installation on a single machine). Further, with Crystal Enterprise Professional, you can specify the server pieces you want running on each machine.

Scale your reporting framework to meet increasing demands

Crystal Enterprise provides you with a scalable, web-based solution for managing the access and delivery of hundreds or thousands of mission-critical Crystal reports to every decision-maker—across the enterprise and beyond.

As your company's web reporting needs increase, you can upgrade seamlessly to Crystal Enterprise Professional without having to reinstall; you can also add additional licenses and components so that Crystal Enterprise grows right along with you.

Schedule reports to maintain up-to-date information

Schedule important reports to run on a regular basis so that everyone has access to the most current information about your enterprise.

Publish reports to the Web in seconds

With just a few easy steps, you can publish your reports to the Web for viewing by all with the Report Publishing Wizard.

Manage folders for sharing reports

Share reports across the enterprise or across the Web by publishing them to the default Guest favorites folder on the Automated Process Scheduler (APS). Users can also publish to other folders, as determined by the Administrator.

Administer easily from one central console

By consolidating administrative tasks, the Crystal Management Console (CMC) makes administering your web reporting solution quick and easy—regardless of the size of your enterprise.

Use the CMC to gain immediate control over what can or cannot be seen, run, or managed by everybody. You define all users' viewing and managing privileges for particular reports.

Integrate Crystal Reports performance with existing IT investments

Trust your Crystal web reporting solution to work to its full potential with your existing web server investment.

Crystal Enterprise Standard has improved web server support, including compatibility with Domino web servers through DSAPI interface, and Apache servers through a DSO (Dynamic Shared Object) module on Solaris and Linux. Crystal Enterprise Standard also supports CGI on Solaris and Linux.

Use scripting on the Crystal Enterprise WCS

The Crystal Enterprise Web Component Server now provides complete support for scripting.

Complete XML support for the Web

Crystal Reports now fully supports the Extensible Markup Language (XML) adopted by the World Wide Web Consortium (W3C) for delivering content over the Internet.

XML is emerging as the standard data format for the whole data-related industry because it's recognized across applications and across platforms. Keeping pace with the current advance in technology, Crystal Reports allows you to report off existing XML data and to export your work to XML format.

Access XML data directly

Access your existing XML data sources directly, as if they were normal database tables. If necessary, combine your XML data with standard database information in your report. If you don't yet have an existing XML data source, open the ODBC folder of the Data Explorer and locate the sample XML data source installed with Crystal Reports.

Export report data straight to XML

Export your report data to XML format quickly from within Crystal Reports. This allows you to send the data to other eCommerce applications used to read and manipulate information.

Accomplish more with faster processing

Maximize the work you accomplish with Crystal Reports by taking advantage of the enhancements made to the Crystal Report Engine, whose improved memory management brings faster performance—especially on multi-processor machines.

In addition, the Report Engine now speeds up processing of on-demand subreports and reports with parameter fields, supports increased multithreaded database access, and allows for an unlimited number of simultaneous jobs.

Improved interactivity with parameters

Using parameters not only enables user-driven reporting, but also increases report performance, especially over the Web. Create multi-purpose reports to ensure that everyone receives the right information quickly.

All of Crystal Reports' web-enabled report viewers have been improved to allow report designers and end users to utilize parameters to their full potential over the Web. Whether users prefer the Java, ActiveX, or DHTML viewers, they can interactively regulate the information they see by accepting default report parameters, picking from listed options, specifying multiple values, or entering ranges.

Customizable DHTML Report Viewer

Customize the Crystal Reports Dynamic HTML viewer to integrate seamlessly with the design of an existing eCommerce web site or corporate intranet. The DHTML viewer exposes the operations of its toolbar (via JavaScript), thereby giving report designers the ability to customize both its workings and its look-and-feel.

The enhanced DHTML viewer is so flexible that users can view reports, drill down on charts, link to on-demand subreports, or quickly export valuable data—all without even seeing that their experience is powered by Crystal Reports.

Optimize report performance with Crystal Reports documentation

Learn all the ways to improve the performance of your reports with the newest chapter in the Crystal Reports user documentation and online help. In this performance guide—"Designing Optimized Web Reports" on page 105—special attention is given to topics related to fast, efficient web reporting. The guide also specifies how to leverage your DBMS power by effectively using the SQL Expression Editor, Grouping on Server, and other server-side processing. You can go as "in-depth" as you like, to get the performance you need.

Log on to APS folders from Crystal Reports

Log on quickly and easily to your Crystal Enterprise Automated Process Scheduler (APS) from within Crystal Reports. When you use the Log on APS command on the File menu, Crystal Reports will detect the presence of Crystal Enterprise on your machine and display the appropriate APS folders in the Open/Save As dialog box.

This feature is also available in Crystal Reports Standard Edition.

Display web reports independent of server location

Use Relative URLs to share your reports and their associated report objects on any server, in any directory. Each report remains independent of its actual location, because Relative URLs eliminate dependence upon any particular server or virtual directory.

Extend reporting abilities with additional applications

With Crystal Enterprise Standard, use the ePortfolio, other samples, or any client application designed by your Administrator (using the SDK) to access shared reports.

Report Designer features

These enhancements are available in all editions of Crystal Reports, unless specified otherwise.

Create Report Alerts for instant notification

Set up Report Alerts to notify you when your report data hits a certain target, or to remind users to perform particular tasks. These custom messages, created in Crystal Reports, appear when specified conditions are met by report data. Because Report Alerts are specific to each report, you decide when to use them. And the message is created by you, so it can be specific to your data.

Export data to PDF, XML, RTF, and improved HTML

Export your reports—in whole or in part—directly to the popular Portable Document Format (PDF), and then distribute these reports by email, over the Web, or in print. Alternatively, export report data straight to XML, or take advantage of the new Rich Text Format (RTF) exporting feature, which is based on the Crystal Reports Encapsulated Page Format (EPF).

Note: In addition, Crystal Reports now supports page ranged export to PDF and RTF formats. Also, the performance of the HTML and DHTML export has been greatly improved.

Access additional data sources

With Crystal Reports' extended data access, you can now report directly off XML data and data stored in ACT2000 and Exchange 2000 (through ADO). Crystal Reports 8.5 also includes Merant 3.6 ODBC driver updates.

Build your own SQL queries with the Crystal SQL Designer, which now supports using native drivers as a data source (MS SQL Server, DB2, Oracle, Informix, and Sybase).

Note: Merant ODBC drivers, XML data access, and the Crystal SQL Designer are not available in Crystal Reports Standard Edition.

Copy report objects to other applications

You can now copy entire report objects, including charts and maps, to your Windows clipboard. This allows you to copy the data values from any report and paste them into other applications as bitmap images, text files, metafiles, and so on.

Summarize data across hierarchical groups

Make large, complex data sets easier to understand than ever by taking advantage of the new ability to create Subtotals, Grand Totals, and Summary fields for hierarchical groups.

Enhance formulas with new and improved functions

Alter report formatting by using new alerting functions in conjunction with Report Alerts. IsAlertEnabled( ), IsAlertTriggered( ), and AlertMessage( ) allow you to create conditional formulas that format report objects whenever the Report Alert's conditions are met.

To increase the options available when you're conditionally formatting cross tabs and OLAP grids, the GridRowColumnValue function has been extended and made more usable.

Customize default font options

Customize your default font settings to display fields, field titles, and text objects with fractional point sizes (10.5, 15.5, and so on). Just make the appropriate changes to the Fonts tab of the Options dialog box (available on the File menu).

Visualize data faster with improved charting and mapping

Explore the enhanced charting performance and improved interfaces when you're creating and formatting charts within your reports.

Select the new "Automatically set chart options" command, and the Chart Expert takes care of the finer details for you, so you can visualize your data faster than ever. You'll also find additional tooltips in both the Chart Expert and the Map Expert.

Analyze data types quickly

Find out what type of data resides in each of your database fields by clicking the new Show Field Type button in the Field Explorer.

Save preview pictures by default

Show users preview pictures of all your reports by saving a thumbnail image along with each report file. Crystal Reports does this automatically, with all your reports, if you select Save Preview Picture in the default Reporting option tab (File > Options > Reporting).

Developer features

These enhancements are specific to developers using Crystal Reports Developer Edition. If you purchased the Standard or Professional edition of Crystal Reports, you can download an evaluation copy of the latest Report Designer Component from `http://www.seagatesoftware.com`.

Embed the Crystal Reports Designer Control

The Embeddable Crystal Reports Designer Control is a new ActiveX control for the Report Designer Component (RDC). Developers can now embed this control, so their end users can design reports from within the developers' applications.

Utilize the unified RDC object model

The unified RDC object model (craxddrt.dll) is a combination of the RDC design time and run time object models. The newly unified model enables developers to use both the Embeddable Crystal Reports Designer ActiveX Control and features in the RDC run time object model at the same time.

Re-distribute the ASP Web Report Server

The enhanced ASP Web Report Server (rptserver.asp) is no longer dependent upon the Web Component Server, so developers can easily distribute their ASP web applications.

Access XML data at run time

Use the Report Designer Component to access XML data streams at run time and pass that information to objects in report applications.

Facilitate administration with the License Manager

Keep track of your Crystal Reports licenses and key codes with the License Manager (available only with Crystal Reports Developer Edition). This new administrative tool makes it easy for you to ensure that your product installation always meets your reporting needs. It also allows you to check the number of concurrent user licenses being used in the ASP environment. Access the License Manager from the Crystal Reports Tools program group on the Start menu.

Please note that the Report Designer Component will work only within the concurrent license limit when used as a reporting server.

Report Designer Component 8.5

Designed for Microsoft Visual Studio and other COM-based development environments, the feature-rich Report Designer Component 8.5 (RDC) gives developers unprecedented control over report layout and formatting for web and Windows application reporting.

Reflecting the enhanced features of the Crystal Report Engine, the RDC now supports most of the new Crystal Reports features, including PDF export, XML export and data access, alerting, summaries for hierarchical groups, and many more.

RDC run time distribution

The viewer has been enhanced to make it easier for developers to distribute their RDC applications.

Lotus Domino sample applications

If you're using Crystal Reports in conjunction with Lotus Domino, see the new Lotus Domino sample applications. These documents demonstrate how Crystal Reports increases the value of your Lotus Domino investment.

If you followed the default installation, you should find the sample documents in your Program directory (usually `C:\Program Files\Seagate Software\Crystal Reports\Samples\En\Code\Lotus Domino`). You can also serve these reports up from within a Lotus Notes application by opening either Crazev6.nsf or Xglobal.nsf from within Lotus Notes.

Additional information

Crystal Reports product news

`http://www.seagatesoftware.com/products/crystalreports/`

Crystal Reports demos

`http://www.seagatesoftware.com/products/crystalreports/showme/`

Crystal Enterprise product news

`http://www.seagatesoftware.com/products/crystalenterprise/`

Product information

`http://www.seagatesoftware.com/products/`

Developer Zone

`http://www.seagatesoftware.com/products/dev_zone/`

Online support, samples and tech briefs

`http://support.seagatesoftware.com/homepage/`

Training and consulting

`http://www.seagatesoftware.com/services/`

Seagate Software homepage

`http://www.seagatesoftware.com/homepage/`

Installing Crystal Reports 3

This chapter shows you how to install Crystal Reports locally from the product CD. It also describes how to install Crystal Reports to a network server, and how to install from a network server to a workstation machine. Additional topics include "Customizing your installation" on page 27 and "Installing Crystal Enterprise" on page 29.

Installing Crystal Reports

The Crystal Reports Installation Wizard works with Microsoft Windows Installer to guide you through the installation process. The Installation Wizard automatically recognizes your computer's operating system and updates files as required.

This chapter provides step-by-step instructions for installing Crystal Reports and shows how to customize your installation. The main topics are:

- Installation requirements
- Installing on a local machine from CD
- Installing to and from a network server
- Customizing your installation
- Installing Crystal Enterprise.

As one of the final steps in the installation process, you'll be asked if you want to register the product. Follow the on-screen instructions to complete this process.

Installation requirements

Local installation (from CD) requirements

- Microsoft Windows 95/98/2000, Windows NT 4.0, or higher
- Minimum RAM: 16 MB (32 MB for Windows NT)
- Recommended RAM: 32 MB
- Minimum hard drive space required (all editions): 60 MB
- Maximum hard drive space required:
  - Developer Edition: 350 MB
  - Professional Edition: 350 MB
  - Standard Edition: 155 MB.

We also recommend having an additional 100 MB of free disk space on your C: drive for use by Windows during the installation. If your system does not meet these requirements, the program may not run correctly.

Network installation requirements

- Microsoft Windows 95/98/2000, Windows NT 4.0, or higher
- Minimum RAM: 16 MB (32 MB for Windows NT)
- Recommended RAM: 32 MB
- Typical hard drive space required on a network server: 217 MB
- Typical hard drive space required on a workstation: 105 MB.

We also recommend having an additional 100 MB of free disk space on your C: drive for use by Windows during the installation. If your system does not meet these requirements, the program may not run correctly.

Installing on a local machine from CD

If you are installing Crystal Reports on a computer running Windows NT or Windows 2000, you must have Administrator privileges. The installation process creates registry entries and may update some system files that require Administrator rights.

Close all currently running programs and stop as many services as possible when installing Crystal Reports.

If you want to limit the features you install, see "Customizing your installation" on page 27.

Note: The Crystal Reports 8.5 installation removes the Web Component Server that was included in the Professional and Developer editions of Crystal Reports 8. To regain the ability to distribute reports over the Web, install Crystal Enterprise, which is included in the box with the Professional and Developer editions of Crystal Reports 8.5. For details, see "Installing Crystal Enterprise" on page 29.

To install on a local machine

1 Insert the Crystal Reports CD and, if the CD does not start automatically, browse to your CD-ROM drive and double-click **Setup.exe**.

  Note: Depending on the configuration of your current system, you may receive a dialog box informing you to update existing files. If this happens, click Yes and restart your machine. The Installation Wizard updates the required files.

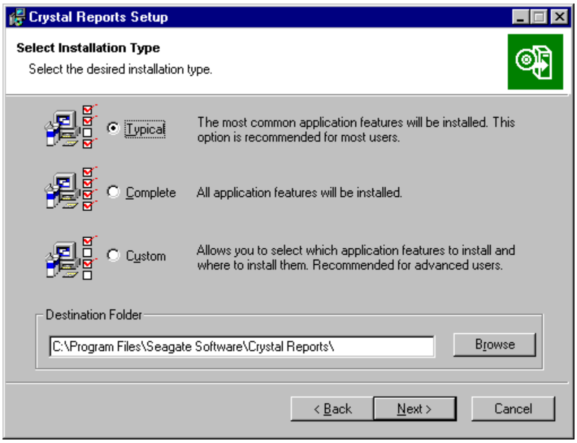
2 Read and accept the License Agreement to proceed with the installation.

3 In the User Information dialog box, type your name, organization, and the CD Key Code.

  Tip: The CD Key Code is printed on the sticker on the back of the CD envelope.

4 Click **Next**.

The Select Installation Type dialog box appears.



Crystal Reports Setup

Select Installation Type

Select the desired installation type.

● Typical — The most common application features will be installed. This option is recommended for most users.

○ Complete — All application features will be installed.

○ Custom — Allows you to select which application features to install and where to install them. Recommended for advanced users.

Destination Folder: C:\Program Files\Seagate Software\Crystal Reports\

5 Choose the type of installation that you want to perform:

- **Typical** installs the most common application features.
- **Complete** installs most application features and subcomponents.
- **Custom** enables you to choose the features that you want installed, to specify where they will be installed, and to check the disk space required by each feature. For details, see "Customizing your installation" on page 27.

6 Click **Browse** if you want to install Crystal Reports to a directory different from the default location.

The default is `C:\Program Files\Seagate Software\Crystal Reports\`

7 Click **Next**.

The Start Installation dialog box appears.

8 Click **Next** to begin copying files to your local drive.

Installing to and from a network server

A network installation of Crystal Reports involves two steps:

1 Run the Administrator's installation to copy files to a server machine on the network. See "Installing Crystal Reports to a network" on page 25.

2 Access the server machine from a workstation, and run Setup.exe to install Crystal Reports on the workstation. See "Installing Crystal Reports from a network" on page 26.

Installing Crystal Reports to a network

This procedure must be performed by a network administrator who has write access and network privileges. When this procedure is complete, end users will be able to access Setup.exe from the network to install Crystal Reports onto their local machines.

If you are installing Crystal Reports on a computer running Windows NT or Windows 2000, you must have Administrator privileges. The installation process creates registry entries and may update some system files that require Administrator rights.

Close all currently running programs and stop as many services as possible when installing Crystal Reports.

Note:

- The Crystal Reports 8.5 installation removes the Web Component Server that was included in the Professional and Developer editions of Crystal Reports 8. To regain the ability to distribute reports over the Web, install Crystal Enterprise, which is included in the box with the Professional and Developer editions of Crystal Reports 8.5. For details, see "Installing Crystal Enterprise" on page 29.
- If users do not have the Microsoft Windows Installer configured on their machines, the setup process detects the workstation's operating system and installs the appropriate Microsoft Windows Installer package.

To install Crystal Reports to a network

1. Initialize the Administrator's installation by running **Setup.exe** with the additional command line switch **/a**.

   For example, click the Start button, click Run, and type:

   ```
   <path> Setup.exe /a
   ```

   where `<path>` is the location of the Crystal Reports setup program.

2. In the Admin Installation dialog box, click **Browse** to choose the network location where you want to install the Crystal Reports files.

3. Click **Next**.

4. In the Admin Installation verification dialog box, click **Next** to begin copying the files to the network.

   When the files have finished copying to the network, users can double-click Setup.exe to begin the installation.

Installing Crystal Reports from a network

If your network administrator has installed Crystal Reports to the network, make sure you have read privileges to that network before beginning this process.

If you are installing Crystal Reports on a computer running Windows NT or Windows 2000, you must have Administrator privileges. The installation process creates registry entries and may update some system files that require Administrator rights.

Close all currently running programs and stop as many services as possible when installing Crystal Reports.

Note: The Crystal Reports 8.5 installation removes the Web Component Server that was included in the Professional and Developer editions of Crystal Reports 8. To regain the ability to distribute reports over the Web, install Crystal Enterprise, which is included in the box with the Professional and Developer editions of Crystal Reports 8.5. For details, see "Installing Crystal Enterprise" on page 29.

To install Crystal Reports from a network

1. Access the network server that contains the Crystal Reports installation files.
2. Double-click **Setup.exe**.

   Note: Depending on the configuration of your current system, you may receive a dialog box informing you to update existing files. Click Yes and restart your machine. The Installation Wizard updates the required files.
3. Read and accept the License Agreement to proceed with the installation.
4. In the User Information dialog box, type your name, organization, and the CD Key Code.

   Tip: You may need to contact your Administrator for the CD Key Code.
5. Click **Next**.

   The Select Installation Type dialog box appears.
6. Choose the type of installation that you want to perform:
   - **Typical** installs the most common application features.
   - **Complete** installs most application features and subcomponents.
   - **Custom** enables you to choose the features that you want installed, to specify where they will be installed, and to check the disk space required by each feature. For details, see "Customizing your installation" on page 27.

   Note: If you want to install certain features so they run from the network, choose the Custom installation.

7 Click **Browse** if you want to install Crystal Reports to a directory different from the default location.

The default is `C:\Program Files\Seagate Software\Crystal Reports\`

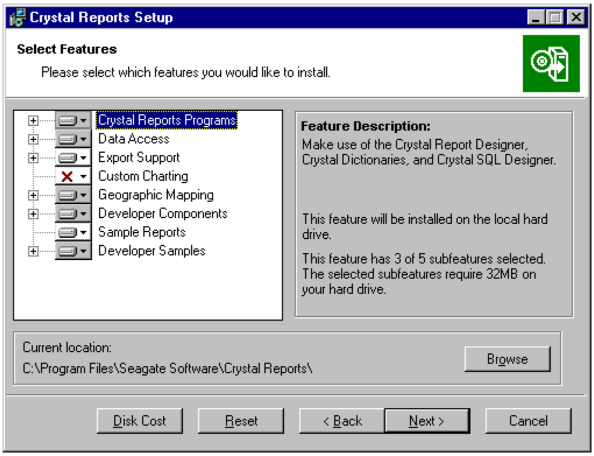
8 Click **Next**.

The Start Install dialog box appears.

9 Click **Next** to begin copying files to your local drive.

Customizing your installation

Selecting the Custom installation option invokes the Select Features dialog box, which allows you to install specific features, to change the default location of various features, and to check the amount of disk space required by each feature.



Crystal Reports Setup

Select Features

Please select which features you would like to install.

- Crystal Reports Programs
- Data Access
- Export Support
- ✗ Custom Charting
- Geographic Mapping
- Developer Components
- Sample Reports
- Developer Samples

Feature Description:
Make use of the Crystal Report Designer, Crystal Dictionaries, and Crystal SQL Designer.

This feature will be installed on the local hard drive.

This feature has 3 of 5 subfeatures selected. The selected subfeatures require 32MB on your hard drive.

Current location:
C:\Program Files\Seagate Software\Crystal Reports\

The icons in the feature tree indicate whether the feature and its subfeatures will be installed or not:

- A white icon means that the feature and all its subfeatures will be installed.
- A shaded icon means that the feature and some of its subfeatures will be installed.
- A red X means that the feature or subfeature is either unavailable or will not be installed.

To select the configuration and location of a feature or subfeature, click its icon.

Note: Each feature or subfeature can have its own configuration and location.

Use the following table to determine your installation options for each feature or subfeature:

Type of feature installation	Description of what is installed
Will be installed on local hard drive	■ Installs the feature on the local hard drive. ■ Uses the Typical install settings to install some of the feature's subfeatures to the local hard drive.
Entire feature will be installed on local hard drive	■ Installs the feature and all of its subfeatures on the local hard drive.
Will be installed to run from CD/network	■ Runs the feature and its Typical subfeatures off the CD/network. **Note:** Some subfeatures are not set up to run from the CD/network. These subfeatures will be installed on your local hard drive.
Entire feature will be installed to run from CD/network	■ Runs the feature and all of its subfeatures off the CD/network. **Note:** Some subfeatures are not set up to run from the CD/network. These subfeatures will be installed on your local hard drive.
Feature will be installed when required	■ Installs the feature or subfeature from the CD/network when first used. **Note:** This option is available only for the Crystal Reports Programs feature and its subfeatures: Crystal Report Designer, Crystal Dictionaries, and Crystal SQL Designer.
Entire feature will be unavailable	■ Neither the feature nor its subfeatures are installed.

Note: Subfeatures are listed below each feature. A subfeature can have a different type of installation than its parent feature.

Installing Crystal Enterprise

Crystal Enterprise replaces Crystal Report's web capabilities and introduces new features such as report scheduling, security, and scalability.

The Crystal Enterprise Standard CD (shipped with Crystal Reports 8.5) includes a temporary product license keycode that lets users install and operate the Crystal Enterprise system for a period of 30 days. To obtain a free, non-expiring product activation license keycode, users must register their copy of Crystal Enterprise Standard. Please use the Crystal Registration Wizard located in the Crystal Enterprise Program Group to complete your registration.

The version of Crystal Enterprise included with the Professional and Developer editions of Crystal Reports 8.5 installs all the components necessary for running Crystal Enterprise on one machine.

The following components are installed when you install Crystal Enterprise:

- ePortfolio
- Crystal Publishing Wizard
- Crystal Management Console (CMC)
- Crystal Configuration Manager
- Crystal Import Wizard
- Crystal Web Component Server (WCS)
- Crystal Automated Process Scheduler (APS)
- Crystal Cache Server
- Crystal Page Server
- Crystal Input File Repository Server
- Crystal Output File Repository Server
- Crystal Job Server.

All services are enabled after the installation is complete. An Administrator account is created as well as a guest user account. Sample reports are also installed.

Getting Started

Crystal Enterprise uses a wizard to guide you through the installation process. The installation wizard automatically recognizes your computer’s operating system and updates files as required.

System requirements

A stand-alone installation of Crystal Enterprise has the following minimum system requirements:

- Microsoft Windows NT 4 SP5, Windows 2000
- Microsoft Internet Information Services (IIS) 3, Netscape Enterprise Server 3.6 (and the latest SP), Netscape FastTrack 3.01 (and the latest SP), or a web server that supports the Common Gateway Interface (CGI)
- 128 MB RAM minimum
- 400 MB free disk space
- Microsoft Internet Explorer 4.01 SP2 minimum, 5.5 SP1 recommended, Netscape 4.72 minimum (4.75 recommended).

If your system does not meet these requirements, the program may not run correctly.

Running the installation

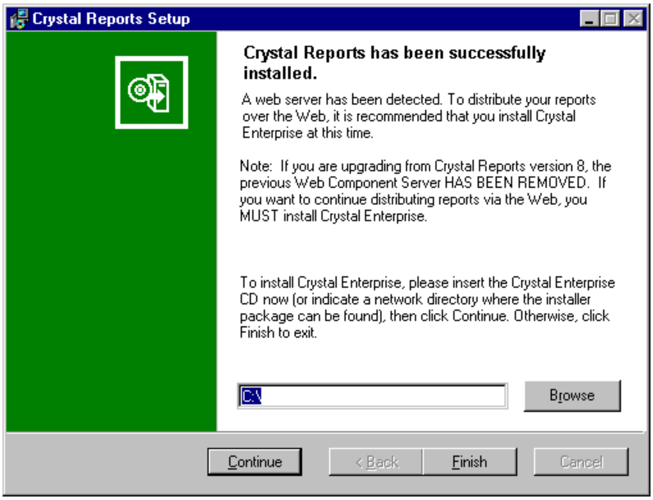
This section describes two options for installing Crystal Enterprise:

- “Installing Crystal Enterprise immediately after Crystal Reports” on page 31
- “Installing Crystal Enterprise at a later time” on page 32.

The first option can be used if you receive a message after installing or upgrading Crystal Reports; the second option can be used at any time.

Installing Crystal Enterprise immediately after Crystal Reports

If the computer you're installing or upgrading Crystal Reports on has a web sever, the installation program prompts you with the following dialog box:



Crystal Reports has been successfully installed.

A web server has been detected. To distribute your reports over the Web, it is recommended that you install Crystal Enterprise at this time.

Note: If you are upgrading from Crystal Reports version 8, the previous Web Component Server HAS BEEN REMOVED. If you want to continue distributing reports via the Web, you MUST install Crystal Enterprise.

To install Crystal Enterprise, please insert the Crystal Enterprise CD now (or indicate a network directory where the installer package can be found), then click Continue. Otherwise, click Finish to exit.

C:\

If you want to install Crystal Enterprise now, insert the Crystal Enterprise CD, or enter the path of the network directory you want to run the installer from. To continue, go to step 2 of "Installing Crystal Enterprise at a later time."

If you don't want to install Crystal Enterprise now, click Finish to exit the dialog box. You can install later by using the "Installing Crystal Enterprise at a later time" procedure.

Installing Crystal Enterprise at a later time

1 Insert the Crystal Enterprise CD and, if the CD does not start automatically, browse to your CD-ROM drive and double-click **Setup.exe**.

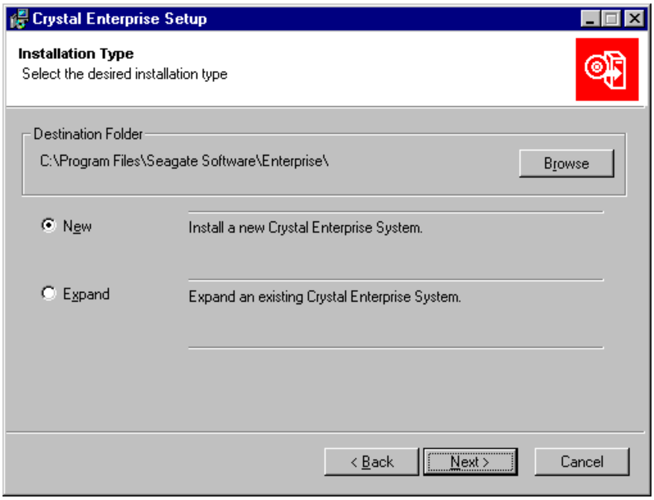
The Wise Installation Wizard starts the installation and the Crystal Enterprise Setup program searches for previously installed components.

2 Follow the instructions displayed on your screen regarding licensing, and so on.

3 In the User Information dialog box, enter your name, organization name, and license key.

4 Click **Next**.

The Installation Type dialog box appears:



Crystal Enterprise Setup

Installation Type

Select the desired installation type

Destination Folder

C:\Program Files\Seagate Software\Enterprise\

● New — Install a new Crystal Enterprise System.

○ Expand — Expand an existing Crystal Enterprise System.

5 Click **Browse** if you'd like to specify a different destination folder from the one chosen by default.

Search for the new folder on your computer.

6 Select **New** as your installation type.

7 Click **Next**.

8 Click **Next** on the Start Installation dialog box when you are ready to begin the installation.

The installation of files begins immediately. When the installation program has finished copying files, the final setup screen appears.

9 Clear **Launch Report Publish Wizard** if you don't want the wizard to begin immediately.

10 Click **Finish**.

Quick Start 4

This chapter begins with an overview of the sample reports and data commonly used with Crystal Reports. It then presents an overview of the Report Experts—both new and advanced users can benefit from looking at the Standard Report Expert as it has many steps in common with the other Report Experts. Finally, the chapter offers two tutorials: one for new users and another for more advanced users.

Learning how to use Crystal Reports

You can teach yourself how to use Crystal Reports by choosing from the methods available in this chapter:

- you can study the sample reports and sample database included with Crystal Reports
- you can use the detailed descriptions and instructions in the "Quick start for new users" on page 39
- you can use the summaries and topic cross-references in the "Quick start for advanced users" on page 61—especially useful if you're already familiar with reporting concepts.

Each method is a helpful way to learn and understand Crystal Reports and, although any one might be enough to get you up and running, you can always come back to this chapter and consult the other methods as you need them.

Sample reports

Crystal Reports comes with three kinds of professionally designed sample reports, described below. By default, these samples are installed in your Program directory (usually: `\Program Files\Seagate Software\Crystal Reports\Samples\En\Reports`). The Lotus Domino samples are installed at: `\Program Files\Seagate Software\Crystal Reports\Samples\En\Code\Lotus Domino`).

Lotus Domino

The Lotus Domino folder includes sample reports for those using Crystal Reports with Lotus Domino. The samples include:

- lotusmap.rpt
- world wide sales.rpt

You can use these reports in a Lotus Notes application by opening either Crazev6.nsf or Xglobal.nsf from within Lotus Notes.

Feature Examples

The Feature Examples folder includes a selection of reports designed to illustrate basic concepts and common report functions. These samples include:

- chart.rpt
- crosstab.rpt
- date range formula.rpt
- formatting.rpt
- formulas.rpt
- group by first letter.rpt

- group by intervals.rpt
- group.rpt
- hierarchical grouping.rpt
- hyperlinks.rpt
- OLAP cube report.rpt
- percentages.rpt
- record selection1.rpt
- record selection2.rpt
- running totals group.rpt
- running totals.rpt
- sort multiple field.rpt
- sort.rpt
- subtotal group.rpt
- subtotal multiple level.rpt
- summary group.rpt
- summary with formula.rpt.

General Business

The General Business folder includes samples that cover a wide range of reporting needs for many of the departments in an organization.

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.

Sample data - Xtreme.mdb

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

Xtreme.mdb is a database that contains data for Xtreme Mountain Bikes, a fictitious manufacturer of mountain bikes and accessories. The database includes the following tables:

- **Credit**
  Information from customer credit memos, such as credit authorization IDs and amounts.

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

Xtreme.mdb also includes:

- a Select query (Top Customers)
- a Parameter query (Credit_Limits)
- an Update query (Update Query).

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.

Report Experts

The tutorials in this chapter show you how to build a report from scratch. As a complement or an alternative, however, you may want to use the Report Experts available through the Crystal Report Gallery. There are eight Report Experts:

- Standard
- Form Letter
- Form
- Cross-Tab

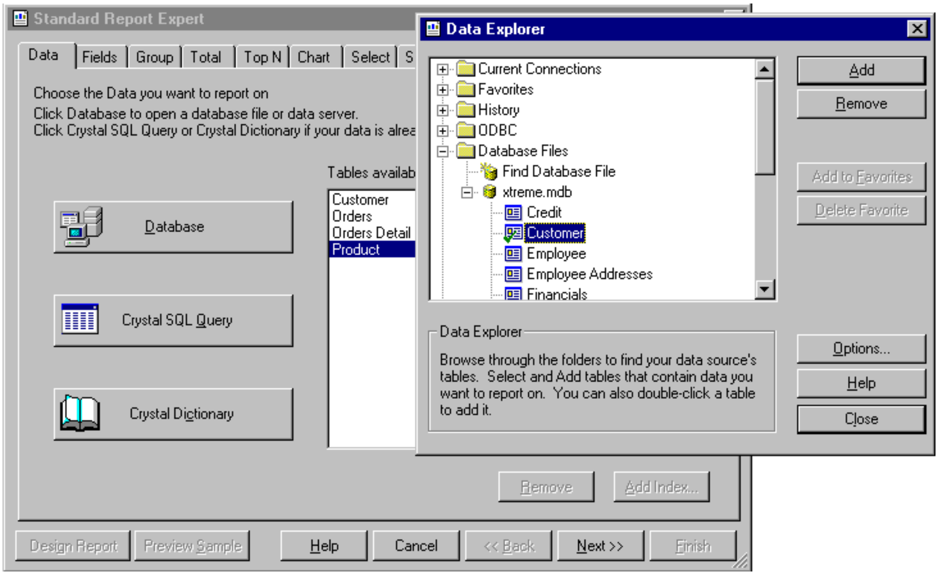
- Subreport
- Mail Label
- Drill Down
- OLAP.

Each Expert guides you through the creation of a report by providing a series of tabs. Many of the experts have tabs unique to a specific type of report. For example, the Mail Label Expert has a tab that allows you to specify the type of mailing label you want to use.

Standard

The Standard Report Expert is the most generic of the experts. It provides eight tabs, many of which are common to the other Report Experts. The Standard Report Expert guides you through choosing a data source and linking database tables. It also helps you add fields and specify the grouping, summarization (totals), and sorting criterion you want to use. Finally, the Standard Report Expert leads you through chart creation and record selection.

The Style tab contains pre-defined layouts for you to apply to your report to give it more impact.



Choose the Data you want to report on
Click Database to open a database file or data server.
Click Crystal SQL Query or Crystal Dictionary if your data is alrea

Browse through the folders to find your data source's tables. Select and Add tables that contain data you want to report on. You can also double-click a table to add it.

Form Letter

The Form Letter Report Expert offers an easy solution for creating form letters that use your database as a source of client information. The expert contains much of the functionality of the Standard Report Expert, but has a Form Letter tab that lets you define the text and database fields that will appear in each section of your letter. You can also use the Form Letter tab to import text you have created in another application.

Form

The Form Report Expert lets you create reports that can be printed on pre-defined forms (company invoices, statements, and so on). This expert is much like the Standard Report Expert, but features a Form tab where you choose images of your company logo or forms. Logos and forms can be applied to header or details sections so your report data prints in the appropriate location on your report.

Cross-Tab

The Cross-Tab Report Expert guides you through the creation of a report in which your data is displayed as a cross-tab object. Three special tabs (Cross-Tab, Style, and Customize Style) help you create and format the cross-tab itself.

Subreport

The Subreport Expert lets you create a main report and a subreport at the same time. The expert offers all the Standard Report Expert's flexibility for creating the main report. On the additional Subreport tab, you can choose an existing report to use as the subreport, or you can create a new report. New subreports are created by repeating the steps used in the Standard Report Expert.

Mail Label

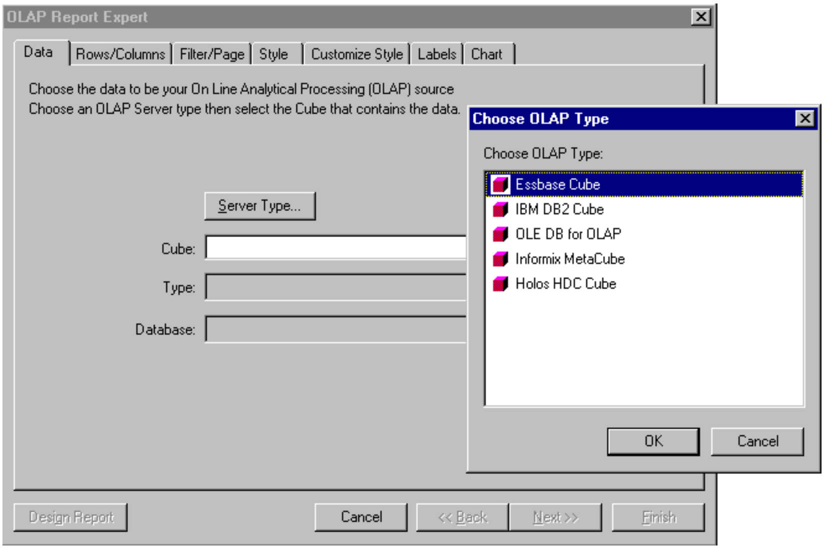
The Mailing Labels Report Expert lets you create a report that is formatted to print on any size mailing label. You can use the Label tab to select a commercial label type, or you can define your own layout of rows and columns for any multi-column style report.

Drill Down

The Drill Down Report Expert guides you through the creation of a report that hides sections and makes them available for viewing only though drill down. The expert offers all the functionality of the Standard Report Expert. The Drill tab displays a list of sections that can be hidden. Hidden sections don't appear on your report until you click the appropriate field. Hidden data can be used for summaries and totals.

OLAP

The OLAP Report Expert lets you create a report in which your OLAP data is displayed as a grid object. Although similar to the Cross-Tab Report Expert in several ways, the OLAP Report Expert appears to be different due to the requirements of working with OLAP data sources. You first specify the location of you OLAP data, and then you choose the dimensions you want to include in the grid. Next you filter the report data and choose and customize the style of the grid object. Finally, you can define labels for your grid and insert a chart, if you wish.



Choose the data to be your On Line Analytical Processing (OLAP) source
Choose an OLAP Server type then select the Cube that contains the data.

This is only a brief overview of the eight Report Experts available in Crystal Reports.

Quick start for new users

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

In this tutorial, you will get an introduction to the program 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.

You begin by learning the basic concepts: selecting a database, placing some fields on the report, and then selecting specific records to be included. You will then learn how to:

- insert and move database fields
- add and format a title

- display a report in the Preview tab so you can fine-tune your work
- use the Select Expert to ensure the report includes only the data you need
- move objects
- group and sort data
- insert pictures
- print a report.

Before you begin

This tutorial assumes you are familiar with Microsoft Windows and uses conventional terms and procedures common to the Windows environment. If you are not familiar with Windows, please refer to the documentation that came with Microsoft Windows for further explanation.

The default font for all report sections in the program is set to Times New Roman, 10 point. If you have changed the default font, or if your printer does not support this font, the field size, field spacing, and screen shots will look different than those included in this tutorial.

This tutorial has been designed using Microsoft Windows 95, Windows 98 and Windows NT. Screen shots may vary slightly if you are using a different platform.

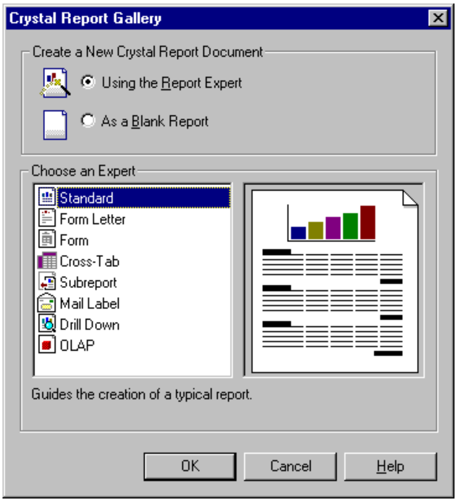
If you are not familiar with the Crystal Reports environment, review "Formatting" on page 199, which describes working with the grid, free-form placement, using guidelines, and formatting activities.

Creating the report

1. Click **New** on the Standard toolbar.

   The Crystal Report Gallery appears.

   The gallery contains a number of experts to guide you through the creation of specific types of reports. Since you will 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.



Crystal Report Gallery

Create a New Crystal Report Document

● Using the Report Expert

○ As a Blank Report

Choose an Expert

Guides the creation of a typical report.

2 Select **As a Blank Report** on the Crystal Report Gallery dialog box and click **OK**.

The Data Explorer dialog box appears.

Note: You can also create reports based on SQL/ODBC data sources, dictionary files, or query files. See "The Crystal SQL Designer" on page 621, and "Dictionaries" on page 639, for more information.

Selecting a database to use

The next step in creating a report is to select a database. Select the Xtreme.mdb sample database for this tutorial.

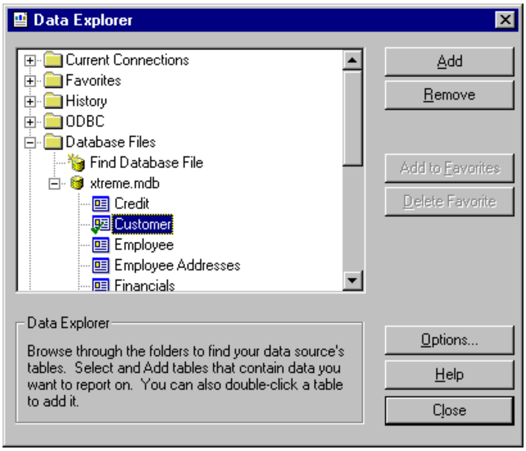
To select a database

1 In the Data Explorer dialog box, select **Xtreme.mdb** from **Database Files**.

By default, this file was installed in the `\Program Files\Seagate Software\Crystal Reports\Samples\En\Databases` directory.

Note: You may have to use the Find Database File option to find the Xtreme database. If you wish to see database and server properties, right-click the database in the Data Explorer and select Properties from the shortcut menu.

2 Expand the **Xtreme.mdb** folder to see a list of tables.



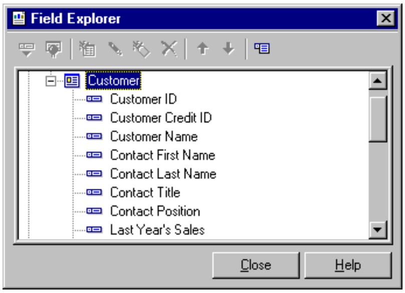
Data Explorer

Browse through the folders to find your data source's tables. Select and Add tables that contain data you want to report on. You can also double-click a table to add it.

Because you are dealing only with customers in this tutorial, you will select the Customer table.

3 Select Customer and click **Add** and then **Close**.

The Field Explorer dialog box appears with "Database Fields" selected.



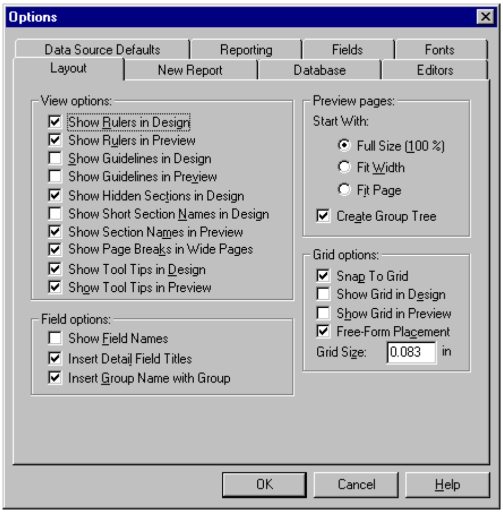
Note: If more than one table is selected in the Data Explorer dialog box, the Visual Linking Expert appears. For more information on linking, see "Linking multiple tables" on page 90.

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 in which you are working, simply look at the shaded area to the left of the report which always displays either the section names or the initials that designate the names. See "Design tab" on page 82.

If the Show Short Section Names in Design check box is selected in the Options dialog box, then the Report Header, Page Header, Details, Report Footer and Page Footer section names will appear as RH, PH, D, RF and PF respectively. If this check box is not selected, follow these steps:

1. On the **File** menu, click **Options**. The Options dialog box appears with the Layout tab active.



View options:
☑ Show Rulers in Design
☑ Show Rulers in Preview
☐ Show Guidelines in Design
☐ Show Guidelines in Preview
☑ Show Hidden Sections in Design
☐ Show Short Section Names in Design
☑ Show Section Names in Preview
☑ Show Page Breaks in Wide Pages
☑ Show Tool Tips in Design
☑ Show Tool Tips in Preview

Field options:
☐ Show Field Names
☑ Insert Detail Field Titles
☑ Insert Group Name with Group

Preview pages:
Start With:
● Full Size (100 %)
○ Fit Width
○ Fit Page
☑ Create Group Tree

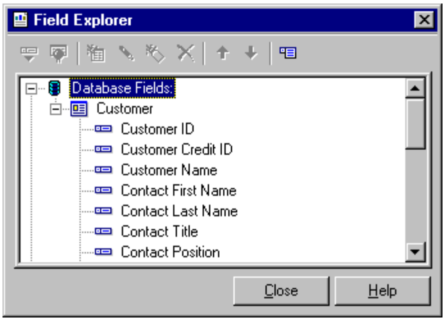
Grid options:
☑ Snap To Grid
☐ Show Grid in Design
☐ Show Grid in Preview
☑ Free-Form Placement
Grid Size: 0.083 in

2. Select the **Show Short Section Names in Design** check box.

3. Click **OK** to return to the report.

Inserting a field

The Field Explorer dialog box appears with "Database Fields" selected since you will want to insert database fields when you create a new report.



Database Fields
Customer
Customer ID
Customer Credit ID
Customer Name
Contact First Name
Contact Last Name
Contact Title
Contact Position

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

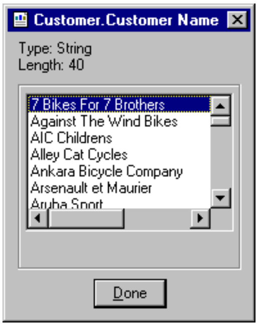
You can move the dialog box to another location on the screen by clicking on its title bar and dragging it to a new location. You can also resize the dialog box by dragging any of its edges with the Resizing cursor.

You will now start placing objects on the report by inserting the Customer Name field.

To insert a field

1 Highlight a field name in the Field Explorer 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 right-clicking and selecting Browse from the shortcut menu. The Browse dialog box appears listing the field name, type, length, and a subset of field values.



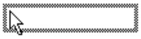
Customer.Customer Name

Type: String
Length: 40

7 Bikes For 7 Brothers
Against The Wind Bikes
AIC Childrens
Alley Cat Cycles
Ankara Bicycle Company
Arsenault et Maurier
Aruba Sport

2 Click the **Customer Name** field and drag it into the **Details** section of the report.

An object frame appears with the Arrow cursor as you drag the field onto the report:



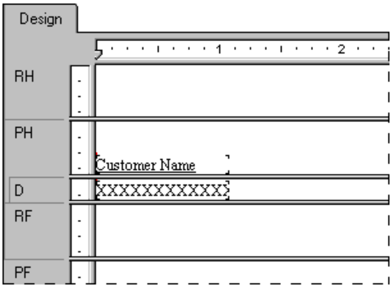
- the object frame represents the object you have just selected for placement
- the size of the object frame approximates the size of the data in the field selected.

3 Move the object frame as far to the left as you can in the Details section.

As you move the object frame, guidelines that extend to the rulers appear to help you place the object exactly where you want it.

If you move the field too far to the left, the Arrow cursor turns 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.

The Design tab should look similar to this:



Understanding fields

Before going any further, take a look at the field you just placed in the Details section:

- First of all, the object frame indicates that when the report is printed, a field value will appear where the box is positioned.
- The X's in the object frame indicate that the database field contains a text string. Other data types have different character representations. For example, a currency data type is represented by $55,555.56.

  Note: The X's appear only if you haven't selected the Show Field Names check box on the Layout tab of the Options dialog box.
- The number of X's in the object frame is the data width, the maximum number of characters in the field as defined by the database. The width of the object frame 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. Later in this tutorial you will learn how to make changes to these properties.
- The line spacing is adjusted to the point size selected for the characters in the field.

Adding additional fields

Next, you will insert two additional fields in the report. This time, however, you will use the Ctrl-click combination to add them at the same time.

To add additional fields

1. Highlight the **City** field in the Field Explorer dialog box, press the Ctrl key on your keyboard, and then highlight the **Country** field. Release the Ctrl key.

   If you scroll through the field list, you will notice that both fields remain selected.

   Note: Using the Ctrl-click combination allows you to select a non-continuous range of fields. The Shift-click combination can be used to select several fields from the list that are continuous.

2. Drag the fields to place them.

   As the cursor is moved over the report, an object frame appears along with the Arrow cursor.

3. 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 the report in the same order in which they are listed in the Field Explorer dialog box.

Selecting fields

When a field is selected, the object frame 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 object frame and click once. The handles appear, indicating the object is selected.
- Move the cursor away from the object frame and click in an empty part of the window. The handles disappear.

That’s all it takes to select and deselect objects.

Resizing fields

To resize the field, follow these steps:

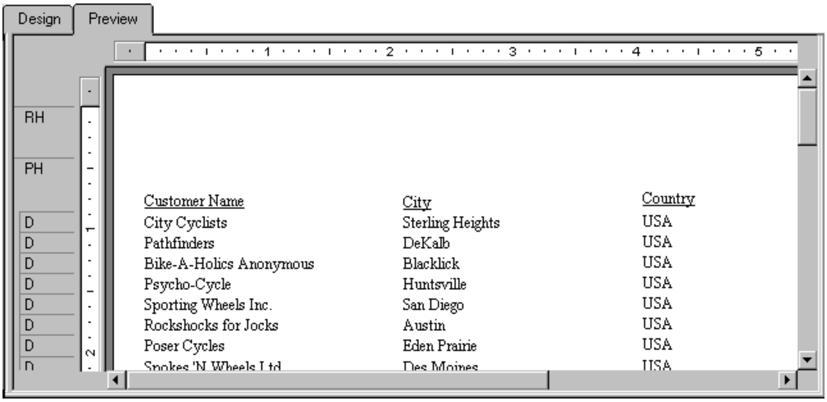
1. Click the **Customer Name** field in the Details section to select it.
2. Press the Ctrl key and click the field heading to select both objects.
3. Move the cursor over the resizing handle on the right edge of the fields until the cursor turns into a Resizing cursor.
4. Resize the fields to the left until they are approximately two inches in length.

Reviewing your work

Now let's see how the report looks with three fields placed and positioned.

1. Click **Print Preview** on the Standard toolbar to activate the **Preview** tab.

   The screen should look similar to this:



Customer Name	City	Country
City Cyclists	Sterling Heights	USA
Pathfinders	DeKalb	USA
Bike-A-Holics Anonymous	Blacklick	USA
Psycho-Cycle	Huntsville	USA
Sporting Wheels Inc.	San Diego	USA
Rockshocks for Jocks	Austin	USA
Poser Cycles	Eden Prairie	USA
Spokes 'N Wheels Ltd.	Des Moines	USA

   Note: The first time you preview a report, you must click Preview on the Standard toolbar to activate the Preview tab. The Preview tab appears to the right of the Design tab. You can then switch between designing and previewing the report by clicking the corresponding tab.

   You have the beginnings of a customer list report, but you still have several fields to add.

2. When you are finished reviewing the report, return to the **Design** tab by clicking it.

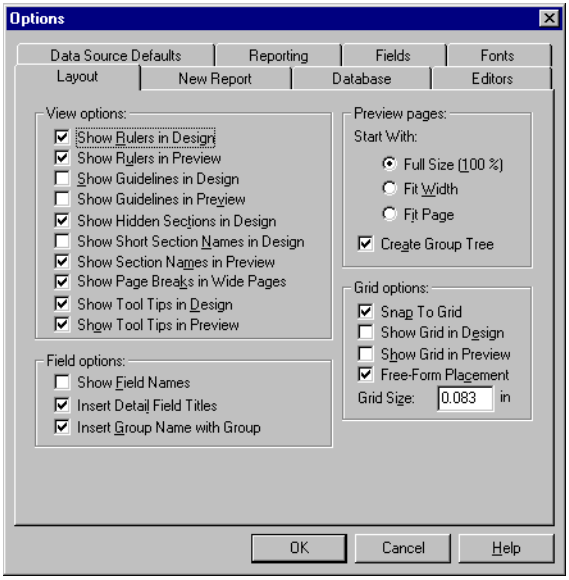
Displaying field names

Field pictures (object frames containing character representations) have been discussed, but there may be times when you want to see the field names in the Design tab.

To display field names

1 On the **File** menu, click **Options**.

The Options dialog box appears with the Layout tab active.



View options:
☑ Show Rulers in Design
☑ Show Rulers in Preview
☐ Show Guidelines in Design
☐ Show Guidelines in Preview
☑ Show Hidden Sections in Design
☐ Show Short Section Names in Design
☑ Show Section Names in Preview
☑ Show Page Breaks in Wide Pages
☑ Show Tool Tips in Design
☑ Show Tool Tips in Preview

Field options:
☐ Show Field Names
☑ Insert Detail Field Titles
☑ Insert Group Name with Group

Preview pages:
Start With:
● Full Size (100 %)
○ Fit Width
○ Fit Page
☑ Create Group Tree

Grid options:
☑ Snap To Grid
☐ Show Grid in Design
☐ Show Grid in Preview
☑ Free-Form Placement
Grid Size: 0.083 in

2 Select the **Show Field Names** check box and click **OK**.

Now in the Design tab you will see the actual field names instead of character representations (X, $, #, etc.).

Combining database fields in a 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. This allows you to control the formatting of both fields by making changes to only one object. When you insert fields in a text object, the fields are automatically trimmed (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.

To combine fields in a text object

ab

1 Click **Insert Text Object** on the Standard toolbar. As you move the cursor over the report, an object frame appears next to it.

2 Insert the field to the right of the fields in the **Details** section.

As you drag the field, the Design tab automatically scrolls to the right, if necessary. When you click to place the object, a text object appears and the horizontal ruler of the Design tab changes to a ruler/tab selector that is used for editing the text object. If you click an empty area of the report or a field object, the standard Design tab ruler appears.

3 Click once on the border of the text object to select it for resizing. Handles appear on all sides of the object.

4 Move the cursor over the right sizing handle of the text object and increase the width by about 1 inch. You may need to scroll to the right and continue resizing.

5 Double-click inside the text object to select it for editing. Notice the insertion point is now flashing within the text object.

6 Highlight the **Contact Last Name** field in the Field Explorer dialog box. Remember, you can move the Field Explorer dialog box by dragging and dropping it by its title bar.

7 Drag the field to the text object.

8 Move the cursor over the text object until the cursor becomes a Drag and Drop cursor.

9 Click once to place the field in the text object. The cursor now appears after the **Contact Last Name** field, within the text object.

10 Type a comma and a space after the Contact Last Name.

11 In the Field Explorer dialog box, highlight the **Contact First Name** field.

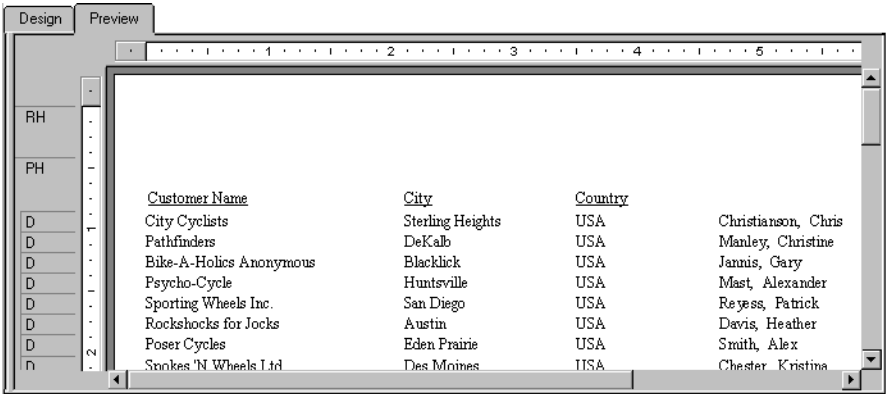
12 Drag the field to the text object.

13 Move the cursor over the text object until the 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 field will be inserted to the right of the comma and space.

14 You are finished inserting fields, so **Close** the Field Explorer dialog box.

15 Click the **Preview** tab to look at the fields you just placed.

The report should now look similar to this:



Customer Name	City	Country	
City Cyclists	Sterling Heights	USA	Christianson, Chris
Pathfinders	DeKalb	USA	Manley, Christine
Bike-A-Holics Anonymous	Blacklick	USA	Jannis, Gary
Psycho-Cycle	Huntsville	USA	Mast, Alexander
Sporting Wheels Inc.	San Diego	USA	Reyess, Patrick
Rockshocks for Jocks	Austin	USA	Davis, Heather
Poser Cycles	Eden Prairie	USA	Smith, Alex
Spokes 'N Wheels Ltd	Des Moines	USA	Chester, Kristina

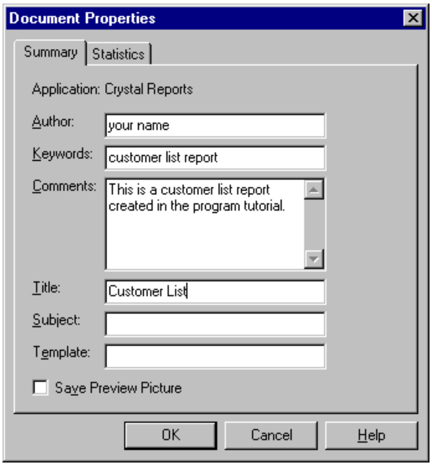
Adding summary information

The next step is to add summary information to your report. Adding summary information allows you to specify the author, title, and subject of the report, as well as any keywords or comments related to the report. If a template is used when creating the report, you can specify that as well. When you add summary information, users can find information related to the report quickly.

To add summary information

1 On the **File** menu, click **Summary Info**.

The Document Properties dialog box appears with the Summary tab active.



Application: Crystal Reports

Author: your name

Keywords: customer list report

Comments: This is a customer list report created in the program tutorial.

Title: Customer List

Subject:

Template:

□ Save Preview Picture

2. Enter information about your report in the text boxes provided. Be sure to enter the title “Customer List” in the **Title** text box. This information will be used in the next section of the tutorial.
3. Click **OK** when finished.

Adding a title

As you can see, the report looks incomplete without a title. Although you can add a title using a text object, you can also tell the program to take the title information directly from the Title text box in the Document Properties dialog box.

To add a title

1. Click the **Design** tab.
2. On the **Insert** menu, click **Field Object**.
3. In the Field Explorer, scroll down to **Special Fields** and expand it.
4. Choose **Report Title**.
5. Move the cursor over the report. An object frame appears.
6. Position the object frame in the upper left-hand corner of the Page Header (PH) section of the report and click once to place the object.
7. Click the **Preview** tab to review your changes.

The report title object now displays the title that you entered in the Title text box of the Document Properties dialog box.

Formatting objects

Now you can format the report title. This time, however, you will remain in the Preview tab to do the work. This will make it easier to see your work while you are formatting the title.

To format an object

1. To center the title, you will first need to expand the **Title** field box so that it's about the same width as the data in your report. To do this, select the object by clicking it.
2. Position the cursor on the right edge of the object 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.

   You have created a large field that extends from the left edge to the right edge of the report.
3. With the report title object still selected, click **Align Center** on the Formatting toolbar. The title is centered within the object.

4 Right-click the object and choose **Font** from the shortcut menu. The Format Editor appears with the **Font** tab active.

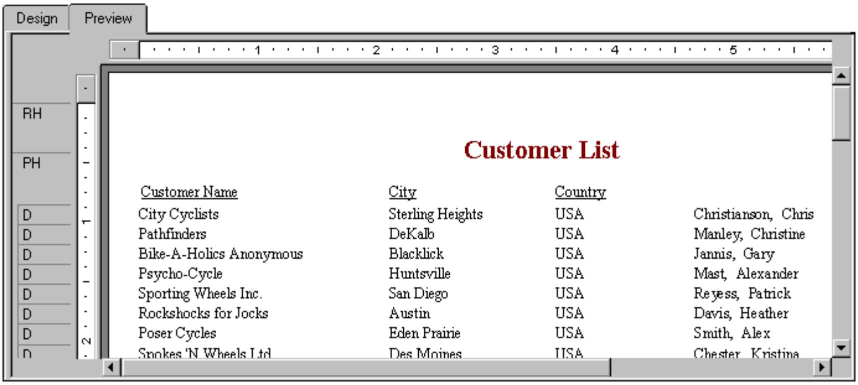
5 Set the report title to a larger, bolder version of the active font by selecting **Bold** from the **Style** list and **16** (or a point size suitable to the font you are using) from the **Size** list.

6 Change the color of the text by selecting **Maroon** from the **Color** list. Notice that the **Sample** box shows an example of how the text will look.

7 Click **OK** when finished.

8 Resize the report title object vertically to accommodate the increased size of the title.

The title is now formatted to stand out on the report.



Customer List

Customer Name	City	Country	
City Cyclists	Sterling Heights	USA	Christianson, Chris
Pathfinders	DeKalb	USA	Manley, Christine
Bike-A-Holics Anonymous	Blacklick	USA	Jannis, Gary
Psycho-Cycle	Huntsville	USA	Mast, Alexander
Sporting Wheels Inc.	San Diego	USA	Reyess, Patrick
Rockshocks for Jocks	Austin	USA	Davis, Heather
Poser Cycles	Eden Prairie	USA	Smith, Alex
Spokes 'N Wheels Ltd	Des Moines	USA	Chester, Kristina

Adding a field heading

As you can see, the Contact Name field is the only field without a heading. In this section you will create a heading using a text field.

To add a field heading

1 Click the **Design** tab to return to it.

2 Click **Insert Text Object** on the Standard toolbar. As you move the cursor over the report, an object frame appears next to it.

3 Place the object in the Page Header (PH) section, above the contact name object.

4 Type "Contact Name" (without the quotation marks) in the text object edit box.

5 While the text object is still in edit mode, highlight the field heading.

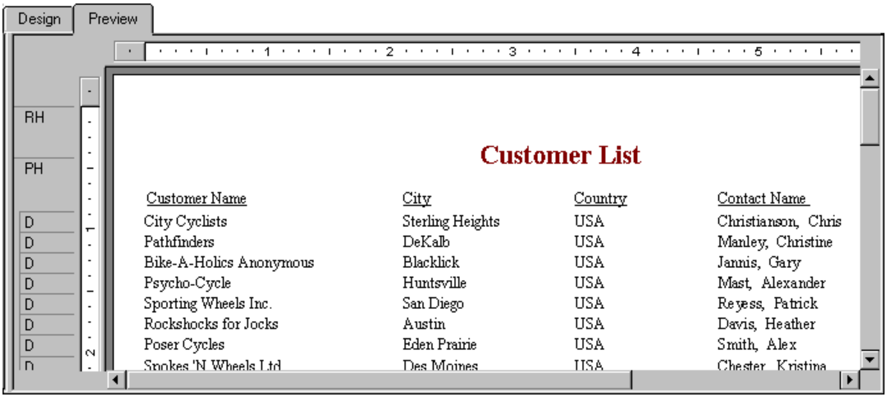
6 Click **Underline** on the Formatting toolbar.

The Contact Name field now has a heading that looks just like the other field titles.

Saving the report

1. Click **Save** on the Standard toolbar to save your work. Since this is the first time you are saving the report, the Save As dialog box appears displaying the default directory where the file will be saved.
2. Type Custlist.rpt in the **File name** box and click **Save**. Your report is saved to the default directory or another directory you chose.
3. Click the **Preview** tab to view the report.

The report should now look similar to the following:



Customer List

Customer Name	City	Country	Contact Name
City Cyclists	Sterling Heights	USA	Christianson, Chris
Pathfinders	DeKalb	USA	Manley, Christine
Bike-A-Holics Anonymous	Blacklick	USA	Jannis, Gary
Psycho-Cycle	Huntsville	USA	Mast, Alexander
Sporting Wheels Inc.	San Diego	USA	Reyess, Patrick
Rockshocks for Jocks	Austin	USA	Davis, Heather
Poser Cycles	Eden Prairie	USA	Smith, Alex
Spokes 'N Wheels Ltd.	Des Moines	USA	Chester, Kristina

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

Record Selection

Crystal Reports 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 the report
- save a report, including the 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. It is easy to restrict lists like this using the Select Expert. See "Record Selection" on page 125.

Entering the selection criteria

When you scroll through the report, you will see that it contains information for customers from many different countries. In this step, you will limit the number of countries displayed to the USA.

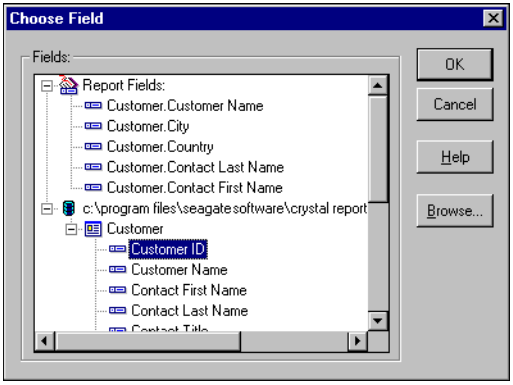
To enter selection criteria

1 To begin, click the **Design** tab to return to design mode.

2 Click an empty area of the report to make sure all fields are deselected.

3 Click **Select Expert** on the Standard toolbar.

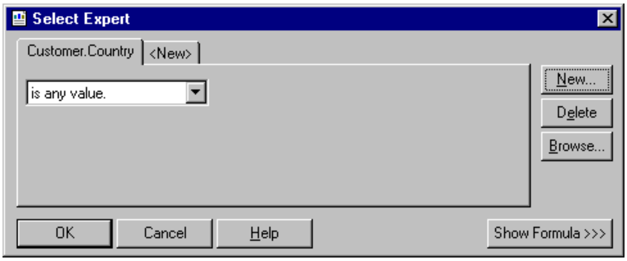
The Choose Field dialog box appears.



This dialog box lists all the fields currently in the report in the Report Fields section and then lists all fields that are available from each table in the database fields section.

4 Since you are going to base record selection on the country field, highlight Country in the **Fields** list and click **OK**.

The Select Expert appears.



Imagine that you are completing the following 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*, which implies there is no restriction on the record selection.

5 Click the arrow on the operators box to see what other options you have. Since you want only those records where the Country is USA, select the **is equal to** condition.

A new box appears on the right. The dialog box sentence now reads:

```
Select all records where a customer’s country is equal to
```

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

6 Click the arrow on the empty list. A list of all the country values appears. Select **USA** from the list.

The sentence now reads:

```
Select all records where a customer’s country is equal to USA
```

7 Click **OK** to return to the **Design** tab.

8 Click the **Preview** tab to review the results of your work.

The Change in Record Selection Formula Detected dialog box appears.

9 Click **Refresh Data**.

10 Save this version of the report without overwriting the original report by choosing **Save As** from the File menu and giving the new report the name USA.rpt.

Congratulations! You have started formatting your report and have 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.

Deleting a field

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

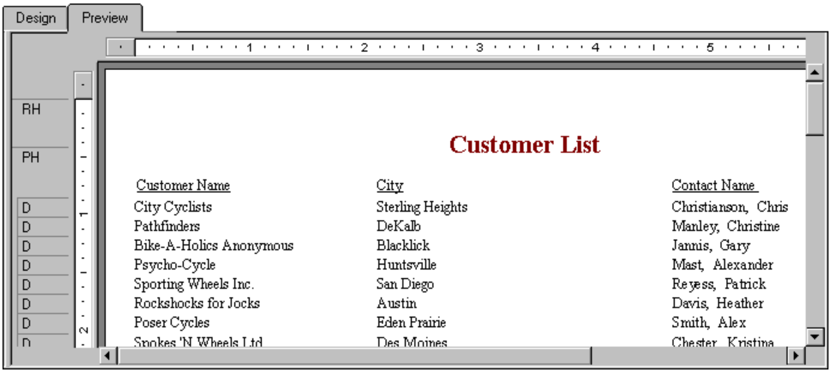
To delete a field

1 Select the **Country** field and the **Country** column heading using the Ctrl-click combination.

2 Press Delete on your keyboard.

That is all it takes to delete fields from the report.

The report should now look similar to this:



Customer List

Customer Name	City	Contact Name
City Cyclists	Sterling Heights	Christianson, Chris
Pathfinders	DeKalb	Manley, Christine
Bike-A-Holics Anonymous	Blacklick	Jannis, Gary
Psycho-Cycle	Huntsville	Mast, Alexander
Sporting Wheels Inc.	San Diego	Reyess, Patrick
Rockshocks for Jocks	Austin	Davis, Heather
Poser Cycles	Eden Prairie	Smith, Alex
Spokes 'N Wheels Ltd	Des Moines	Chester, Kristina

Balancing field spacing

Now that the Country field has been deleted, there is a large amount of white space between the City and Contact Name fields. You might be satisfied with the spacing as it stands, but it might be more readable if the columns were better balanced across the page.

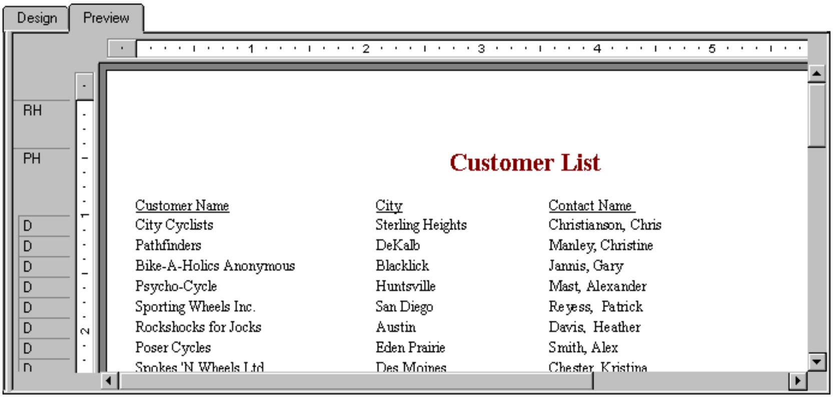
To balance field spacing

1 Return to the **Design** tab. Select the **Contact Name** field and its field heading by using the Ctrl-click combination.

2 Place the cursor over one of the two highlighted text objects and drag them to the left, closer to the **City** field.

3 Click the **Preview** tab and review your work again.

The report should look similar to this:



Customer List

Customer Name	City	Contact Name
City Cyclists	Sterling Heights	Christianson, Chris
Pathfinders	DeKalb	Manley, Christine
Bike-A-Holics Anonymous	Blacklick	Jannis, Gary
Psycho-Cycle	Huntsville	Mast, Alexander
Sporting Wheels Inc.	San Diego	Reyess, Patrick
Rockshocks for Jocks	Austin	Davis, Heather
Poser Cycles	Eden Prairie	Smith, Alex
Spokes 'N Wheels Ltd	Des Moines	Chester, Kristina

The spacing between the fields is much better, but it looks as if the report title is off-center.

4 Click the report title object to select it.

5 Position the cursor on the right handle of the object until the cursor turns into a Resizing cursor. Drag the right edge of the object frame until it is even with the right edge of the data in the **Contact Name** field object.

The report title automatically recenters itself based on the size of the object.

Grouping and sorting

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

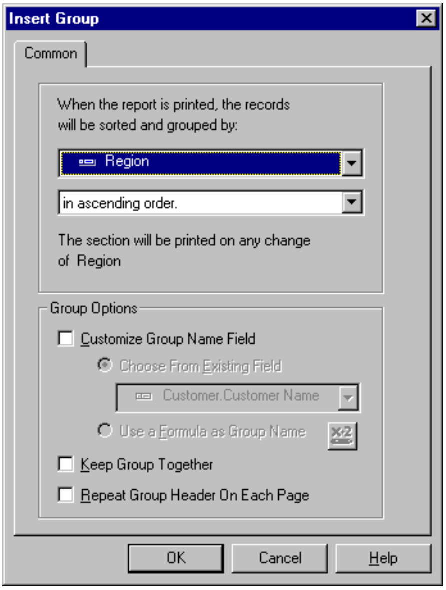
Grouping the report

In many reports you need to break the data into groups in order to make it easier to read and to understand. Crystal Reports lets you do this easily. For this customer list, you will group the customers by region and then sort the customers alphabetically within each group.

To group a report

1 While on the **Design** tab, click **Insert Group** on the Supplementary toolbar.

The Insert Group dialog box appears.



Insert Group

Common

When the report is printed, the records will be sorted and grouped by:

Region

in ascending order.

The section will be printed on any change of Region

Group Options

☐ Customize Group Name Field

● Choose From Existing Field

Customer.Customer Name

○ Use a Formula as Group Name

☐ Keep Group Together

☐ Repeat Group Header On Each Page

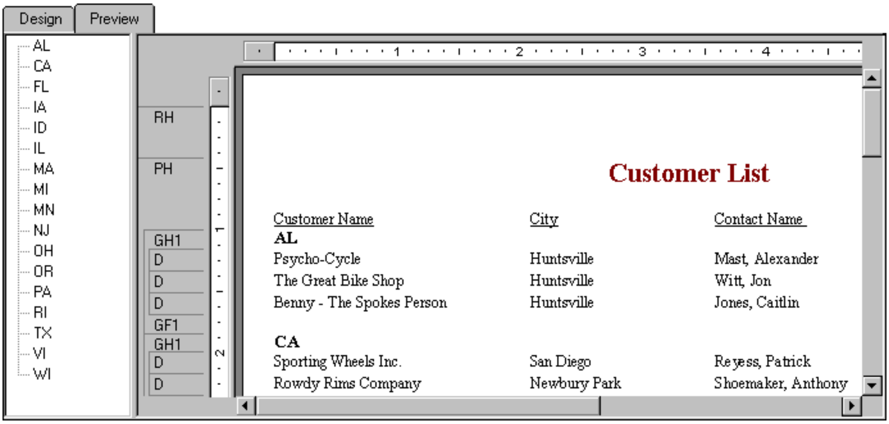
2 Select Region from the Customer table in the first drop-down list. The program takes all records with the same value in the region field and place them together in a group on the report.

3 Select **in ascending order** from the second drop-down list. The region grouping will be displayed on the report in alphabetic 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 the program shows that the report has been grouped.

5 Click the **Preview** tab to see what the report looks like.



Customer List

```
Customer Name              City             Contact Name
AL
Psycho-Cycle               Huntsville       Mast, Alexander
The Great Bike Shop        Huntsville       Witt, Jon
Benny - The Spokes Person  Huntsville       Jones, Caitlin

CA
Sporting Wheels Inc.       San Diego        Reyess, Patrick
Rowdy Rims Company         Newbury Park     Shoemaker, Anthony
```

6 If the group tree is not visible, select **Toggle Group Tree** on the Formatting toolbar to see the groups included in the report.

You can view the group of interest by clicking on the group name in the Group Tree. For example, to see the Texas customer group, click TX in the Group Tree. The program jumps to the Texas group, displaying that group in the Preview tab. The Group Tree allows you to quickly jump to a specific group of interest instead of scrolling through the report looking for the group. For more information on the group tree, see "Group Tree view" on page 86.

Note: For many reports, you will want to insert summaries, subtotals and grand totals. For example, when creating a sales report rather than a customer list, you would want to calculate the total sales amount for each region. See "Sorting, Grouping, and Totaling" on page 137.

Understanding “live” group headers

When a group is inserted, a group name field is automatically inserted in the Group Header section of the report. The group name field displays the current group’s name. For example, if you group by region, and preview the report, the group header for the CA (California) group shows “CA.”

The group field name is automatically formatted to stand out from the records in the group.

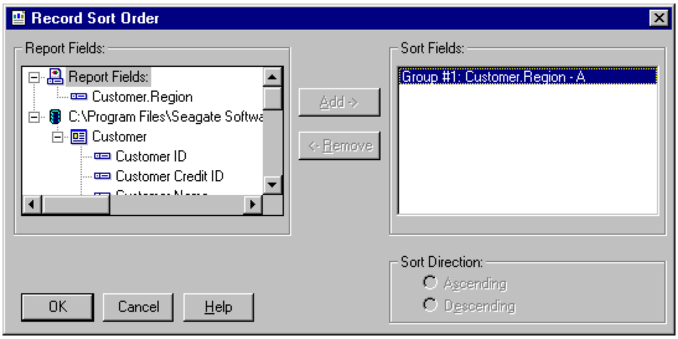
Sorting records

In a typical customer list report, customer names are listed alphabetically. In this example, you will sort the customer names alphabetically within each region.

To sort records

1 While on the **Preview** tab, click **Sort Order** on the Standard toolbar.

The Record Sort Order dialog box appears.

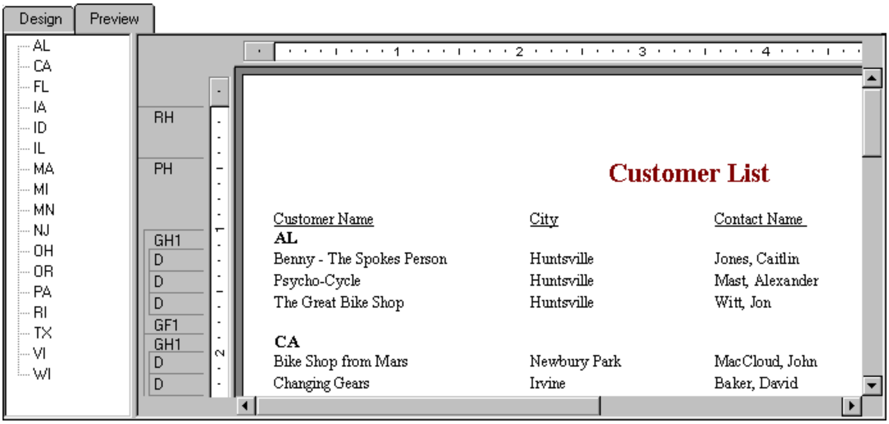


The Report Fields list box displays all fields currently on your report. You can choose to sort based on any of these fields.

The Sort Fields list box displays the fields that are already sorted in the report. Since the region field has already been sorted, the sorting you are about to do will be within each region, and not for the entire report.

2 Highlight the **Customer Name** field and click **Add**. Notice that the **Customer Name** field now appears in the **Sort Fields** list box.

3 Select **Ascending** for the Sort Direction and click **OK**. The report should now look similar to the following:



Customer List

Customer Name	City	Contact Name
AL		
Benny - The Spokes Person	Huntsville	Jones, Caitlin
Psycho-Cycle	Huntsville	Mast, Alexander
The Great Bike Shop	Huntsville	Witt, Jon
CA		
Bike Shop from Mars	Newbury Park	MacCloud, John
Changing Gears	Irvine	Baker, David

Notice that the records within each group are in alphabetic order.

Completing the report

You have just one step left to complete the report. A company logo needs to be added to the first page of the report.

Inserting a company logo

In this section, you will place a company logo at the top of the first page of the report.

To insert a company logo

1 While on the **Design** tab, click **Insert Picture** on the Supplementary toolbar.
The Open dialog box appears.

2 Choose the **Xtreme.bmp** and click **Open**. An object frame appears as you move the cursor over the report. The object frame represents the logo you will place.

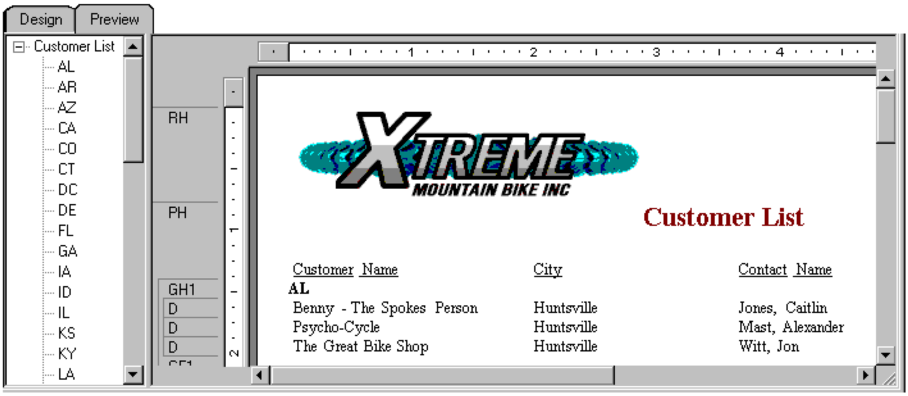
Tip: The Xtreme.bmp file can be found in the Databases subfolder of the Samples\En folder.

3 Position the object frame in the upper left-hand corner of the Report Header (RH) section of the report and click once to place it.
Placing the logo in the Report Header (RH) section ensures that the logo is printed only on the first page of the report.

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

4 Click the **Preview** tab to view the report.

The final report should look like this:



Customer List

Customer Name	City	Contact Name
AL		
Benny - The Spokes Person	Huntsville	Jones, Caitlin
Psycho-Cycle	Huntsville	Mast, Alexander
The Great Bike Shop	Huntsville	Witt, Jon

5 Save the report by clicking **Save** on the Standard toolbar.

You have just completed your first report. It is an impressive report, and Crystal Reports made it easy to do.

Related topics

For information about distributing reports (printing, exporting, etc.) see the printing topics in "Formatting" on page 199 and "Distributing and Viewing Reports" on page 337.

Quick start for advanced users

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.

If you're not an experienced user, check the "Quick start for new users" on page 39.

To choose a report type and data source

1 In Windows, click **Start**, point to **Programs**, then select Crystal Reports.

2 Click **As a Blank Report** and **OK** to open the Data Explorer dialog box.

As an alternative to creating a blank report, you can choose one of eight report experts. View style samples for each by selecting from the list of experts.

After choosing an expert, the Report Expert appears. You can build your report on a database, a query or a dictionary.

Choosing Crystal SQL Query or Crystal Dictionary allows you to search for the appropriate file. Choosing Database opens the Data Explorer dialog box.

3. Browse through the contents of the Data Explorer to find the database tables you want to use for your report.

   Tip: The More Data Sources option offers a variety of resource-specific data sources for you to choose from.

4. Select a table you want to use in your report and click **Add**.

5. After you have added all the tables you want to work with, click **Close** on the Data Explorer.

   The Design tab appears with Report Header, Page Header, Details, Page Footer, and Report Footer areas. A report is created by inserting and formatting items in each of these areas.

Note: To use additional database tables for a report and match them up on a record-by-record basis, click Link Expert on the Supplementary toolbar, select the table(s), and then set up the links in the Visual Linking Expert when it appears.

To work with report elements on the Design tab

1. Each of the default report areas contains a single section. To add additional sections, click **Section Expert** on the Standard toolbar and use the Section Expert to add the desired sections.

   Once you have added sections to an area, you can move, merge, and delete them in the Section Expert. See "Designing with guidelines" on page 211.

2. To turn the grid on or off, choose the **Options** command from the **File** menu and select or clear the Show Grid in Design check box in the Options dialog box when it appears. See "Using the grid" on page 209.

3. 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.

   Some further ways you can use guidelines:

   - Drag a field to a guideline until it snaps to the guideline.
   - Drag 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 "Designing with guidelines" on page 211.

4. If the Field Explorer dialog box is not visible, click **Insert Fields** on the Standard toolbar.

   The Field Explorer dialog box appears with "Database Fields" selected. Expand the active database table(s) to display a list of fields. To speed the entry of multiple fields, this box will remain on-screen until you click Close. This dialog box can be moved to a new location or resized, if you wish. See "Placing data on the report" on page 91.

5. Select the field(s) you want to appear on the report.

   You can select and place them one at a time, or 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.

   When you place multiple fields, they appear in the same order that they appear in the Field Explorer 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:

   - The field names and field types can be viewed by selecting the Show Field Names check box on in the Options dialog box (Layout tab).
   - The program automatically places field titles in the Page Header section unless the Insert Detail Field Titles check box is not selected in the Options dialog box (Layout tab).
   - If additional Details sections are added to the report, field titles will only be placed in the Page Header section for fields in the Details A (the original) section of your report.

6. Once the objects are in place, you may want to adjust the report sections. To do this, right-click the shaded area to the left of the section ruler and use the shortcut menu that appears:

   - To expand a section to accommodate an additional line, choose the Insert Line command.
   - To have the program automatically align the objects in the section horizontally, choose the Arrange Lines command.
   - To reduce the size of a section to eliminate unnecessary white space above and below objects, choose the Fit Section command. See "Using white space between rows" on page 235.

7. To create a report title, you must first enter the title in the Document Properties dialog box. On the **File** menu, click **Summary Info**. Enter a title in the **Title** text box of the Document Properties dialog box. Click **OK**.

8. Choose **Report Title** from the Special Fields list in the Field Explorer. A rectangular placement frame appears when you move the cursor over your report. Click once in the Report Header (RH) section to place the report title. The report title field contains the text that you typed in the **Title** text box of the Document Properties dialog box. See "Adding a title page to the report" on page 101.

9. To see how the results will print, click **Print Preview** on the Standard toolbar.

   To speed processing time while building a report, you can preview the report using only a small subset of the available data. To do this, go to the File menu, point to Print and then click Preview Sample. See "Preview tab" on page 85.

In either case, the program takes you to the Preview tab. You can fine-tune a report in the Preview tab while viewing the results as actual report data. You can also close the Preview tab and continue working on the report in the Design tab.

To use other reporting features

1 If you want to format a field, change the placement or width of a field, or insert a subtotal or grand total, click the field to select it. Handles appear on the top, bottom, and sides of each selected field:

- To change the placement of the field(s), use the mouse to drag the field placement frame to its new position.
- To change the width of the field, use the mouse to drag the right or left handle.
- Right-click the field to format font, alignment within field, number, currency, date display, border, color, indentation, or to summarize the field. A shortcut menu appears listing various commands for formatting and summarizing the field.

Tip: Many font and formatting options are available on the formatting toolbar.

Note: To apply formatting only under certain conditions, click Conditional Formula next to the formatting property in the Format Editor, and create a formula that defines those conditions. See "Working with conditional formatting" on page 237.

2 To create a formula that makes data calculations or comparisons, click **Insert Fields** on the Standard toolbar. When the Field Explorer dialog box appears, select **Formula Fields**.

3 Click **New**. The Formula Name dialog box appears. Enter a name for the formula and click **OK**. The Formula Editor appears.

4 Enter the formula in the Formula Editor. Enter fields, operators, and functions by selecting them from their respective lists or by typing them in. You can check the formula syntax by clicking **Check**.

5 When you are finished editing, click **Close** to return to the Field Explorer dialog box. Click **Insert** to place the formula just like you would a database field. See "Using Formulas" on page 381.

6 To insert a subreport (a report within a report), click **Insert Subreport** on the Supplementary toolbar and choose an existing report to import as a subreport or use the Create Report Expert to create a new subreport. See "Inserting subreports" on page 497.

If you want the records in a subreport to match up with the records in a primary report, click the Link tab of the Insert Subreport dialog box and specify the link in the Subreport Links dialog box when it appears. See "To link a subreport to the data in the primary report" on page 501.

7. To insert a cross-tab object in a report, click **Insert Cross-Tab** on the Supplementary toolbar and set up the cross-tab in the Format Cross-Tab dialog box when it appears. See "Cross-Tab Objects" on page 299.

8. To create a parameter field (a field that prompts you for a value whenever you retrieve data for a report), click **Insert Fields** on the Standard toolbar, then select Parameter Fields in the Field Explorer dialog box when it appears. Click **New** to set up a parameter field. Once created, you can insert the parameter field in a report like a database field or select it from the **Fields** list in the Formula Editor.

   Parameter fields can be used in reports (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 "Parameter Fields" on page 477.

9. To add a chart, click **Insert Chart** on the Standard toolbar. See "Charting" on page 247.

10. To add a map, click **Insert Map** on the Standard toolbar. See "Mapping" on page 265.

11. To insert a spreadsheet, picture, or other OLE object that you can edit from within the Report Designer using the tools from the object's native application, choose **OLE Object** from the Insert menu. See "OLE" on page 289, and "Working with static OLE objects" on page 294.

12. To change the record sort order, click **Sort Order** on the Standard toolbar. The Record Sort Order dialog box appears. Highlight the field(s) you want to use for sorting the report data and the sort direction. See "Sorting single and multiple fields" on page 139.

13. To limit the report to specific records (for example, the records of California customers who have year-to-date sales greater than $10,000), click the first field on which you want your selection to be based and then click **Select Expert** on the Standard toolbar. When the Select Expert appears, set up the record selection criteria.

14. To print the report, click **Print** on the Standard toolbar.

That's it! It is that easy to build a report.

Report Design Concepts 5

This chapter describes the basic concepts of report design and helps you decide what information you want to include in your report.

Basic report design

The purpose of this chapter is to suggest a structured approach to preparing a Crystal report. This approach includes the following elements:

- deciding on the content of the report
- developing a prototype on paper.

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

Deciding on the content of the report

Before you do anything else, you should outline the information you want the report to provide. The following sections provide a guide to making that outline.

Stating the purpose

What is the overall purpose of the report?

Reports are management tools. Their purpose is to help you quickly grasp the essential elements and relationships found in raw data, to help you make effective decisions. For a report to be effective, it has to present the correct 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 may 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 the 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 representatives, compare this year's numbers to last year’s, 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.

Defining the purpose of the report before you start is a critical step in the overall process.

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).

These individuals will be interested in different aspects of the report:

- a sales representative will use the report to evaluate individual sales performance and compare this performance to that of other representatives in the region
- the regional sales manager will use the report to evaluate regional representatives and compare the region's performance to that of other regions
- the national sales manager will use the report to evaluate the performance of regional managers and 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 and warehouse locations.

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

Determining the layout of the report

What is the report title going to be?

Write out a working title for the report. You may decide to change it later, but at least you will have a title to use when creating the prototype report.

What identifying information is needed in the header and footer?

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.

The information can come from a variety of sources, depending on the kind of information you plan to use.

- 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.
- Crystal Reports can generate information such as the current date or page numbers.

Finding the data

What data do you want to use in the report?

Do you know the type of database you are reporting from? Will you be reporting off a data file, SQL/ODBC, or a Dictionary?

If you do not know, ask the database administrator in your organization for help in setting up the database type and location of the data. See "Accessing Data Sources" on page 573.

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

If not, your MIS professional, database administrator, or co-workers will have to help you become familiar with the data.

What specific data should appear in the body of the report?

The body should contain all the data needed 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.

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

- Much of the data in a typical report is taken directly from data fields. Which data fields will be used, 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 placed directly into the report using text objects (headings, notes, labels, and so on).

Does the data exist or does it need to be calculated?

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 (for example, sales commission, 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 "Specifying formulas" on page 386.

What types of fields contain data?

You should take the time to get to know the data type for data fields that will be used in your calculations. Since formula functions and operators work with specific kinds of data, it is important to recognize the data type you are working with, before you start any calculations. For example, some functions require numeric data, while others work with only date or time fields.

Manipulating the data

Do you want the data organized into groups?

How? By customer? By date? By hierarchy? Or by other criteria? provides several options for grouping data in a report. See “Grouping data” on page 140.

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

Crystal Reports gives you both alternatives. See “Understanding sort options” on page 138.

Do you want the report to contain only specific records or groups?

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. can be used to select records based on simple date ranges or comparisons, or to create complex formulas to identify the records to be included. Take a few minutes to determine the records needed for the report and list the criteria to be used for selecting those records. See “Selecting records” on page 126.

Do you want to summarize the data?

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

allows you to do this, and it also allows the grand total (or the grand total average, grand total count, and so on) to be placed at the bottom of the selected column. See “Calculating a percentage” on page 164 and “Selecting top or bottom N groups” on page 158.

What information should be flagged on the report?

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

How do you want information 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.

gives you the opportunity to underline report elements, and change the font type, size, or color used for specific report items. It allows you to put borders around items and to draw lines and boxes (to break the report into sections), set off headings, and so on. All of these formatting tools can be used to highlight key data on a report. See “Formatting” on page 199.

Determining printing area 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.

In what order will the areas print on the report?

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. For example, 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 do report objects print?

The way objects print will determine how you design your report. This will help you decide where to place charts, Cross-Tabs, and formulas to get specific results.

Report Header

Objects placed in the Report Header area print once, at the beginning of the report.

- charts 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.

Page Header

Objects placed in the Page Header area print at the beginning of each new page.

- charts or Cross-Tabs cannot be placed in this section
- formulas placed in this area are evaluated once per page, at the beginning of each new page.

Group Header

Objects placed in the Group Header area print at the beginning of each new group.

- charts 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.

Details area

Objects placed in the Details area print with each new record.

- charts or Cross-Tabs cannot be placed in this area
- formulas placed in this area are evaluated once for each record.

Group Footer

Objects placed in the Group Footer area print at the end of each group.

- charts 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.

Report Footer

Objects placed in the Report Footer area print once at the end of the report.

- charts 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.

Page Footer

Objects placed in the Page Footer area print at the bottom of each page.

- charts and Cross-Tabs cannot be placed in this area
- formulas placed in this area are evaluated once per page, at the end of each new page.

Developing a prototype on paper

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

To design a paper prototype

1. Get the same size paper you will be using for the finished report.
2. Position the title and other descriptive header information, using boxes or lines to represent report elements.
3. Position the footer information.
4. Review the page layout for balance.
5. Look at the information you intend to include in the body of the report.
    - count the number of fields being used and estimate the appropriate spacing between fields
    - use rectangles to pencil in the fields within the 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.
6. Use small boxes to indicate group values and totals.
7. Place random flags in the column where you want flags to appear.
8. Darken any elements you want highlighted to make them stand out from the rest of the prototype.
9. Review the finished product for layout and balance, and make changes as needed.

Introduction to Reporting 6

This chapter shows you the basics of creating a report in Crystal Reports. First you'll learn about the report design environment, then about how to select database tables, place objects on a report, and how to sort, group, and total your report data.

Report creation options

Each time you create a new report, you have three options:

- use a Report Expert
- use another report as a template
- create a report from scratch.

You will probably use each option at some time.

Report Experts

The Report Experts help create reports as quickly as possible and many new users and developers alike prefer to create the majority of their reports using them. 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 reports step-by-step.

You can quickly create the report and see how it looks against the actual data. And best of all, if you then decide to make changes, you can get back to the Report Expert to further modify the report.

Another report

To build a new report based on one that already exists, another report can be used as a template. Open the report you want to use as a template by using the Open an Existing Report option and save it to a new file using Save As (found on the File menu). Use this option whenever you think templates can save time. Templates are useful to:

- Create a new report with a different grouping or different record selection than that of an existing report.
- Reconstruct a report based on an earlier time period using the same report structure used today.
- 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 without disturbing the underlying links. Then, without relinking, you can build all your new reports based on this report.

New report

The As a Blank Report option is used to create a report from scratch. This is useful when you want the full flexibility and control of building a report from the ground up, or when a report type is different from the many report types available in the Experts.

The As a Blank Report option was chosen for the "Quick start for new users" on page 39, because the process of creating a report from scratch most fully illustrates the basics of reporting.

Report data sources

Crystal Reports provides three different data types:

- Database (which includes ODBC and SQL data sources)
- Queries
- Dictionaries.

Database

In the Report Experts, choose the Database option to report on any of the standard (not client-server) PC databases, or any SQL or ODBC data sources.

Typically, the standard PC databases are those 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; the program has built-in capabilities to directly open database files and tables designed in Microsoft Access, dBASE, FoxPro, Clipper, Btrieve, and Paradox, among others. Once the program is installed on your system, you can immediately begin creating reports based on these databases by selecting the appropriate file.

See “Accessing Data Sources” on page 573.

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 use:

- an SQL server to create, store, and manipulate database files, tables, fields, and records
- an SQL client interface, allowing workstation users to retrieve data.

See “Using SQL and SQL databases” on page 535.

Open Database Connectivity (ODBC) is a standard developed by Microsoft through which many different types of data can be accessed by a single application. An application need only communicate with one set of ODBC files 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.

The Database option was chosen for the “Quick start for new users” on page 39, because the sample data, Xtreme.mdb, is a Microsoft Access database.

Queries

A query is 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), the 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 quite a while. Since the SQL Designer eliminates the need to understand SQL, it can get you building effective queries right away.

The SQL 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 SQL Designer helps create queries, even if you have no knowledge of SQL whatsoever. By answering a few questions on a set of sequential tabs, you provide the program with all the information needed 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 SQL 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 SQL Designer can be a powerful tool for many of your information gathering needs.

Dictionaries

In many large organizations, data is stored in a variety of places. It may be in different databases, on different servers, and so forth. IS departments often use naming schemes for tables and fields that are logical and predictable, but the names assigned may seem cryptic to non-technical staff. Enabling staff to create reports themselves in such an environment can create extensive training and support problems and possibly compromise data integrity. Dictionaries provide a solution to these problems. They allow you to provide your staff with ready access to the data in a form that they can understand, but they also allow you to maintain complete control and security over your data resources.

A dictionary is a structured, simplified, and secure view of organizational data that you can create for some or all of the users in your organization.

A dictionary is an optional metalayer that you can place between the user and the data. Using the capabilities of the metalayer you can:

- design a single, dynamic view of all the data that is necessary to create organizational reports and queries
- include 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
- limit access to specific columns of data (for example, letting only executives see the salary column of the employee data files)

- place restriction formulas on specific rows of data (for example, allowing managers to view employee information only for those employees with a salary under $40,000)
- 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 enable you to add an additional layer of security between the user and the data.

Note:

- When a dictionary is used to create a report the only data used in the report is data accessed through the dictionary. You cannot use a dictionary and some other data source in the same report. Because the dictionary is often used to impose data security, it would breach that security to allow unrestricted data access in a dictionary report.
- A subreport based on a different data source can be included in a primary report based on a dictionary.
- Dictionaries are an optional metalayer. Crystal Reports can be used without ever using dictionaries.

See “Dictionaries” on page 639.

Choosing data sources and database fields

Crystal Reports makes it simple to select data sources and database fields by providing easy-to-use functionality in the Data Explorer dialog box and the Field Explorer dialog box. Each of these dialog boxes uses the familiar Windows tree structure to allow you to navigate through the possible choices.

The Data Explorer

The Data Explorer provides an integrated tree view of all data sources you can use with Crystal Reports. With the Data Explorer, you can:

- create a report using one of the currently connected data sources
- create a report based on a customized “Favorites” list of data sources (which you also maintain within the Data Explorer)
- create a report based on a list of recently accessed data sources (that the Data Explorer automatically maintains for you)
- create a report by connecting to an existing data source (for example, a data file residing locally, or an ODBC data source that has already been set up)
- create a report by first creating a new connection (for example, a new ODBC data source, or an OLE DB File Data Link).

Tree View

The main window of the Data Explorer shows a tree view of possible data sources you can select when creating a report. The tree is made up of folders for:

- Current Connections
- Favorites
- History
- ODBC
- Database Files
- More Data Sources
- Metadata / Query.

For a brief description of each of these folders, see "Selecting the data source" on page 88.

Shortcut Menu

You can right-click any item in the Data Explorer to see a shortcut menu with the following options:

- **Add to Report**
  Use this option to add a table or stored procedure to your new report. This option is also available by clicking the Add button on the Data Explorer.
- **Add to Favorites**
  Use this option to add a selected data source to the Favorites folder. This option is also available by clicking the Add to Favorites button on the Data Explorer.
- **Remove from Report**
  Use this option to remove a table or stored procedure from your report. This option is also available by clicking the Remove button on the Data Explorer.
- **Properties**
  Use this option to obtain detailed information on the selected item.
- **Rename Favorite**
  Use this option to rename a data source in the Favorites folder.
- **Delete Favorite**
  Use this option to remove a data source from the Favorites folder.

Options

The Options button on the Data Explorer provides access to the options you can set globally on the Database tab of the Options dialog box.

The Field Explorer

Use the Field Explorer dialog box to insert, modify or delete fields on the Design and Preview tabs of Crystal Reports. You'll see the Field Explorer after you've selected a data source for a new report, or when you select Field Object from the Insert menu.

Tree View

The Field Explorer shows a tree view of database fields and special fields that you can add to your report. It also shows formula fields, SQL expression fields, parameter fields, running total fields, and group name fields that you have defined for use in your report.

Fields that have already been added to the report, or fields that have been used by other fields (such as formula fields, groups, running total fields, summaries, and so on) have a green check mark next to them.

Toolbar and Shortcut Menu

The toolbar provides buttons with tool tips and hot-key combinations. You can right-click any item in the tree view to bring up a shortcut menu.

The toolbar and the shortcut menu offer these functions:

- **Insert to Report**
  Use this option to add a field to the report. You can insert more than one field at a time by selecting multiple fields, right-clicking, and choosing Insert to Report.

  Alternatively, to insert a field, you can drag and drop it in the Design or Preview tabs.
- **Browse**
  Use this option to browse data for a database field or formula field.

  When you right-click a database field, the shortcut menu has Insert to Report and Browse Data enabled.
- **New**
  Use this option to create a formula field, SQL expression field, parameter field or running total field.
- **Edit**
  Use this option to modify an existing formula field, SQL expression field, parameter field or running total field.
- **Rename**
  Use this option to modify the name of an existing formula field, SQL expression field, parameter field or running total field.
- **Delete**
  Use this option to remove a formula field, SQL expression field, parameter field or running total field. You can also select multiple fields, right-click, and choose Delete to remove them all at once.

- **Move Up (or Down)**
  Use this option to change the order of parameter fields.

  When you right-click a parameter field, all the functions except Browse Data are enabled.
- **Show Field Type**
  Use this option to see the field type (string, number, and so on) when you're looking at a list of database fields.

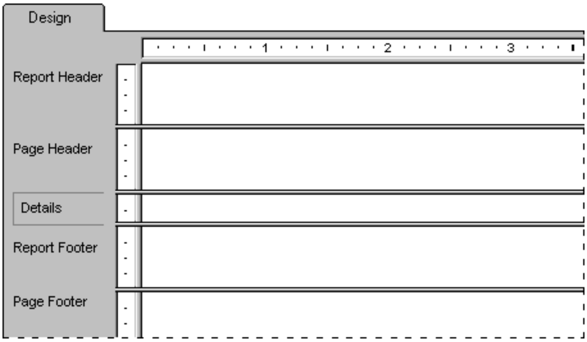
Group Name Fields

You can insert an existing Group Name field shown in the Field Explorer by right-clicking it and selecting Insert. Unlike a formula field, parameter field or running total field, however, you cannot create a Group Name field through the Field Explorer. (A Group Name field is created when you insert a group.)

About the report design environment

Design tab

When working with Crystal Reports, you will probably use the Design tab more than any other part of the program.



The Design tab is the place you do most of the initial work when creating a report. It designates and labels the various sections of the report. You can do the initial formatting, place objects in the sections where you want them to appear, specify sorting, grouping, and totaling needs, and so forth.

The Design tab provides a very efficient environment for designing a report because you work in the tab with data representations, not with data itself. When a field is placed 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 or network resources needed to gather the data.

The report created in the Design tab is a kind of virtual report; it has the structure and 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

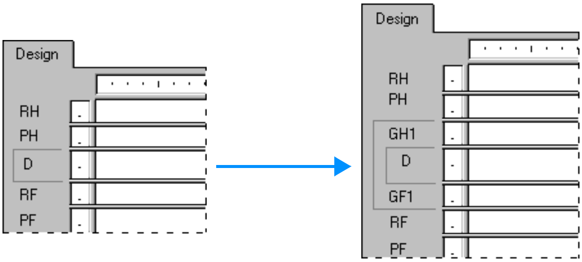
When you first begin creating a report, Crystal Reports automatically creates five areas in the Design tab.

- **Report Header**
  This section is generally used for the report title and other information you want to appear at the beginning of the report. It can also be used for charts 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, the name of the document, and other similar information. This section can also be used to display field titles above the fields on a report.
- **Details**
  This section is used for the body of the report, and is printed once per record. The bulk of the report data generally appears in this section.
- **Report Footer**
  This section is used for information you want to appear only once at the end of the report (such as grand totals) and for charts 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.

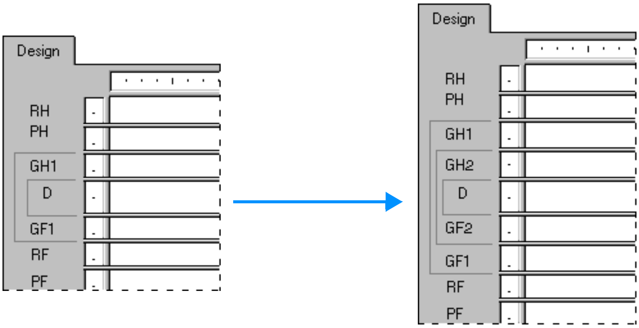
If a group, summary, or subtotal is added to the report, the program creates two additional sections:

- **Group Header**
  This section typically holds the group name field, and can be used to display charts or cross-tabs that include data specific to the group. It is printed once at the beginning of a group.
- **Group Footer**
  This section generally holds the summary value, if any, and can be used to display charts or cross-tabs. It is printed once at the end of a group.

When a group, summary, or subtotal is added, the Group Header area appears directly above the Details area and the Group Footer area appears directly below the Details area.



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 newly added areas can contain one or more sections. By default, they each contain a single section.

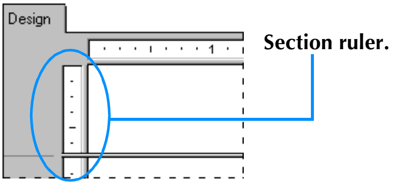
Identifying and working with areas and sections

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 selected the Show Short Section Names in Design check box on in the Options dialog box.
- 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. The section ruler is used to add, remove, and move guidelines, and to provide a visual reference when you are placing objects. See "Designing with guidelines" on page 211.



Section ruler.

Whenever a new section is added, the program creates a ruler for that section. See "Using multiple sections in reports" on page 187.

Other Design tab capabilities

There are several other capabilities built into the Design tab. With the Design tab, you can:

- Resize a section by dragging its boundary. See "Resizing a section" on page 186.
- Split a section (create two sections from one) by clicking its left boundary. See "Splitting a section" on page 185.
- Add horizontal and vertical guidelines by clicking the rulers. See "Designing with guidelines" on page 211.
- Zoom in and out on a report at any magnification from 25% to 400% of the original size. See "Using the zoom feature" on page 100.

Preview tab

To preview a report before printing it, click Print Preview on the Standard toolbar.

The program gathers the data, makes the necessary calculations, and displays the report in the Preview tab. 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.

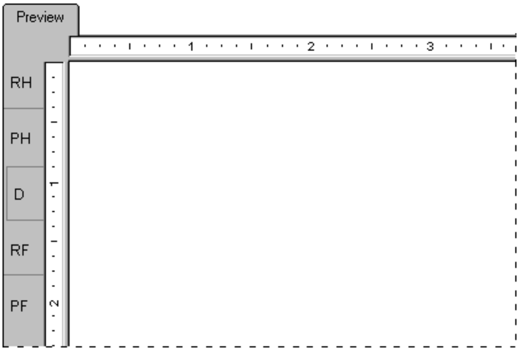
The program works with data in the following manner:

- the first time the Preview tab is used, 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.

Crystal Reports provides two views for previewing a report:

- Standard view
- Group Tree view.

Standard view

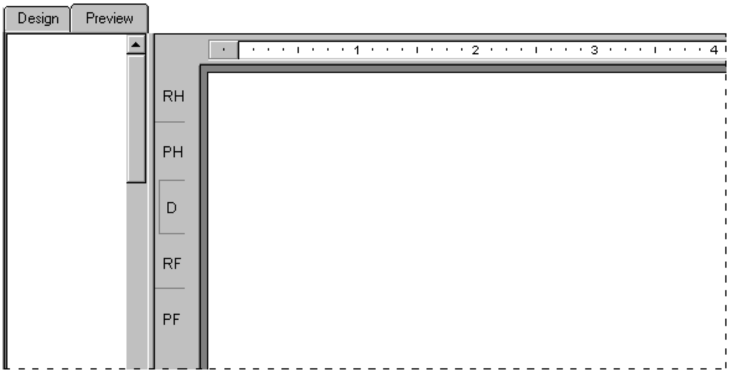


In standard view, the report is displayed a page at a time. Using the navigation buttons in the Preview tab, you can move to the beginning or end of the report, or you can move backward and forward through the report one page at a time. For shorter reports or reports in which you're primarily interested in seeing the "bottom line" totals, the standard view provides all of the functionality you need.

The Data Age indicator

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.

Group Tree view



The Group Tree view can be shown or hidden using Toggle Group Tree on the Formatting toolbar.

The Group Tree view presents a split screen:

- the right pane displays the report
- the left pane displays a high level outline of the report, showing the hierarchy of groups and subgroups in a familiar tree format.

The Group Tree normally displays the names of the groups and subgroups you created in your report. You can, however, customize these names using the Group Options section of the Insert Group or Change Group Options dialog box. For more information on customizing Group Names in the Group Tree view, see "Grouping data" on page 140.

When you click the tree node for the group that interests you, the program jumps immediately to the part of the report that contains the information for that group. For longer reports or reports in which you wish to jump back and forth between different groups, the Smart Navigation features of the Group Tree view make your work extremely efficient.

Comparisons with the Design tab

You have the same formatting capabilities in the Preview tab as you do in the Design tab. Menus (both menu bar and shortcut menus), the Standard and Supplementary toolbars, and the Formatting toolbar remain active, providing essentially the same functionality you have when working with a report in the Design tab. However, when you are making numerous changes, it is quicker to make the changes in the Design tab. Some additional things to consider are:

- The Design tab and Preview tab are tied together internally. Any changes made in one are reflected in the other.
- The Preview tab has a single vertical ruler at the left of the tab rather than the individual section rulers seen in 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 appear only once in the Design tab, they print each time a section prints in the Preview tab.
- The Record counter, the Data Age indicator (see "The Data Age indicator" on page 86), and the Page Forward/Page Back controls (see "Preview tab" on page 85), are all active in the Preview tab.
- The Preview tab highlights every value when you select a field; whereas, only the field frame is highlighted in the Design tab.

Working in the Preview tab has a different feel from 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 highlight the field, sizing handles appear on the frame and the frame changes color.

In the Preview tab, however, you are working with the actual data. Instead of a field frame representing many field values, the values themselves appear. Some additional things to consider are:

- When you highlight 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 in the field.
- Likewise, when you select a summary value, you are actually selecting all 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 differences in appearance, 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.

Creating a new report

Selecting the data source

After deciding which option you want to use for creating your report (see "Report creation options" on page 76), the next step is to select a data source to use.

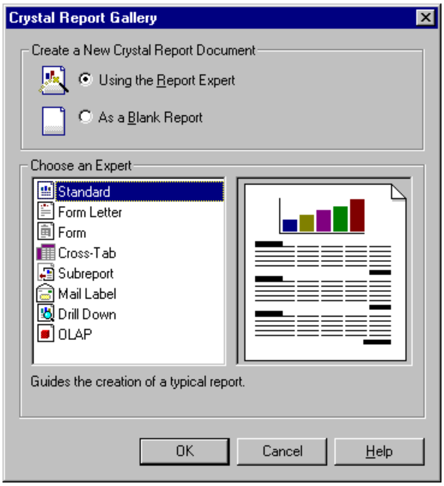
Most data sources can be chosen through the Data Explorer dialog box. The Data Explorer appears when you choose Database in a Report Expert, or when you create a report from scratch using As a Blank Report.

Note: This procedure uses the Report Experts to show you how to select a data source. The Data Explorer dialog box is used the same way for reports created from scratch.

To select the data and start a new report

1. On the Standard toolbar, click **New**.

   The Crystal Report Gallery appears.



Create a New Crystal Report Document

● Using the Report Expert

○ As a Blank Report

Choose an Expert

- Standard
- Form Letter
- Form
- Cross-Tab
- Subreport
- Mail Label
- Drill Down
- OLAP

Guides the creation of a typical report.

2 Double-click the Expert for the type of report you want to create.

The first tab in each Expert is a Data tab. Use the options on this tab to select your data source through the Data Explorer dialog box.

3 On the **Data** tab, click **Database**.

4 Use the tree view of the Data Explorer to select your data source:

- **Current Connections** shows a list of data sources you are currently connected to.
- **Favorites** shows a list of data sources you commonly use and have maintained in your Favorites list.
- **History** shows a list of data sources you have used recently. The last five data sources used are displayed.
- **ODBC** shows a list of ODBC data sources you have already configured for use. You can use Create New Data Source to configure a new ODBC data source.
- **Database Files** shows a list of standard PC databases that reside locally. You can use Find Database File to browse for a PC database using the Open dialog box.
- **More Data Sources** shows a list of other data sources that can be accessed through OLE DB and native drivers.
- **Metadata / Query** lets you browse for a dictionary (.dct or .dc5), or query (.qry) file.

Related topics

"Accessing Data Sources" on page 573.

"Selecting a query for a report" on page 635.

"Using a dictionary" on page 641.

Linking multiple tables

If the report contains data from two or more database tables, they need to be linked at this point when creating reports. Database tables are linked so records from one database match related records from another. For example, if you activate a Suppliers table and a Product table, the databases are linked so that each product (from the Product table) can be matched up with the supplier (from the Supplier table) that made the product.

The majority of reports will probably require data from two or more tables, so linking will be necessary. The process of linking is made easy with the Visual Linking Expert. See "Linking tables" on page 515.

Note: It isn't necessary to link tables in reports created from a query because any links required by the data have already been processed.

To add and link multiple tables

1 Choose **Add Database to Report** from the **Database** menu.

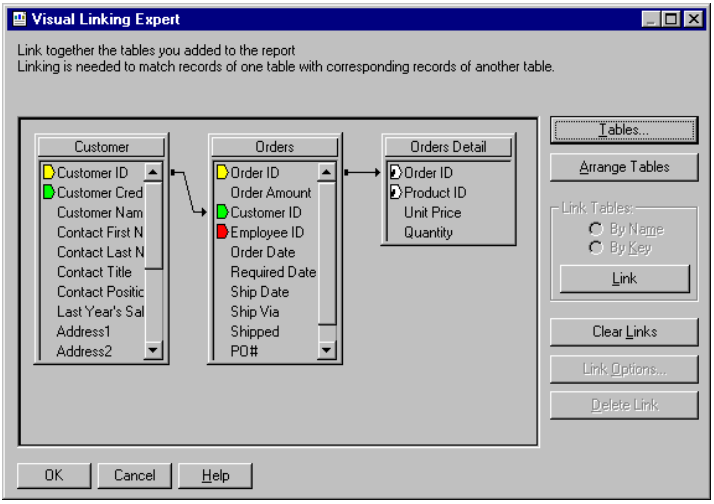
The Data Explorer dialog box appears.

2 Select the new database you want to use in the report. See "Selecting the data source" on page 88.

3 Click **Close** when finished with the Data Explorer.

The Visual Linking Expert appears and displays the databases currently available for linking.

Note: If Auto-Smart Linking is selected on in the Database tab of the Options dialog box, you will not need to manually create links between the tables. See "Linking tables" on page 515.



Link together the tables you added to the report
Linking is needed to match records of one table with corresponding records of another table.

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

Note: You can link tables by table name or by foreign key information.

5 To link databases automatically, click **Link**.

6 To specify the tables, click **Tables**.

7 Click **OK** when finished.

Note: When manually creating links, the field you are linking "to" must be an indexed field. For more information on indexed fields and tables, see "Indexed tables" on page 514.

The Visual Linking Expert closes, and you are returned to your report. The linked databases are now available for use in your report. If you are not satisfied with the link, you can modify it using the Visual Linking Expert.

Related topics

"Understanding Databases" on page 509.

"SQL join types (ODBC data sources)" on page 527.

Placing data on the report

Placing data on a report is a very important task. You need to know what type of data should be placed on the report and where on the report it should be placed.

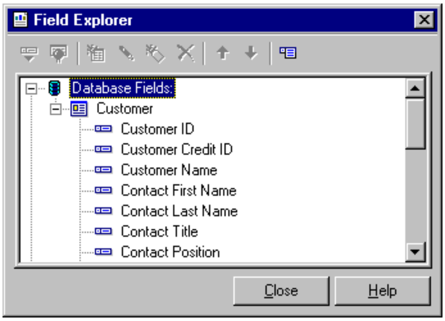
Database fields

Much of the data placed on a report are database fields, displaying data as it is stored in the database. For example, in the "Quick start for new users" on page 39, the Customer Name, City and Country fields are placed on the report. Normally, database fields will be placed in the Detail section, but under certain circumstances, they will be placed in other sections of the report.

To insert a database field

1 On the Standard toolbar, click **Insert Fields**.

The Field Explorer dialog box appears with "Database Fields" selected. All the fields in the chosen database(s) are listed. To speed the report building process, this dialog box remains on-screen until you click Close. You can move this dialog box wherever you wish.



2 Select the field you want to appear in the report.

3 Click **Browse** to review the values in the selected field.

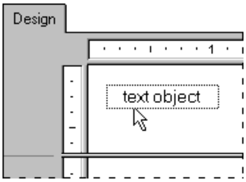
4 Click **Insert to Report** to place it in the report.

Text objects

Text objects are used in reports for a multitude of purposes. They are a powerful way of inserting titles; labeling summaries and other data on your report; and of easily combining database fields. For example, in the "Quick start for new users" on page 39, text objects are used to easily display the two contact name database fields as one object, to insert a column heading for the concatenated contact name, and to insert a title in your report.

To insert a text object

1 On the Standard toolbar, click **Insert Text Object**. An empty object frame appears.



2 Position the text object where you want it to appear in the report.

Click once on the border of the text object to select it for resizing and moving.

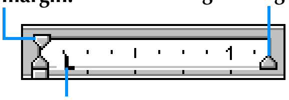
Double-click inside the text object to select it for editing. The Design tab ruler changes to a text object ruler sized to the length of the selected object. To the left, a tab indicator appears. By clicking the tab indicator, you can cycle through the four tab options available.

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 the position on the ruler where you want to insert it.

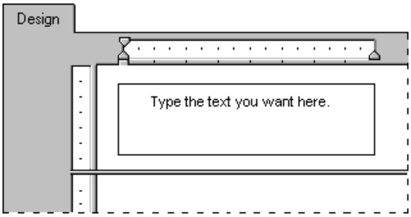
The ruler allows you to add indents and align text within the text object.

Note: When you first insert the text object into the report, the object is automatically selected for editing.

To insert a database field in a text object

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

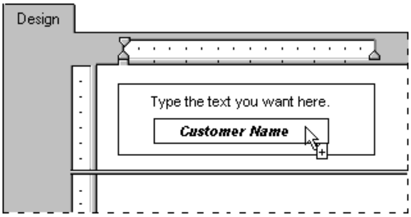
1 Insert a text object in the report.



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

3 On the Standard toolbar, click **Insert Fields**.
The Field Explorer dialog box appears with “Database Fields” selected.

4 Highlight the database field to be inserted and drag it into the text object. As you move the Drag and Drop cursor over the text object, the program displays a movable insertion point.



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

Special fields

To display information such as Page Numbers, Print Date, and Report Comments use the commands in the Special Fields tree view of the Field Explorer dialog box.

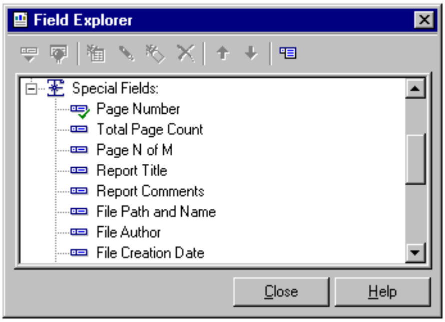
To insert a special field

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

1 On the Standard toolbar, click **Insert Fields.**

The Field Explorer dialog box appears with "Database Fields" selected.

2 Scroll down to **Special Fields** and expand it by clicking.



3 Choose a command from the Special Field list to insert in the report.

Each special field is inserted into the report as an object. An object frame appears. You can now place it on the report.

Note: To change the formatting of an inserted object, click the object to select it and click Format on the Supplementary toolbar. The Format Editor appears where you can make the desired changes. See "Formatting" on page 199.

Formula fields

To display data that is a calculated value, you need to create a formula field and place that formula field on the report. For example, if the database only stores the order dates and ship dates for orders but you need to display the number of days it takes to ship the order, you must create a formula field that will calculate the number of days between ordering and shipping. This is just one example of the use of formula fields. See "Using Formulas" on page 381, for an introduction to formulas.

Running total fields

To display a total that evaluates each record and provides a running sum of all the values in a field (or all the values in a certain set of values), a running total field needs to be created and placed in the report. If the first three values in a field were 2, 4, and

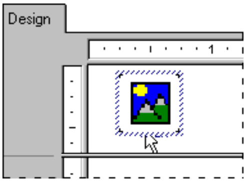
6, a running total would print 2, and then 6 (the sum of 2 + 4), and then 12 (the sum of 2 + 4 + 6). See "Running Totals" on page 171, for an introduction to running totals.

Picture fields

When designing reports, there will be times when you want to include a picture. For example, you may wish to put a company logo in the report header.

To insert a picture

1. On the Supplementary toolbar, click **Insert Picture**.
   The Open dialog box appears.
2. Select the desired picture file from the file list and click **Open** to return to the report.
   An object frame appears with the picture inside, ready to be positioned.



3. Position the picture object where you want it to appear in the report and right-click once.

SQL Expression fields

SQL expressions are like formulas, but they are written in Structured Query Language (SQL), not in the Crystal Reports formula language. An SQL expression can be used to query the database for specific sets of data. You can sort, group, and select based on SQL expression fields. See "The Crystal SQL Designer" on page 621, for an introduction to SQL expressions.

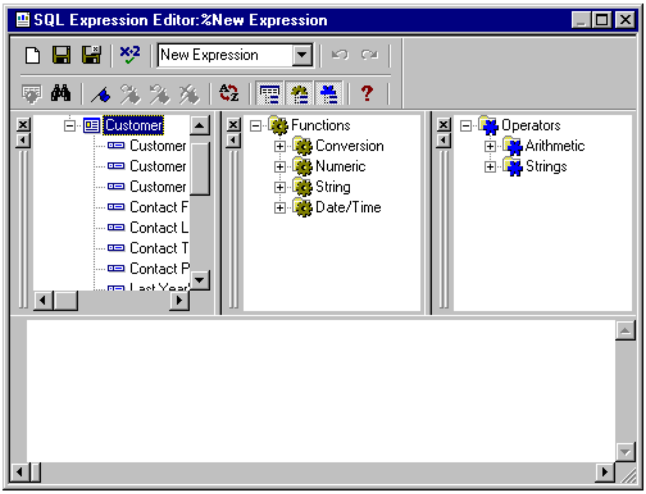
When an SQL/ODBC data source is used to design a report, the program adds SQL Expressions Fields to the Field Explorer dialog box. This new option allows SQL expressions to be entered in the Field Explorer and then inserted into a report quickly and efficiently.

To create an SQL Expression field

1. On the Standard toolbar, click **Insert Fields.**
   The Field Explorer dialog box appears with "Database Fields" selected.
2. Scroll down to **SQL Expression Fields** and highlight it. Click **New**.
   The SQL Expression Name dialog box appears.

3 Enter a name in the **Name** box, and then click **OK**.

The SQL Expression Editor appears.



Note: The SQL Expression Editor functions in the same way as the Formula Editor. For a discussion of these, tools see "Formula components and syntax" on page 385.

4 Type the expression in the SQL Expression Editor.

5 Click **OK**.

Hyperlink fields

You can select an object on the Design or Preview tab and create a hyperlink to a web site, a local HTML file, a field value, an e-mail address, or another Crystal Report file.

The hyperlink is saved with your report and is available to other users as a way of viewing additional information.

Note: Crystal Reports also lets you create hyperlinks with Relative URLs, so your web reports retain independence from their location on any particular server. To use a Relative URL, first set the Hyperlink type to "A file"; then delete the default prompt—file:///—from the File Name field, and replace it with your Relative URL.

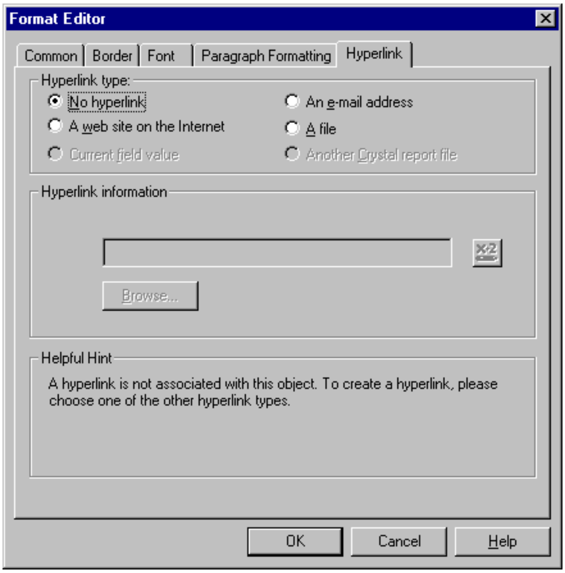
To create a hyperlink field

1 Select a report object on the **Design** or **Preview** tab.

Note: If you want to create a hyperlink to another Crystal Report file, do not select a report object.

2 On the **Insert** menu, click **Hyperlink**.

The Hyperlink tab of the Format Editor dialog box appears.



Hyperlink type:

● No hyperlink
○ A web site on the Internet
○ Current field value
○ An e-mail address
○ A file
○ Another Crystal report file

Helpful Hint

A hyperlink is not associated with this object. To create a hyperlink, please choose one of the other hyperlink types.

3 Click the type of hyperlink you want to create. The available types are:

- No hyperlink
- A web site on the Internet
- Current field value
- An e-mail address
- A file
- Another Crystal report file.

Note: Not all hyperlink types are available at all times. The object you select and its location on the report determine which types are available.

4 After you have chosen a hyperlink type, enter the appropriate hyperlink information (the URL of a web site, for example).

5 Click **OK** when you are finished.

The hyperlink is inserted as appropriate. Click it on the report to go to the appropriate site, e-mail address or other report.

Note: If you set the hyperlink type to “Another Crystal Report file”, you are creating an on-demand subreport (see “Subreports” on page 493). Just as one subreport cannot contain another subreport, you cannot create a hyperlink between two subreports.

BLOB (Binary Large Object) fields

A BLOB field is a database field whose data consists of Binary Large Objects—such as bitmap graphics, images, OLE objects, metafiles, and so on. Inserting a BLOB field into your report allows you to access these binary objects as you would other data types.

In general, Crystal Reports allows you to access BLOB fields containing:

- device-independent bitmaps (DIB)
- JPEG, TIFF, or PNG images.

In addition, if your data resides in a Microsoft Access database, then Crystal Reports enables you to report on BLOB fields containing OLE 1 and 2 objects and metafiles.

You insert BLOB fields as you would any other database field. For details, see "To insert a database field" on page 91.

A BLOB field object differs from other database field objects in that it offers options to control cropping, scaling, and sizing—just like an inserted picture or OLE object. To access these options, right-click the BLOB field object, select Format Graphic from the shortcut menu, and click the Picture tab.

Note: To retain the ability to activate and edit an OLE object with its server application, you should insert the object into your report independently, either by linking or by embedding, rather than inserting it in a BLOB field. For more information about inserting linked and embedded OLE objects, see "Inserting OLE objects into reports" on page 292.

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 "Quick start for new users" on page 39, you format the title, add a text object to identify the Contact Name information, and insert the company logo. See "Formatting" on page 199.

Record selection

Record selection, the task of paring down the data 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 "Quick start for new users" on page 39, has information from both United States and international customers. Record selection is used to create a report listing only customers in the United States. See "Record Selection" on page 125 and "Grouping data" on page 140.

Grouping, sorting, and summarizing data

Once a basic report is created, you will want to organize the data by grouping related information, sorting individual records, summarizing, subtotaling, and grand totaling.

Grouping records

To organize the data, you may want to group related data together. For example, in the "Quick start for new users" on page 39, after grouping the Customer List by region, you will divide the list into regional groups. That way, a sales manager for the California region could quickly locate the California group and focus exclusively on the customers within that region. See "Grouping data" on page 140.

Sorting records

Crystal Reports allows you to specify the order in which you want the records on your report displayed. For example, after grouping in the "Quick start for new users" on page 39, you sort the records within each region in alphabetic 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 "Sorting single and multiple fields" on page 139, and "Sorting records within groups" on page 144.

Summaries, subtotals, and grand totals

Many reports 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, allowing 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).

Using the drill-down option on summarized data

You can drill down on your data, to show the data behind individual groups, using the Drill-down cursor. See "Sorting, Grouping, and Totaling" on page 137.

Drill-down cursor

Crystal Reports allows you to drill down on group or summary information in the Preview tab in both the Standard and the Group Tree view (See "Standard view" on page 86 and "Group Tree view" on page 86). When you position the cursor over any summary value that you can drill down on, the program displays a Drill-down cursor.

Tip: Group headers appear on the Drill-down tab just as they do in the main report itself.

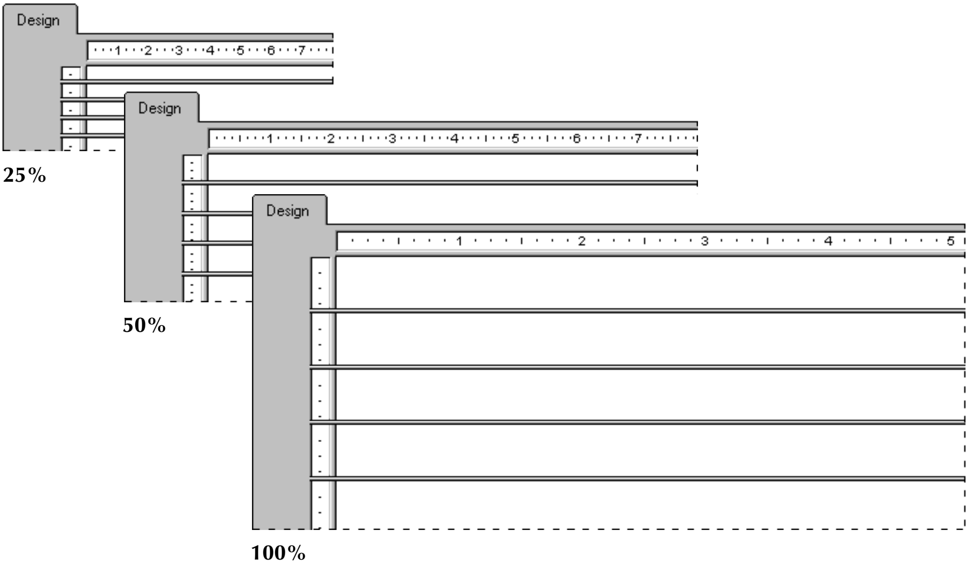
If you then double-click, the program reveals the details behind that specific summary value. For example, if the Drill-down cursor becomes active over the city summary, you can double-click to see the details behind that summary.

- 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.

Using the zoom feature

You can easily zoom in on a report. You can choose any magnification from 25% to 400%. This feature is active in both the Design and Preview tabs.

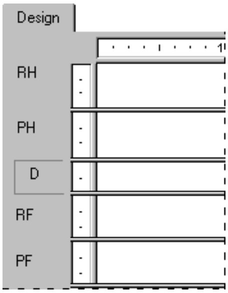
To zoom the report in or out, set the zoom level in the Zoom Control box found on the Standard toolbar.



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 focus attention on the details of the report.

Inserting page headers and footers

You can use page headers and footers by placing the information in the Page Header or Page Footer sections of the Design tab.



- information to appear only on the first page of the report goes in the Report Header (RH)
- information to appear only on the last page of the report goes in the Report Footer (RF)
- information to appear at the top of every page goes in the Page Header (PH)
- information to appear at the bottom of every page goes in the Page Footer (PF).

Text, fields, or formulas can be used in these sections just as in the Details section.

Related topics

"Creating footers after the first page" on page 241.

Adding a title page to the report

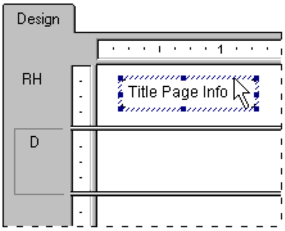
Crystal Reports provides a quick, easy way to add a title page to a report by selecting Report Title from the Special Fields in the Field Explorer dialog box. In order to use this field, you must have a title entered in the Summary tab of the Document Properties dialog box. See "Adding summary information to the report" on page 103.

To add a report title

1. On the Standard toolbar, click **Insert Fields.**

   The Field Explorer dialog box appears with "Database Fields" selected.
2. Scroll down to **Special Fields** and expand it by clicking.

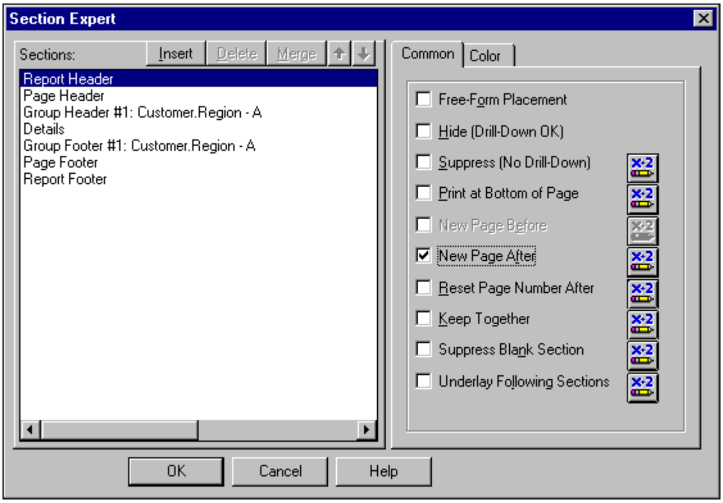
3 Select **Report Title**, and click **Insert to Report**. An object frame appears when the cursor is moved over the report.



4 Move the object frame to the Report Header section and click once to place the frame.

5 With the report title selected, click **Section Expert** on the Standard toolbar.

The Section Expert appears.


Section Expert

Sections: Insert Delete Merge

Report Header
Page Header
Group Header #1: Customer.Region - A
Details
Group Footer #1: Customer.Region - A
Page Footer
Report Footer

Common | Color

□ Free-Form Placement
□ Hide (Drill-Down OK)
□ Suppress (No Drill-Down)
□ Print at Bottom of Page
□ New Page Before
☑ New Page After
□ Reset Page Number After
□ Keep Together
□ Suppress Blank Section
□ Underlay Following Sections

OK Cancel Help

6 With the Report Header section highlighted, select the **New Page After** check box.

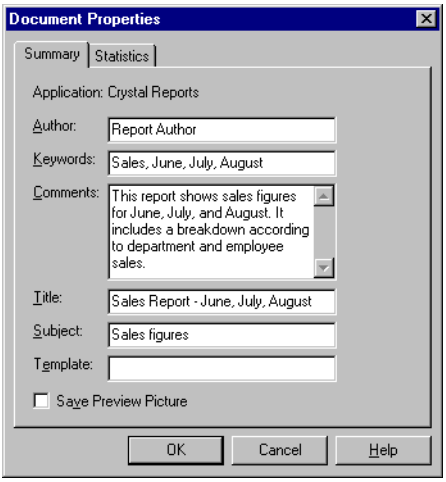
Now the title will appear on the first page and the report will begin on the second page.

Adding summary information to the report

There may be times when you want to include non-printing comments with a 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.).

Summary Info 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 Summary Info, the Document Properties dialog box appears with the Summary tab active.



Application: Crystal Reports

Author: Report Author

Keywords: Sales, June, July, August

Comments: This report shows sales figures for June, July, and August. It includes a breakdown according to department and employee sales.

Title: Sales Report - June, July, August

Subject: Sales figures

Template:

□ Save Preview Picture

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

Beyond basic reports

Once you are comfortable with the basics of reporting, you will be ready to investigate the more powerful reporting features of Crystal Reports, including:

- charts (see "Charting" on page 247)
- OLE objects (see "OLE" on page 289)
- form letters (see "Form letters" on page 191)
- subreports (see "Subreports" on page 493)
- cross-tabs (see "Cross-Tab Objects" on page 299)
- multi-section reports (see "Multiple Section Reports" on page 181).

Designing Optimized Web Reports 7

This chapter suggests ways to enhance reports so they take advantage of performance enhancements made to Crystal Reports. While the suggestions made here are *especially* important for optimizing the performance of reports distributed over the thin-wire environment of the Web, the majority of the guidelines and procedures are applicable to all of your reports.

Overview

Whether you distribute your enterprise reports over a local network, a corporate Intranet, or the Web, you can use Crystal Reports' powerful, built-in performance features to deliver web reporting speed.

Crystal Reports automatically provides these major performance advantages—even if you don't apply any of the strategies set out in this chapter:

- **Page-on-demand technology**

  Page-on-demand report access lets users download only the specific report pages they need to see, thus improving response times and reducing web traffic. Further, placeholders and partial page technology allow you to view report pages and data over the Web immediately, without having to wait for the processing of large objects, such as graphics and subreports.
- **Optimized, multithreaded Report Engine**

  The Crystal Report Engine's enhanced multithreading capabilities and thread-safe database drivers allow you to continue working on your important tasks, while many other operations are processed simultaneously in the background. The Report Engine also minimizes the number of passes made through the data, speeds up processing with improved memory management, and handles subreports and parameters as efficiently as possible.

In addition to these built-in features, the "Key strategies for optimizing web reports" discussed in the following sections bring additional performance benefits, which are often substantial. When you design new reports (or improve your old ones) in accordance with these strategies, reports run faster and tie up fewer processing resources. Consequently, report users can easily access the data they need—faster than ever.

Tip: If you're new to web reporting, or to reporting in general, reading this chapter will prepare you for future reporting tasks—you'll gain important insight into designing faster, better reports.

Note: Understanding databases and how they work is often important when considering performance. See "Databases overview" on page 510 for background information.

Key strategies for optimizing web reports

- As your company grows, Crystal Reports and Crystal Enterprise can grow along with you. For details, see "Scale with Crystal Enterprise Reports" on page 107.
- For tips and considerations on how to create fast, interactive reports, see "Making the right design choices" on page 108.
- For information about getting the most out of your existing database, see "Streamlining your reporting environment" on page 112.

- To minimize data transfer and enhance report performance, see "Using enhanced record selection formulas" on page 115.
- To decrease the processing and data transfer times of grouped, sorted, or totalled reports, see "Improving grouping, sorting, and totaling" on page 121.

In general, report performance issues are similar, no matter how you distribute your reports. By following these strategies, you'll notice significant improvements not only in multi-user web environments, but also in single-user situations.

Scale with Crystal Enterprise Reports

Crystal Enterprise extends the value and scalability of Crystal Reports. It provides eBusiness with a scalable, web-based solution for managing the access and delivery of hundreds or thousands of Crystal reports to every decision-maker—across the enterprise and beyond.

Crystal Enterprise provides a flexible, managed reporting solution. It gives you the ability to deliver the report-design power of Crystal Reports, atop a scalable, web-based infrastructure for managing a central report repository, user security, and report scheduling and processing.

If you purchased the Professional or Developer edition of Crystal Reports, then you also received Crystal Enterprise Standard on CD. This edition of Crystal Enterprise runs on a single machine, and provides anonymous user access to reports over the Web. With the CD, you receive five free concurrent access licences.

As your enterprise reporting needs grow, you can easily scale your system to Crystal Enterprise Professional, which can be installed on as many machines as are necessary. Further, with Crystal Enterprise Professional, you can specify which server pieces you want running on each machine.

Installing, configuring, and upgrading

To install and configure Crystal Enterprise Standard, consult the documentation included with the CD. The Crystal Enterprise documentation also describes how to add additional concurrent access licenses to your system, and how to scale with Crystal Enterprise Professional.

The Crystal Enterprise Standard CD (shipped with Crystal Reports 8.5) includes a temporary product license keycode that lets users install and operate the Crystal Enterprise system for a period of 30 days. To obtain a free, non-expiring product activation license keycode, users must register their copy of Crystal Enterprise Standard. Please use the Crystal Registration Wizard located in the Crystal Enterprise Program Group to complete your registration.

Making the right design choices

This section offers design options and considerations that will help improve the performance of your reports. The topics covered range from basic suggestions, such as updating older reports to the fastest file format, to more involved decisions, such as whether to use live or saved data, and how to use subreports efficiently.

When designing your reports, and especially when designing reports for the Web, you should allow report users to drive the data they see. In other words, display summarized information, so each user can navigate the report quickly and then drill down to access additional data. In this way, web traffic and response times are minimized, because only the data requested by the user is transferred from the database server.

These are only a few of the benefits of designing user-driven reports:

- report users gain interactive control over the type and quantity of information they view over the Web
- data transfer and network traffic decrease, because only the information requested by users is returned from the database server
- when users need real-time reporting of live data over the Web, user-oriented reports respond quickly and communicate efficiently with the database server
- reports become more useful, because each user customizes the report's contents, thereby creating a reporting solution specific to his or her particular decision-making problem.

Using faster report formats

The quickest way to help improve the performance of your old reports is to save them in the latest Crystal Reports format. Crystal Reports has been enhanced to process version 8 reports faster than ever before: update your older reports to take advantage of these enhancements.

To update the format of an older report, just open it in Crystal Reports and select Save from the File menu. The older version of the report will be replaced with a version 8.5 report.

Note: If for some reason you need to keep an old report in its original file format, use the Save As command (instead of Save) and enter a new name for the version 8.5 report.

Choosing between live and saved data

When reporting over the Web, the choice to use live or saved data is one of the most important decisions you'll make. Whichever choice you make, however, Crystal Reports displays the first page as quickly as possible, so you can see your report while the rest of the data is being processed.

Live data

Live reporting gives users real-time access to live data, straight from the database server. Use live data to keep users up-to-date on constantly changing data, so they can access information that's accurate to the second. For instance, if the managers of a large distribution center need to keep track of inventory shipped on a continual basis, then live reporting is the way to give them the information they need.

In conjunction with Crystal Enterprise, Crystal Reports supports live reporting. However, you should first consider whether or not you want all of your users hitting the database server on a continual basis. If the data isn't rapidly or constantly changing, then all those requests to the database do little more than increase network traffic and consume server resources. In such cases, you may prefer to use reports with saved data.

To ensure the efficiency of real-time reporting, read all of the suggestions in this chapter. These topics, however, are of particular importance:

- "Taking advantage of on-demand subreports" on page 111.
- "Performing grouping on server" on page 121.
- "Incorporating parameter fields into record selection formulas" on page 118.

Saved data

Reports with saved data are useful for dealing with data that isn't continually updated. When users navigate through reports with saved data, and drill down for details on columns or charts, they don't access the database server directly; instead, they access the saved data. Consequently, reports with saved data not only minimize data transfer over the network, but also lighten the database server's workload.

You can schedule these reports within Crystal Enterprise, so they automatically refresh from the database on a predetermined basis. For example, if your sales database is only updated once a day, or once a week, then you can run the report on a similar schedule and save it with data. Sales representatives then always have access to current sales data, but they aren't hitting the database every time they open a report. Alternatively, you can refresh reports with saved data on an as-needed basis.

If you do use reports with saved data, incorporate the other suggestions in this chapter to make sure your reports are designed for optimum performance.

To save a report with data, first make sure the Save Data with Report option is selected on the File menu; then Save your report.

Designing summary reports

Designing and distributing summary reports is a relatively easy way to ensure that users quickly find the data they need over the Web. A summary report can include as much data as any other report. However, by hiding a summary report's Details section, you avoid overwhelming users with data they may not immediately need.

When the Details section is hidden, users navigate with the Group tree first, to locate the desired data. Then, by drilling down on the report, they can request specific data, which is returned quickly without unnecessary records. This is especially important to improving navigation of long summary reports, which might consist of hundreds, thousands, or even tens of thousands of pages.

To facilitate navigation in this way, you first need to group the data and insert the summary fields you want to include in your report. For detailed information and instructions, see "Grouping data" on page 140 and "Summarizing grouped data" on page 156.

Once you've grouped and summarized your report data, hide the Details section (and any other large report sections) so that users can easily navigate to the data that's important to them.

Hiding details in a summary report

1. Open your report in Crystal Reports.

   If you haven't yet created a grouped and summarized report, open "summary group.rpt" from the Feature Examples samples folder.
2. On the **Format** menu, click **Section** to open the Section Expert.
3. In the **Sections** list, click **Details**.
4. On the Section Expert's **Common** tab, select the **Hide (Drill-Down OK)** check box.
5. Click **OK**.

You'll notice the details are now hidden. To view the details, navigate through the report using the Group tree, and then drill down on the appropriate area of the report.

Note: For details on minimizing data transfer with summary reports, see "Performing grouping on server" on page 121.

Using subreports carefully

For general information on subreporting, see "Subreports" on page 493.

Taking advantage of on-demand subreports

If your report has a section that handles a large number of records, you can put that section into an on-demand subreport. An on-demand subreport appears as a hyperlink in the primary report. When you open the primary report, no data is retrieved for the on-demand subreport until you drill down on the hyperlink.

For example, when designing a report that shows each employee's quarterly sales for each product and each product type, you might also want to track each employee's progress by including weekly sales information. This additional data, however, may not be of interest to many users viewing the report. In such a case, extract the weekly sales portion of the report and attach it as an on-demand subreport. Detailed information about weekly sales is retrieved from the database only when a user drills down on the on-demand subreport.

Many report objects—such as large cross-tabs, advanced charts, and maps—are ideal candidates to include in on-demand subreports, so that the object is not processed until you drill down.

To insert an on-demand subreport, see "Inserting subreports" on page 497 and "Creating an on-demand subreport" on page 506.

Tip: You could also place such report objects in a hidden Details section of a report that uses the Perform Grouping On Server option. When you do this, the database server performs the majority of the processing, and only a subset of the records is transferred from the server to the local machine.

Linking tables instead of linking subreports

Whenever possible, coordinate your report data by linking database tables with the Visual Linking Expert, rather than by linking regular subreports—that is, subreports that aren't on-demand subreports—to the main report. Since each subreport is run as a separate report, linked tables often have a performance advantage.

Related topics

For further general guidelines, see "Database links vs. subreports in one-to-many situations" on page 496.

For complete details, see "Performance considerations in one-to-many links" on page 517.

Using linked subreports

When a subreport is linked, Crystal Reports coordinates the data in the subreport with the matching records in the primary report. If you need to use regular linked subreports—that is, linked subreports that are not on-demand subreports—you should consider these guidelines:

- If the additional data provided by a regular linked subreport is useful to relatively few users, create a linked on-demand subreport instead. Users who need to see the extra data click a hyperlink to view the subreport; users who don't need to see the extra data won't have to download it from the database server.
- In some cases, placing regular linked subreports in the Details section of a main report may hinder performance—especially when your main report contains many records. (This is because you're creating a separate subreport for each record, and a separate query must be run for each database record in the main report.) As an alternative, consider using linked on-demand subreports in the Details section of your main report.

Streamlining your reporting environment

Another step to ensuring that report users receive their information quickly is to assess your reporting environment. What kind of database do you use? How is data organized within the database? How are you connecting to the data you need to report off? How are you linking your database tables? By taking these important considerations into account, you can significantly minimize the amount of data that must physically travel across the network.

Selecting the fastest database and connection

To improve reporting performance, utilize your database to its full potential by having it do the majority of your report processing. Ideally, Structured Query Language (SQL) databases are the most efficient for carrying out this task.

For details on SQL databases, see "Using SQL and SQL databases" on page 535.

Using table indexes

You can also improve the way your data is organized within the database. For optimum processing speed, report off indexed fields on your SQL database. Using table indexes is an easy way to increase the speed of data access and to reduce the time it takes for Crystal Reports to evaluate data.

For a complete description of how table indexing works, see "Indexed tables" on page 514.

Improving table-linking choices

When you add multiple database tables to your report, you link the tables on a common field so that records from one table match related records from another. (Table linking works best if your database tables are indexed.) Linking your database tables in this way is usually much faster than incorporating linked subreports into your main report.

When you link two or more tables, you want your report to read as few records as possible, while at the same time finding all matching records. The best ways to adhere to this guideline are to assess your reporting needs and to plan your strategy before creating your report. When you know exactly what you need from your data source, Crystal Reports makes it easy to get that information.

There are many other specific issues to consider when linking tables. These additional considerations, however, are largely contingent upon your reporting environment. In other words, the steps to optimal table-linking performance depend upon database type, the possibility of indexing tables, and the join type required between tables. You'll find complete descriptions for various reporting scenarios in the section entitled "Linking tables" on page 515.

In the majority of reporting scenarios, the following general procedures should ensure that your tables are linked for improved if not optimal performance.

To link tables effectively

1. Ensure that each database table is indexed on the field you are going to use.
2. Add the database tables to your report, and then link from the primary table to the lookup table on a common field. (Do this instead of inserting a linked subreport and binding it to the data in your primary report).
3. Use a record selection formula that sets range limits on the indexed field in the primary table. This minimizes the number of records in the primary table for which Crystal Reports must find matching records in the lookup table.

Related topics

For complete details on table-linking scenarios, see "Linking tables" on page 515.

For general information on record selection formulas, see "Selecting records" on page 126.

For advanced record selection strategies, see "Using enhanced record selection formulas" on page 115.

Using thread-safe database drivers

If you share and refresh reports over the web, open them using thread-safe database drivers whenever possible. The Crystal Report Engine supports multiple threads. Thus, when you make multiple data requests through thread-safe database drivers, the Report Engine can process all of the requests simultaneously. As the result, you'll be able to view your reports sooner.

Currently, the following ODBC drivers for SQL Server, Oracle, Sybase, and Informix are thread-safe:

- Sqlsrv32.dll (MS SQL Server)
- Crinf13.dll, Crinf14.dll, Crinf913.dll, Crinf914.dll (Merant drivers for Informix)
- Cror713.dll, Cror813.dll, Cror714.dll, Cror814.dll (Merant drivers for Oracle)
- Crsyb13.dll, Crsyb14.dll (Merant driver for Sybase).

Further, the following native database drivers are thread-safe:

- p2smon.dll (ADO)
- p2soledb.dll (OLE DB)
- p2sora7.dll (Oracle)
- p2ssyb10.dll (Sybase)
- p2ldb2.dll (DB2).

Using stored procedures for faster processing

A stored procedure is a compiled SQL program, consisting of one or more SQL statements, which resides and runs on your SQL server. Although stored procedures can take time to set up properly, they can be incredibly powerful.

The advantages of using stored procedures are most noticeable when reporting off large sets of data, or when running reports that demand long, complex calculations. In such cases, you should ideally use a predefined stored procedure that performs the complex work for you on the database server.

Because stored procedures reside on your SQL server, you may need to ask your Administrator about accessing or setting up a stored procedure.

Related topics

For more information on stored procedures, see "Stored procedures" on page 538.

To select a stored procedure, see "SQL stored procedures" on page 562.

Using enhanced record selection formulas

The most important thing you can do to speed up report processing is to limit the amount of data that is returned from the database. Your primary tool for doing this is the record selection formula.

Crystal Reports analyzes your record selection formula and generates an SQL query from it. This SQL query is then processed by the database, which sends the resulting records back to Crystal Reports. Crystal Reports then evaluates the record selection formula locally for each of the records retrieved from the database, thereby calculating the set of records used to generate the report.

Unnecessary records are eliminated at two stages: at the database with the SQL query and within Crystal Reports by the record selection formula. For speed, you want as many records as possible to be eliminated in the first stage. By designing your record selection formula effectively, you can off-load much of the processing to the database server, thus eliminating unnecessary records before returning them to Crystal Reports. This is commonly called “pushing record selection down to the database server.”

This section offers several tips to ensuring that your record selection formulas can be pushed down to the database server.

Pushing down record selection—an example

This example demonstrates the benefits of writing record selection formulas that can be pushed down to the database server.

In the Orders table of the Xtreme sample database, there are 2,001 records, of which 169 have order dates prior to 1997. Suppose you want to report on only those records. On the one hand, you could use this record selection formula:

```
Year ({Orders.Order Date}) < 1997
```

The SQL query generated will send all 2,001 records to Crystal Reports, and then the record selection formula will reduce this to 169. To see this, click Show SQL Query on the Database menu and notice that the SQL query has no WHERE clause. This is because Crystal Reports is not able to push down the Year ( ) function in the WHERE clause.

On the other hand, this record selection formula generates the same report:

```
{Orders.Order Date} < #Jan 1, 1997#
```

This second formula, however, can be performed on the database server, so it is pushed down. The SQL query generated will send only 169 records to Crystal Reports. So, when the record selection formula is evaluated by Crystal Reports, no further records need to be eliminated. Click Show SQL Query on the Database menu and notice that the resulting SQL query has a WHERE clause.

As this example shows, your report's processing speed improves when you enhance your record selection formula. In this case, both formulas generate the same report, but the second takes advantages of the power and optimizations that the database server can use when handling its own data.

Note: If you are new to record selection formulas, you may prefer to begin with the Select Expert or the sample record selection formula templates. For further details, along with an introduction to record selection, see "Selecting records" on page 126.

Record selection performance tips

Consider the following performance-related items when setting up record selection requests:

General

- To push down record selection, you must select "Use Indexes or Server for Speed" in the Report Options dialog box (available on the File menu).
- In record selection formulas, avoid data type conversions on fields that are not parameter fields.

  For example, avoid using ToText( ) to convert a numeric database field to a string database field.
- You are able to push down some record selection formulas that use constant expressions.

PC Databases

- You can only push down record selection on indexed fields.
- You can only push down AND clauses (not OR).

SQL Databases

- You can push down record selection on indexed or non-indexed fields.
- Your SQL server will respond faster if you use indexed fields.
- You can push down AND and OR clauses.
- Record selection formulas containing some types of embedded formulas can be pushed down.
- You should incorporate SQL Expression fields to push down formula calculations necessary for the record selection.

  For example, a record selection of `{@ExtendedPrice > 1000}` where `@ExtendedPrice = (Orders * Quantity)` can't be pushed down. However, if `@ExtendedPrice` is replaced with an SQL Expression field, then it will be pushed down.
- On the Database menu, click Show SQL Query to view the SQL that will be sent to the database server.

Strategies for writing efficient record selection formulas

Note: This section assumes that you are familiar with the Select Expert and that you are reporting off of a SQL database.

Consider the following points when creating a record selection formula:

Consideration 1

Any record selection formula that you generate completely with the Select Expert, without writing pieces of the formula yourself, can be pushed down.

This case actually follows from the points below. However, you can write more types of record selection formulas using the tips below than is possible with the Select Expert. To do this, you need to edit the record selection formula directly with the Formula Editor or from within the large text window in the Select Expert.

To open the Record Selection Formula Editor, click the Report menu, point to Edit Selection Formula, and then select Record from the submenu.

Consideration 2

Any selection formula that is of the form: *DataBaseField SupportedOperator ConstantOrParameterExpression* can be pushed down.

Of course *DataBaseField* is just a database field. *SupportedOperator* is any of =, <>, <, <=, >, >=, StartsWith, Like or In.

ConstantOrParameterExpression is any expression that involves constant values, operators, functions, and parameter fields. It cannot involve variables, control structures, or fields other than parameter fields. By their very definition, constant and parameter expressions can be evaluated without accessing the database.

Note: A constant or parameter expression can evaluate to a simple value, a range value, an array value, or an array of range values. Here are some examples of such expressions:

```
{?number parameter} - 3
Year ({?run date})
CurrentDate + 5
DateDiff ("q", CurrentDate, CDate("Jan 1, 1996"))
Month (Maximum ({?date range parameter}) + 15)
["Canada", "Mexico", "USA", {?enter a country}]
1000 To 5000
[5000 To 10000, 20000 To 30000, 50000 To 60000]
```

A complete example:

{Orders.Order Date} >= CurrentDate - 3

The program can also push down an expression that just contains a Boolean field (without the operator and constant parts).

```
{Orders.Shipped}
Not {Orders.Shipped}
```

Consideration 3

IsNull (*DataBaseField*) can be pushed down.

Consideration 4

SqlExpression SupportedOperator ConstantOrParameterExpression can be pushed down.

For example, the selection formula `{@ExtendedPrice} > 1000` cannot be pushed down if `{@ExtendedPrice} = (Quantity * Price)`. However, if the formula `@ExtendedPrice` is replaced with the equivalent SQL Expression, then your record selection formula will be pushed down.

Consideration 5

When using multiple expressions that follow the above considerations, separate them with AND and OR operators. You can have several of each, and you can use parentheses to give priority. You can also use NOT. For example:

```
{Orders.Order ID} < Minimum({?number range}) Or
{Orders.Order Amount} >= 1000
```

```
(IsNull({Customer.Region}) Or
 {Customer.Region} = "BC") And
{Customer.Last Year's Sales} > 2000
```

Related topics

If your record selection formula does not respond as expected, see "Troubleshooting record selection formulas" on page 134.

Incorporating parameter fields into record selection formulas

Instead of displaying all of a report's data every time the report is opened, you can create parameter fields that prompt users to specify the data they want to see. To decrease the amount of data transferred from the database server, incorporate these parameter fields right into your record selection formula.

In general, parameter fields provide interactivity for users, who respond to the parameter prompts in order to specify the data they want to see. However, by incorporating your parameter fields right into your record selection formula, you not only provide interactivity, but you also decrease data transfer and increase performance.

You can add a parameter field to your record selection formula by using the Select Expert or the Record Selection Formula Editor. When using the Record Selection Formula Editor, you treat the parameter field as you would any other field.

In the following example, you will create a parameter field and then add it to the record selection formula with the Select Expert. For further details on designing and creating parameter fields, see "Parameter Fields" on page 477.

To add a parameter field to your record selection formula

1 Open the sample report **group by intervals.rpt** (located in the Feature Examples sample folder) in the Preview tab.

Take a moment to navigate the Group tree and see how the data is organized; notice also, in the bottom right of the Crystal Reports window, that 270 records were returned for this report. Now click the Design tab.

2 On the **Insert** menu, click **Field Object**.

3 In the Field Explorer, right-click **Parameter Fields** and select **New** from the shortcut menu. The Create Parameter Field dialog box appears.

4 For the **Name** of the parameter, type `SalesQuota`.

5 For the **Prompting text**, type `What was last year's sales quota?`

6 Click the **Value type** list and select **Number**.

7 Make sure **Discrete value(s)** is selected, and then click **OK**.

Note: You've now created the parameter field. The rest of these procedures describe how to add the field to the record selection formula with the Select Expert.

8 On the Report menu, click **Select Expert**. The Choose Field dialog box appears.

9 Select the **Customer.Last Year's Sales** field, and then click **OK**. The Select Expert appears.

10 On the **Customer.Last Year's Sales** tab, click the drop-down list and select **is greater than**. A new drop-down list will appear.

11 Click this second list and select **{?SalesQuota}**.

Tip: Click the Show Formula button to view the new record selection formula, which appears as: `{Customer.Last Year's Sales} > {?SalesQuota}`. Instead of using the Select Expert, you could have created this formula yourself in the Record Selection Formula Editor. (To open the Editor, click Report > Edit Selection Formula > Record.)

12 Click **OK** in the Select Expert.

You've now added your parameter field to the record selection formula. When you switch to Preview mode, or click the Refresh button, you will be prompted for new parameter values. You can then enter a numeric value representing last year's sales quota. The resulting report will display only those customers whose Last Year's Sales exceeds the numeric value you specify.

For instance, if you respond to the parameter by entering 40000, then the report will display only those customers whose sales exceeded $40,000 last year. Notice also, in the bottom right of the Crystal Reports window, that only 59 records are returned for your report, instead of the 270 records that were returned before you put the parameter field into a record selection formula.

By enhancing this report, you've retrieved all the information you needed and, at the same time, you've ensured that the fewest number of records is transferred from the database server.

Related topics

To create enhanced parameters, see "Advanced parameter features" on page 485.

For general information about parameter fields, see "Parameter overview" on page 478.

For general details regarding record selection formulas, see "Record Selection" on page 125.

Using SQL expressions where appropriate

SQL expressions are like formulas, but they are written in Structured Query Language (SQL). They are useful in optimizing report performance because the tasks they execute are normally performed on the database server (as opposed to a regular formula, which is sometimes executed on the local machine).

Don't use SQL expressions exclusively, because Crystal Reports includes its own formula language, which is much more powerful than standard SQL. Both Crystal Syntax and Basic Syntax allow you to enhance and hone your formulas in ways that are either difficult or impossible with SQL. In certain circumstances, however, creating and using SQL Expression fields can speed up the processing of your reports.

Key uses of SQL Expression fields

To maintain optimum report processing speeds, avoid using formulas (whether Crystal or Basic syntax) within record selection formulas. Instead, replace the original formula with an equivalent SQL Expression field, and then incorporate the SQL Expression field into your record selection formula. Doing so will greatly improve the chances of your record selection being pushed down to the server.

In addition, avoid sorting, grouping, or totaling on a formula field (whether Crystal or Basic syntax). Instead, replace the original formula field with an equivalent SQL Expression field, and then sort, group, or total on the SQL Expression Field. Again, this will greatly improve the chances of the processing being done on the server.

Finally, if your database supports Case Logic, and your report needs to summarize an If-Then-Else formula calculation, replace the formula with an SQL Expression field. In such cases, SQL Expression fields enable Crystal Reports to perform the report's grouping on the server. For more information, see "Using SQL Expressions for Case Logic" on page 123.

Related topics

For an introduction to SQL, see "What is SQL?" on page 535 and "The SQL language" on page 539.

For instructions on creating an SQL Expression field, see "SQL Expression fields" on page 95.

For further tips on enhancing your record selection formula, see "Using enhanced record selection formulas" on page 115.

For general record selection procedures, see "Record Selection" on page 125.

Improving grouping, sorting, and totaling

Performing grouping on server

If you're reporting in real-time off live data over the Web, reduce the amount of data transferred from the database server by using the Perform Grouping on Server option. With this option, much data processing is off-loaded to the database server and only a subset of data is read initially. Detail data is returned from the database only when you drill down in a report.

Note: Server-side processing works only for sorted and grouped reports that are based on SQL data sources.

To enable server-side processing

1 On the **File** menu, click **Report Options**.

2 Select **Perform Grouping on Server** on the Report Options dialog box.
This check box is inactive if Use Indexes or Server for Speed is not selected.

Tip: When Use Indexes or Server for Speed is selected, you can quickly enable Perform Grouping On Server from the Database menu.

3 Click **OK**.

Benefits of grouping on server—an example

The following example describes a typical reporting scenario wherein grouping on server can significantly reduce the amount of unnecessary data transferred from the database server.

To decrease data transfer by grouping on server

1 Open the sample report **group.rpt** (located in the Feature Examples folder).
Look in the bottom right corner of the Crystal Reports window and notice that 270 records are returned for this report.

2 On the **Format** menu, click **Section** to access the Section Expert.

3 In the **Sections** list, select **Details**.

4 On the **Common** tab, select **Hide (Drill-Down OK)**. This hides the report's Details section, so that only group header displays in the report. (In this case, the report is grouped by Country.)

5 Click **OK** in the Section Expert. The Details records will disappear from the report.

6 Press F5 to refresh the report data (or click the **Refresh** button on the toolbar).

Notice that 270 records are still returned for the report, even though the Detail records are hidden from view.

7 On the **Database** menu, click **Perform Grouping on Server**.

Now notice that only 72 records are returned for the report. The grouping (by Country) has been performed on the database server, and fewer records have been transferred to the report.

8 Drill down on a country by double-clicking the report. Crystal Reports retrieves the appropriate Detail records as you need them.

For instance, if you drill down on Australia, Crystal Reports quickly retrieves the seven records that make up the group.

By hiding the Details section of this report, you've created a summary report that's easy for users to navigate. Each user can first locate the Country with which he or she is concerned, and can then drill down to retrieve valuable details.

Moreover, by enabling the Perform Grouping on Server option, you've ensured that the initial processing is completed on the database server. Consequently, only the necessary records are transferred to the report.

For more information on server-side processing, see "Server-side processing" on page 541.

Using SQL Expressions for groups, sorts, and totals

For reports using Perform Grouping on Server, avoid sorting, grouping, or totaling on a formula field (whether Crystal or Basic syntax). Instead, replace the original formula field with an equivalent SQL Expression Field, and then sort, group, or total on the SQL Expression field. This will greatly improve the chances of the processing being done on the server.

For details on when else to use SQL Expressions, see "Using SQL expressions where appropriate" on page 120.

Using SQL Expressions for Case Logic

If your database supports Case Logic, and your report needs to summarize an If-Then-Else formula calculation, replace the formula with an SQL Expression field. In such cases, SQL Expression fields enable Crystal Reports to perform the report's grouping on the server.

For instance, suppose that you're reporting off of an MS SQL Server 7 database, which supports Case Logic. You need to include an If-Then-Else calculation in your report, and you need to summarize that calculation for each group in the report. By performing the calculation with an SQL Expression field of the following form, you use your database's ability to process Case Logic:

```
CASE DatabaseTable.”DatabaseField”
WHEN 'SpecifiedValue' THEN Calculation1
ELSE Calculation2
END
```

If a record's `DatabaseField` value is equal to `SpecifiedValue`, then `Calculation1` is performed; for all other records, `Calculation2` is performed. By incorporating the SQL Expression field, you take advantage of your database server's ability to process Case Logic. Your report's grouping consequently takes place on the server, even when you summarize the SQL Expression field elsewhere in the report.

Note: The SQL syntax in this example is specific to MS SQL Server 7. You may need to refer to your database documentation or consult with your Administrator in order to determine the syntax appropriate to your database.

Inserting summary and running total fields where possible

Where possible, avoid creating formulas with global variables to calculate summaries or running totals.

Instead, create summaries by clicking the Insert menu and then selecting the appropriate command (Subtotal, Grand Total, or Summary). Create running total fields by opening the Field Explorer, right-clicking Running Total Fields, and then selecting New from the shortcut menu.

For general details on summarizing data, see “Sorting, Grouping, and Totaling” on page 137 and “Running Totals” on page 171.

Record Selection 8

This chapter shows you how to filter the records you want included in a report. For example, using the record selection tools, you can limit the records in your report to include only records for a specific group of customers, a specific range of account numbers, or a particular date range.

Selecting records

When a field is selected to appear on a report, field values from every record in the active table(s) are printed by default. 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
- records for a specific range of account numbers out of the total number of records in the database
- values from only those records that fall within a particular date range.

Options for selecting records

Crystal Reports includes a very sophisticated formula language that you can use to specify virtually any type of record selection. However, you may not always need the flexibility in record selection that the formula language provides. The Select Expert is designed for such instances.

You can select records in one of two ways:

- "Using the Select Expert" on page 127.
- "Using formulas" on page 128.

Once you are familiar with the Select Expert and formulas, you can use your record selection techniques to improve the performance of your reports. For additional performance tips and advanced record selection strategies, see "Using enhanced record selection formulas" on page 115.

Determining which field(s) to use

When you select records, you are basing your report only on those records that meet some conditions that you have set. You base those conditions on the kind of information you want in the finished report.

Assume, for example, that you want a report that only shows data from California. The challenge is to find the best way to identify those records that come from California.

- If the table used in a report has a State or Region field, you can specify in your request that the program use only those records in which the value in the state field is equal to California (Region is equal to CA).
- If the table does not have a State field and you still want to report only on California data, there may be another way to identify that data.
    - If the table has a Postal 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*).
    - If the table has an Area Code field, you could base your record selection on California Area Codes (Area Code is one of x, y,....z).

Note:

- If the Area Code is stored in the telephone number field, you will not be able to do this same record selection using the Select 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 based on that.
- As a general rule, if you can base your record selection on a number of fields (as in this example), you should select an indexed field instead of a field that is not indexed.

Using the Select Expert

The Select Expert makes it easy to specify the records you want included in your report. When you work with the Select Expert, you select the field to which you want to apply selection conditions and then you specify those conditions.

The Select Expert can be used to set up simple record selection requests. For example:

- customers from Arizona
- orders in the first quarter
- sales over $10,000.

The Select Expert can also be used to set up some sophisticated requests:

- customers whose names start with “A”, “M”, or “S”
- customers from California or Florida who ordered in July.

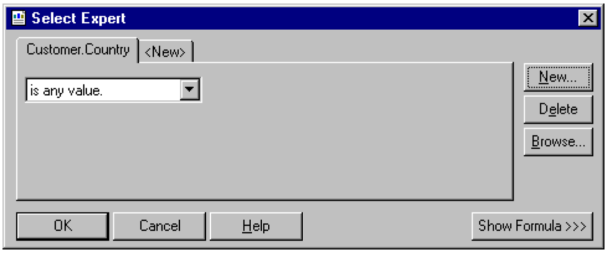
These are all range limit requests. One or more constants define the range. The program compares the field value in each record to the constant(s) and rejects records with values outside the range. The report is limited to values within the range. You can set up all of these types of record selection requests without any previous knowledge of the formula language.

Note: The Select Expert can be used to set up both record selection and group selection requests. When a group name or summary field is selected, the program knows that the selection criteria set up is intended for group selection. In all other cases, the program knows that you are setting up record selection.

To set up record selection using the Select Expert

1 On the **Report** menu, click **Select Expert**.

The Select Expert dialog box appears.



Note: If you click the Select Expert button without first highlighting a field in your report, the Choose Field dialog box appears. Highlight the field on which you want to base record selection and click OK. The Select Expert appears.

2 Use the drop-down lists to enter your selection criteria for the indicated field.

3 Click **OK** when finished.

Tip: To base record selection on more than one field, click the New tab. Select the next field from the Choose Field dialog box.

A selection formula is generated based on your specifications, limiting the report to the records you indicated.

Note: To view the selection formula, click the Show Formula button. The Select Expert expands to show the formula. Click the Formula Editor button to modify the formula.

Using formulas

To set up record selection using a formula

1 On the **Report** menu, point to **Edit Selection Formula** and click **Record**.

2 Enter the formula by typing in the components or selecting them from the component trees.

Note: The resulting formula must be Boolean; that is, it must return either a True or False value.

3 Click **Check** to identify any errors in the formula.

4 Fix any syntax errors the Formula Checker identifies.

5 When the formula has the correct syntax, click **Save and Close**.

Related topics

For sample record and group selection templates, see “Using formula templates” on page 130.

For complete instructions on creating formulas, see “Using Formulas” on page 381.

For advanced record selection strategies and performance tips, see “Using enhanced record selection formulas” on page 115.

Interaction of the Select Expert and the Selection Formula Editor

The Select Expert and the Selection Formula Editor are interactive. That is, record selection criteria you enter via the Select Expert automatically generates a record selection formula which can be reviewed and modified. 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 tools together as a tutorial for learning the formula language.

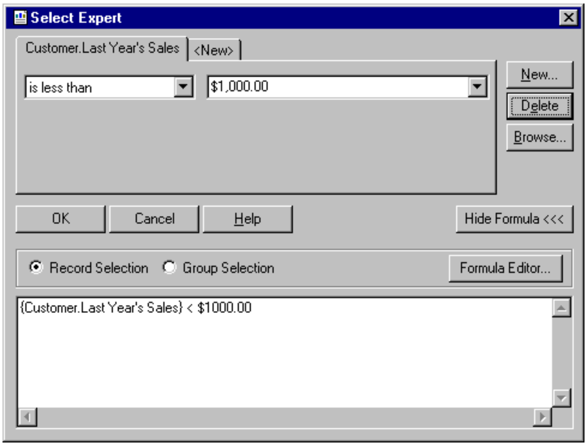
To view the Select Expert formula

1 Right-click the field on which you want to view record selection.

2 Click **Select Expert**.

3 Click the **Show Formula** button.

The Select Expert expands so you can review the formula the program generated based on your selection criteria.



is less than | $1,000.00

● Record Selection ○ Group Selection

```
{Customer.Last Year's Sales} < $1000.00
```

4 Click the **Hide Formula** button when you are done with your review.

5 Use the Select Expert to change your selection formula.

6 Review the updated formula by clicking the **Show Formula** button again.

7 To make changes to the formula, click the **Formula Editor** button in the expanded Select Expert and use the tools in the Record Selection Formula Editor to make your formula changes.

Note: Selection formula components that do not fit any of the fixed criteria in the Select 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 performs that extraction will not be converted to Select Expert selection criteria.

Using formula templates

Record selection formula templates

The following sample formulas can be used as templates to help you create your own selection formulas using the Record Selection Formula Editor. These examples illustrate different selections that you *could* do, not necessarily the best selections.

Selecting records using character strings

```
{file.FIELD} startswith “C”
```

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 ({file.FIELD} startswith “C”)
```

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.).

Selecting records using numbers

Single values

```
{file.FIELD} > 99999
```

Selects those records in which the value in the {file.FIELD} field is greater than 99999.

```
{file.FIELD} < 99999
```

Selects those records in which the value in the {file.FIELD} field is less than 99999.

Range of values

```
{file.FIELD} > 11111 and {file.FIELD} < 99999
```

Selects those records in which the value in the {file.FIELD} field is 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 in which the value in the {file.FIELD} field is greater than 11111 but less than 99999 (both 11111 and 99999 are included in the range of values).

Selecting records using dates

The Month, Day, and Year functions can all be used in examples like the following:

```
Year ({file.DATE}) < 1999
```

Selects those records in which the year found in the {file.DATE} field is earlier than 1999.

```
Year ({file.DATE}) > 1992 and
Year ({file.DATE}) < 1996
```

Selects those records in which 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 in which the year found in the {file.DATE} field falls between 1992 and 1996 (1992 and 1996 are included).

```
Month({file.DATE}) in 1 to 4
```

Selects 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 [1,4]
```

Selects those records in which the month found in the {file.DATE} field is the first or fourth month of the year (includes January and April, excludes February and March).

Selecting records using preset data ranges

The preset date ranges can be used to create selection formulas similar to these:

```
{file.DATE} in LastFullMonth
```

Selects those records 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 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} < CurrentDate
```

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 this kind of formula was used 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.

These templates can be used as they are (with your own data), or they can be combined to create complex formulas.

Choosing a formula template

To choose a formula template

1 Select the template you want to use.

You can do this in one of two ways:

- review the list in "Record selection formula templates" on page 130, and write down the formula of interest
- find the Record Selection Formula Templates in the online help file and copy the formula of interest to the Clipboard.

2. Choose **Edit Selection Formula** from the **Report** menu.
3. Choose **Record** to create a record selection formula.

   The Record Selection Formula Editor appears.

   - or -

   Choose **Group** to create a group selection formula.

   The Group Selection Formula Editor appears.
4. In the **Formula text box**, type in the formula you wrote down in Step 1 or paste it from the Clipboard.
5. 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 for which the value in the {orders detail.QUANTITY} field is greater than 25, replace the existing values with the values you want, so that your selection formula reads:

   ```
   {orders detail.QUANTITY} > 25
   ```
6. Click the **Save and Close** button when finished to exit the Selection Formula Editor.

   - or -

   Click **Refresh** on the Standard toolbar to preview the results.

Pushing down record selection to the database server

The drivers provided with Crystal Reports for SQL data sources allow “pushing down” record selection to the database server. When you specify a record selection formula in a report based on an SQL data source, Crystal Reports analyzes it, generates a SQL query from it and passes the SQL query to the database server. Record selection is then carried out in two stages:

- the first stage of record selection takes place when the database server processes the SQL query and returns a set of records to Crystal Reports
- in the second stage, Crystal Reports further evaluates locally the record selection formula for the set of records returned from the database server.

Since database servers are usually faster machines than workstations, it is to your advantage to specify record selection formulas that can be processed by the server in the first stage. Doing so minimizes record selection on the local machine in the second stage. This process is commonly known as “pushing record selection down to the database server.” The following kinds of record selections can be pushed down to the server:

- selections with indexed and non-indexed fields (indexed fields provide faster response).
- SQL queries with AND and OR clauses.

- SQL expression fields that carry out formula calculations for record selection. (For the types of SQL expressions supported by your SQL server, consult the documentation provided with the server.)

For complete details on pushing your record selection formulas down to the database server, see “Using enhanced record selection formulas” on page 115.

Troubleshooting record selection formulas

To troubleshoot your selection formula, you should first begin by making sure that all of the fields referenced in the selection formula are placed on your report. Then delete the selection formula and test it as you rebuild it, step by step.

To troubleshoot record selection formulas

1. Write down the record selection formula on paper. You will use this written copy to help you reconstruct the selection formula one step at a time.
2. Remove the record selection formula from your report by deleting the formula from the formula text box in the Record Selection Formula Editor.
3. Click **Save and Close** when finished.
4. Make certain that all fields referenced in the record selection formula (the selectors) are on the report physically and are not hidden.

   For example, if one of the selectors is:

   ```
   {customer.POSTAL CODE} > “80000”
   ```

   but the {customer.POSTAL CODE} field is not used on your report (as in the case of a sales report that uses the postal code to define territories but does not include the postal code in the report data), then insert the {customer.POSTAL CODE} field into the report.

   Or, if one of the fields referenced in the selection formula is on the report but is hidden, unhide it by deselecting the Suppress option in the Format Editor for that field.
5. Print the report and verify that the data in those fields which are referenced in the selection formula print satisfactorily. Make certain that all the data prints. For example, if there are *x* total records in the database you should have *x* records printing for each of the referenced fields. This establishes a baseline against which you can compare the results of printing with the selection formula.
6. When you are sure that you are getting satisfactory results without using the selection formula, you can enter the selection formula using only one of the selectors.

For example, to use this as the final selection formula:

```
{customer.POSTAL CODE} > “80000” and {customer.CONTACT LAST NAME}[1] =
“C” and {customer.LAST YEAR'S SALES} >= 5000
```

this formula will select all of those records that show a Postal code greater than 80000, a value in the {customer.CONTACT LAST NAME} field beginning with “C”, and a value in the {customer.LAST YEAR'S SALES} field greater than or equal to 5000.

You might start with this as the first test selection formula:

```
{customer.POSTAL CODE} > “80000”
```

Print the report and evaluate the data that prints when you have only one selector activated. Does the {customer.POSTAL CODE} field show only ZIP codes greater than 80000?

- if it does, then you know that this part of the selection formula is working
- if it does not, then troubleshoot this part of the selection formula.

7 Once the selection formula with one selector activated is working properly, add a second selector. For example, the new selection formula might look like this:

```
{customer.POSTAL CODE} > “80000” and _
      {customer.CONTACT LAST NAME}[1] = “C”
```

8 Print the report and evaluate the data that prints when you have two selectors activated. Evaluate the data in the {customer.CONTACT LAST NAME} field (since you already evaluated {customer.POSTAL CODE} in the last step).

Does the {customer.CONTACT LAST NAME} field show only text strings beginning with the letter “C”?

- if it does, then this part of the selection formula is working
- if it does not, then troubleshoot this part of the selection formula.

9 Once the selection formula with two selectors activated is working properly, add a third selector, then a fourth, etc., until you have tested each selector in the selection formula.

Correcting selections that do not generate data

You may encounter a situation in which you create a record selection formula and, while header and footer information prints on your report, no detail information appears. The problem is the selection formula is rejecting all records. This usually occurs because of an error in the creation of the selection formula.

There are a couple of potential causes of your problem in the selection formula:

- See “Correcting uppercase/lowercase inconsistency” on page 136.
- See “Unwanted spaces appear in selection formula” on page 136.

Correcting uppercase/lowercase inconsistency

Record selection formulas are case sensitive. That is, “Bob” matches only with “Bob”. It does not match with “bob”, “BOB”, “BoB”, “bOB”, “boB”, or “BOb”. Thus, if your selection formula is set to include only those records with “BOB” in the {customer.CONTACT FIRST NAME} field, but all the entries in the {customer.CONTACT FIRST NAME} field are mixed case (“Bob”, for example), the selection formula will find no matches and thus will not print any details for the report.

You can solve this problem by using the UpperCase (str) or LowerCase (str) functions in your selection formula to convert field data to a consistent case before the program begins its selection. For example, if you were using this formula:

```
{customer.CONTACT FIRST NAME} = “BOB”
```

you can change the formula to this:

```
UpperCase({customer.CONTACT FIRST NAME}) = “BOB”
```

This second formula first converts the value of the {customer.CONTACT FIRST NAME} field to upper case characters and then checks to see if the value in that field is equal to “BOB”. Using this formula, any instance of the three letters “b,” “o,” and “b” will be a match, regardless of case, because the case will be converted to uppercase.

You could use the LowerCase function in a similar manner to match with “bob.”

Check your selection formula closely and make sure you have the correct case for any text you are trying to match. If in doubt, use the UpperCase (or LowerCase) function to assure consistency and proper matching.

Another formula which performs much the same functions is the following:

```
“BOB” in UpperCase({customer.CONTACT FIRST NAME})
```

Unwanted spaces appear in selection formula

Spaces are characters, and when you include spaces in the search key of a record selection formula, the formula looks for records with the exact match in the selected field, spaces and all. For example, the following formula:

```
“Mr . “ in {customer.TITLE}
```

will not find any matches with the form of address “Mr.” because there is an extra space in the search key between the letter “r” and the period. Likewise, “Ph. D” will not match “Ph.D”.

Check your selection formula closely, and make sure that the spaces in the selection formula match the spaces in the fields you are trying to match.

Sorting, Grouping, and Totaling 9

Sorting, grouping, and totaling are the steps that turn disorganized data into useful information on a report. This chapter describes the types of sorting, grouping, and totaling you can do within a report.

Sorting data

Sorting means placing data in some kind of order to help you find and evaluate it.

When you first insert a database field into your report, the data within the fields appears in the order in which it was originally entered into the database. Locating information in this kind of report is difficult. It is much easier to review or find information when you can see it sorted in a logical format. For example, you may want to have a customer list sorted alphabetically by name or by country.

Understanding sort options

When you sort, the program asks you to define two things:

- the field you want the sort to be based on (sort field)
- the sort direction.

Sort field

A sort field is the field that determines the order in which data appears on your report. Almost any field can be used as a sort field, including formula fields. A field's data type determines the method in which the data from that field is sorted.

Note: You cannot sort on memo or BLOB fields.

Field Type	Sort Order
Single-character string fields	blanks punctuation numbers uppercase letter lowercase letters
Multiple character string field	two letters three letters four letters, etc. For example: ■ "BOB" comes before "bob" ■ "123" comes before "124" ■ " " (blank) comes before "a" ■ "aa" comes before "aaa"
Currency fields	numeric order
Number fields	numeric order
Date fields	chronological order
DateTime fields	chronological order same-date values sorted by time
Time fields	chronological order

Field Type	Sort Order
Boolean comparison fields	False values (0) True values (1)
Null values	null values non-null values

Sort Direction

Direction refers to the order in which the values are displayed, once sorted.

- **Ascending**
  Ascending order means smallest to largest (1 to 9, A to Z, False to True). The program sorts the records in ascending order based on the values in the sort field you select.
- **Descending**
  Descending order means largest to smallest (9 to 1, Z to A, True to False). The program sorts the records in descending order based on the values in the sort field you select.

Sorting single and multiple fields

In single field sorting, all the records used in the report are sorted based on the values in a single field. Sorting an inventory report by stock number or sorting a customer list by customer number are examples of single field sorts.

In multiple field sorts, the Report Designer 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, the program then sorts those records based on the value in the second sort field.

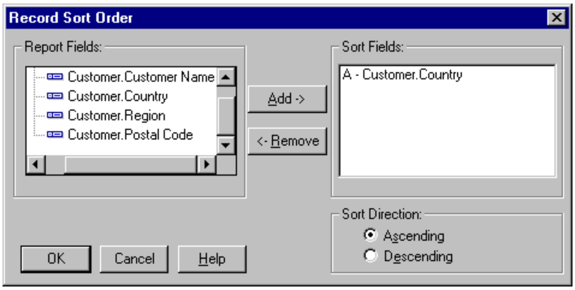
For example, if you choose to sort first by the {customer.COUNTRY} field and then by the {customer.REGION} field, both in ascending order, the report would appear with countries listed in alphabetic order, and regions within each country listed in alphabetic order. Any other fields, such as the postal codes within each region, would remain unsorted.

You create single or multiple field sorts using the same procedure.

To sort your data

1 On the **Report** menu, click **Sort Records**.

The Record Sort Order dialog box appears.



Customer.Customer Name
Customer.Country
Customer.Region
Customer.Postal Code

A - Customer.Country

● Ascending
○ Descending

2 Highlight the field to be sorted from the Report Fields list.

3 Click the **Add** button.

The selected field is added to the Sort Fields list.

4 Specify the sort direction.

5 If sorting by more than one field, highlight the second field you want the data to be sorted by and add it to the Sort Fields list.

Tip: The order of the fields listed in the Sort fields box is the order by which data will be sorted.

6 As you add each field to the Sort Fields list, specify the sort direction.

7 Click **OK** when finished.

Records are sorted based on the values in the Sort Fields list.

Grouping data

Grouped data is data that is sorted and broken up into meaningful groups. In a customer list, for example, a group might consist of all those customers living in the same Zip Code, or in the same Region. In a sales report, a group might consist of all the orders placed by the same customer, or all of the orders generated by a particular sales representative.

Group and sort direction

When data is grouped, four sort and group direction options are available. Direction refers to the order in which the values are displayed.

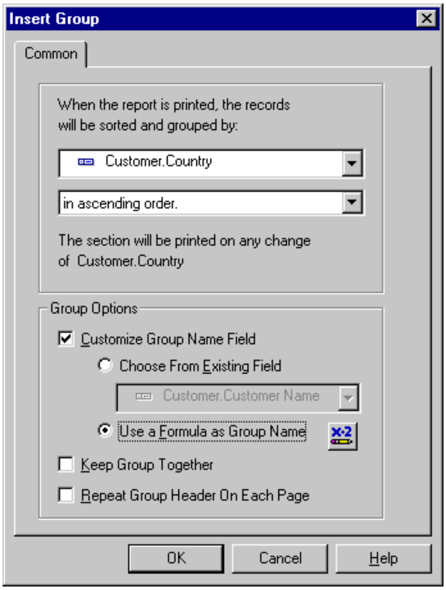
- **Ascending**
  Ascending order means smallest to largest (1 to 9, A to Z, False to True). The program sorts the records in ascending order and then begins a new group whenever the value changes.

- **Descending**
  Descending order means largest to smallest (9 to 1, Z to A, True to False). The program sorts the records in descending order and then begins a new group whenever the value changes.
- **Original**
  Original order is the order the data was originally saved in the database. The program leaves the records in the order in which they appear in their originating database table, and begins a new group whenever the value changes in the group field you select.
- **Specified order**
  Specified order is a user-defined order. The program places each record into the custom group you specify, leaving the records in each group in original order or it sorts them in ascending or descending order, depending on your instructions. See "Creating custom groups" on page 142.

To group data

1 On the **Insert** menu, click **Group**.

The Insert Group dialog box appears.



Insert Group

Common

When the report is printed, the records will be sorted and grouped by:

Customer.Country

in ascending order.

The section will be printed on any change of Customer.Country

Group Options

☑ Customize Group Name Field

○ Choose From Existing Field

Customer.Customer Name

● Use a Formula as Group Name

□ Keep Group Together

□ Repeat Group Header On Each Page

2 Select the field you want the data grouped by from the top drop-down list.

3 Select the sort direction from the second drop-down list.

4 Select the **Customize Group Name Field** check box if you want to show a different value in the group header.

Note: By default, the group header of the report will display the value of the field you are grouping on.

For example, if you grouped by {Customer.CustomerID}, at each change of a group you will see the corresponding customer ID. If you want to display a different value (customer ID as well as customer name), customize the group name field by choosing an alternate data field, or creating a formula.

Tip: To hide the group header name, right-click the group header, select Format Field, and click Suppress on the Common tab of the Format Editor.

5. Click **OK**.

   If your records within each group are unsorted, you need to sort the records within each group. See "Sorting records within groups" on page 144.

Related topics

"Grouping data in intervals" on page 148.

"Creating group headers" on page 165.

Creating custom groups

Usually data is sorted based on the values from a field in the report. Sometimes, however, you may not want to group data based on the values found in one of the fields on your report. For example:

- The field you want to group on does not exist.

  For example, your report may contain a City field and a State field, but no Country field, although you need to group by country.
- The field exists, but you do not want to group on the values in that field.

  For example, you may 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 shades 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.
- The field exists, 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 of selection facilities that are available to you for building record selection queries.

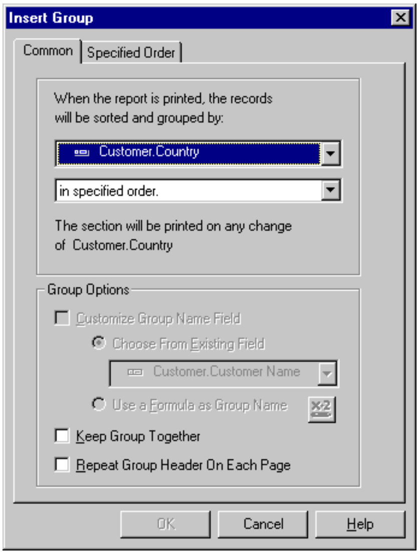
Specified order grouping provides a solution to these custom sorting and grouping challenges. It enables you to create both the customized groups that will appear on a report and the records that each group will contain. The only limitation is that a record can be assigned to only one group.

To follow a tutorial where you create custom groups to rank customers by the amount of business they did in the previous year, see "Grouping data in intervals" on page 148.

To create a custom group

1 On the **Insert** menu, click **Group**.
The Insert Group dialog box appears.

2 Select the field you want the data grouped by from the top drop-down list.



When the report is printed, the records will be sorted and grouped by:

Customer.Country

in specified order.

The section will be printed on any change of Customer.Country

Group Options

□ Customize Group Name Field

● Choose From Existing Field

Customer.Customer Name

○ Use a Formula as Group Name

□ Keep Group Together

□ Repeat Group Header On Each Page

3 Select **in specified order** as your sort option from the second drop-down list.

4 On the Specified Order tab, enter the name of the group in the **Named Group** field.

5 Click **New**.

6 In the Defined Named Group dialog box, use the drop-down lists to select the data to be part of the group.

7 Click the **<New>** tab to add more selection criteria to your specified group, if necessary.

8 Click **OK**.

9 Click **New** to create more custom groups as necessary.

10 Click the **Others** tab to specify how you want to organize the data that is not part of the group(s) you defined.

11 Click **OK**.

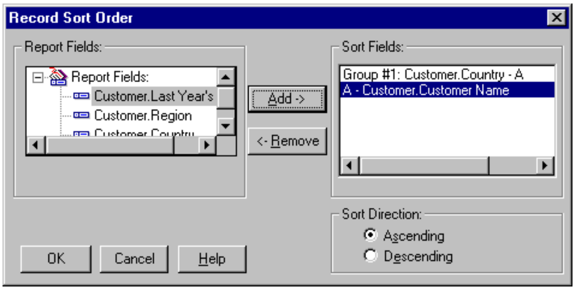
Sorting records within groups

Once you have grouped your data, you can easily sort the records within the groups to further organize the information.

To sort records within groups

1 Once the data is grouped, on the **Report** menu, click **Sort Records**.

The Record Sort Order dialog box appears.



Sort Fields:
Group #1: Customer.Country - A
A - Customer.Customer Name

Sort Direction:
● Ascending
○ Descending

Note: Sort fields that begin with "Group" specify that the sort was done automatically when the data was grouped.

2 Highlight the field you want the records within the groups sorted by and click the **Add** button to add it to the **Sort Fields** list.

Tip: The order of the fields in the Sort Fields box is the order by which data will be sorted.

3 Specify the sort direction.

4 Click **OK** when finished.

Group selection

When you group or summarize data, all the groups in the report are included by default. There may be times, however, when you do not want to include all the groups. For example:

- you might want to see only those groups that have certain group names, or whose summarized values meet a certain condition
- you might want to see only the groups with the highest summary values, or the lowest.

You can select the groups that appear in the report in two different ways:

- See "Using the Select Expert" on page 145.
- See "Using selection formulas" on page 146.

Note: For faster results, limit records through record selection before creating groups. See "Selecting records" on page 126.

Using the Select Expert

The Select Expert can be used to select groups of records in the same way that you select individual records.

When you are setting up group selection criteria, instead of basing the selection criteria on standard fields, as you do for record selection, you base the criteria on group name fields or summary fields.

- If you have grouped your data but have not summarized it, you can only set up group selection based on the group name field. For example, you may want to select only those groups whose Region is Massachusetts:

  ```
  GroupName ({Customer.REGION}) = “MA”
  ```

- If you have summarized your data, you can set up group selection based on either the group name field or the summary field. For example:

  ```
  Sum({Customer.LAST YEAR’S SALES}, {Customer.REGION}) > 10000
  ```

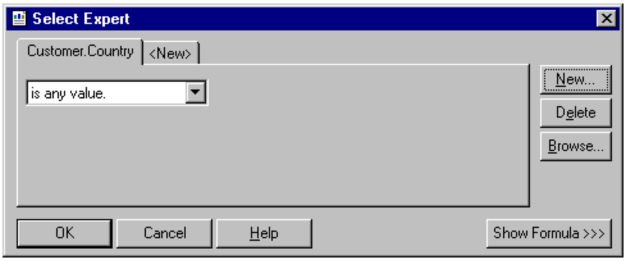
Note: The Select Expert can be used to set up record selection and group selection requests. When a group name or summary field is selected, the program knows that the selection criteria you set up is intended for group selection. In all other cases, the program knows that you are setting up record selection.

To set up group selection using the Select Expert

1 Right-click the summary field on which you want to base group selection and choose **Select Expert** from the shortcut menu.

The Select Expert dialog box appears.

Note: If you click the Select Expert button without first highlighting a summary field in your report, the Choose Field dialog box appears.



is any value.

2 Use the drop-down list to enter your selection criteria for the indicated field.

3 To base the group selection on more than one field, click the **New** tab and choose the next field from the Choose Field dialog box.

Note: If you have not already previewed the report or refreshed the data, there will not be any data saved with the report. Without the data, the program cannot calculate group values; thus, no values appear when you click the arrow in the right drop-down list. 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 available for you to work with.

4. When you are finished, click **OK** in the Select Expert to return to the report.

Using selection formulas

With the Formula Editor you can build your group selection request using group fields, group name fields, and other formulas. As 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.

To create a record or group selection formula

1. On the Report menu, point to **Edit Selection Formula**.
2. Click **Record** to create a record selection formula.

   The Record Selection Formula Editor appears.

   - or -

   Click **Group** to create a group selection formula.

   The Group Selection Formula Editor appears.
3. Enter your selection formula in the Formula text box.

   Note: The resulting formula must be Boolean; that is, it must return either a True or False value.
4. Click **Check** to identify any errors in the formula.
5. Fix any syntax errors the Formula Checker identifies.
6. When the formula has the correct syntax, click **Save and Close**.

   When the program runs the report, it will include only those records or groups of records that you specified.

Troubleshooting group selection formulas

In some cases, no values will print when using a group selection formula, even though there are values that match the selection criteria. Typically, in these cases:

- the group selection formula references another formula
- the referenced formula is one that calculates the value of each group as a percentage of the total value of all groups (in other words a subtotal as a percentage of a grand total).

To correct a group selection formula

1 Use **Xtreme.mdb** to create a report that includes the following fields:

```
{customer.CUSTOMER NAME}
{customer.REGION}
{orders.ORDER ID}
{orders.ORDER AMOUNT}
```

For each order, the report shows the customer that placed the order, the region in which that customer is located, the order ID number, and the amount of the order.

2 Subtotal the {orders.ORDER AMOUNT} field using {customer.REGION} as the sort-and-group-by field.

The program sorts the data by state and calculates a subtotal in the {orders.ORDER AMOUNT} field every time the state changes. See “Subtotaling data” on page 160.

3 Insert a grand total on the {orders.ORDER AMOUNT} field to see the total value of all orders placed.

4 Create a formula named Percent that calculates each subtotal as a percentage of the grand total to see the value of the orders for each region group as a percentage of all orders placed.

```
Sum({orders.ORDER AMOUNT}, {customer.REGION})
      % Sum({orders.ORDER AMOUNT})
```

5 Place the formula in the Group Footer section of the report.

6 Reference the formula (@Percent) in a group selection formula that selects only those groups for which the percentage (of subtotal to grand total) is less than 5% in order to find out which regions individually contributed less than 5% of total sales:

```
{@Percent} < 5
```

When you click **Save and Close**, you will receive the following error message:

```
This formula can not be used because it must be evaluated later.
```

7 Instead of using the formula name (in this case @Percent) in the group selection formula, enter the formula itself (the formula named @Percent). Thus, instead of using the group selection formula:

```
{@Percent} < 5
```

use the group selection formula:

```
Sum({orders.ORDER AMOUNT}, {customer.REGION})
      % Sum({orders.ORDER AMOUNT}) < 5
```

Now when you print, only the regions that contributed less than 5% will print.

Grouping 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 that can be created. In this example, you will rank customers by the amount of business they did in the previous year.

This example uses 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.

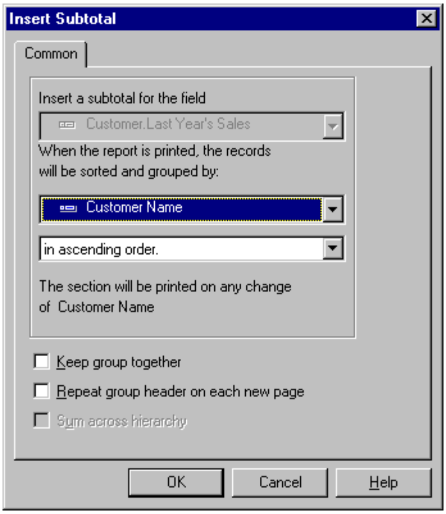
To group data in intervals

1 Create a report using the sample data, **Xtreme.mdb**, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}
{customer.REGION}
{customer.POSTAL CODE}
{customer.COUNTRY}
{customer.LAST YEAR'S SALES}
```

2 Right-click the Last Year's Sales data field, point to **Insert** and choose **Subtotal**.

The Insert Subtotal dialog box appears.



Insert Subtotal

Common

Insert a subtotal for the field: Customer.Last Year's Sales

When the report is printed, the records will be sorted and grouped by: Customer Name

in ascending order.

The section will be printed on any change of Customer Name

☐ Keep group together

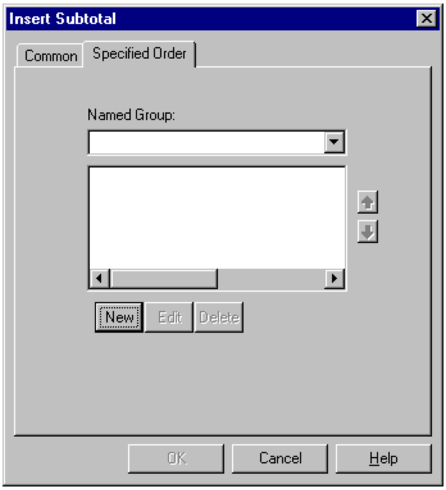
☐ Repeat group header on each new page

☐ Sum across hierarchy

3 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 from this drop-down list.

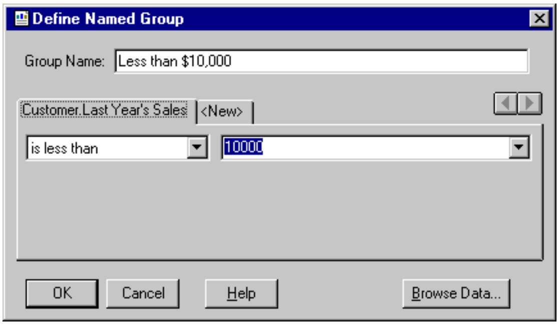
4 Select **in specified order** from the second drop-down list.

The Specified Order tab appears in the Insert Subtotal dialog box.



5 Click the **New** button.

The Define Named Group dialog box appears.


Group Name: Less than $10,000

Customer.Last Year's Sales

is less than 10000

6 Type “Less than $10,000” in the **Group Name** field.

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 of less than $10,000, set the fields so your condition reads:

```
is less than 10000
```

8 Click **OK** to return to the Specified Order Tab.

9 Click **New**.

The Define Named Group dialog box reappears.

10 Set up a second group that contains values between $10,000 and $25,000.

- Type “$10,000 to $25,000” in the **Group Name** field.
- Set the first field so the condition reads: `is between`
- Specify a range of values:
    - type “10000” in the top field
    - type “25000” in the bottom field.

You have now set up the group to contain all values between $10,000 and $25,000.

11 Click **OK** to return to the Specified Order tab.

12 Click **New**.

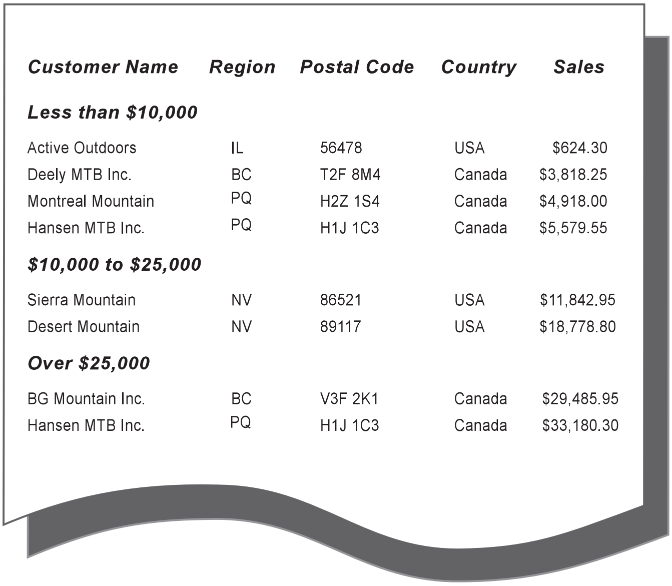
The Define Named Group dialog box reappears.

13 Set up the final group that contains all values over $25,000.

- Type “Over $25,000” in the **Group Name** field.
- Set the first field so the condition reads: `is greater than`
- Type “25000”.

14 Click **OK** to return to the Specified Order Tab.

15 Click **OK**. The report is grouped by interval in a specified order. Your report should look similar to the following 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

Grouping by the first letter of a company name

You might want to break your data into groups based on the first letter of the company name. For example, in a customer list you might want all the "A" customers in a group, then all the "B" customers, and so forth. To do this requires the use of a formula.

Do not worry if you are unfamiliar with formulas. This text will show you what formula you need here and how to enter it.

You can learn more about creating and editing formulas in "Formulas overview" on page 382.

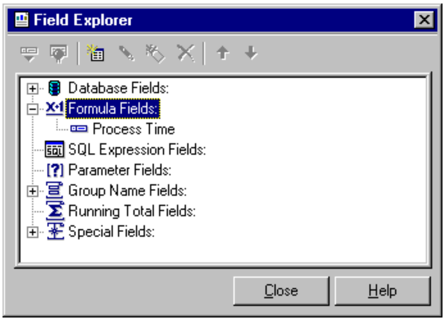
You are going to create a formula that will extract the first letter of each customer's name. Then you will group the data using that formula field as the sort-and-group-by field. The program will sort the data based on the first letter of each customer name and start a new group whenever that letter changes.

To group data by the first letter of a company name

1 Create a report using the sample data, **Xtreme.mdb**, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}
{customer.REGION}
{customer.POSTAL CODE}
{customer.COUNTRY}
```

2 On the **Insert** menu, click **Field Object**.
The Field Explorer dialog box appears.



3 Select **Formula Fields** and click **New**.

4 In the Formula Name dialog box, enter the name you want to identify the formula, for example "First Letter."

5 Click **OK**.

The Formula Editor appears.

6 Choose **Crystal Syntax** from the drop-down list in the top right corner.

7 Type the following formula into the Formula text box:

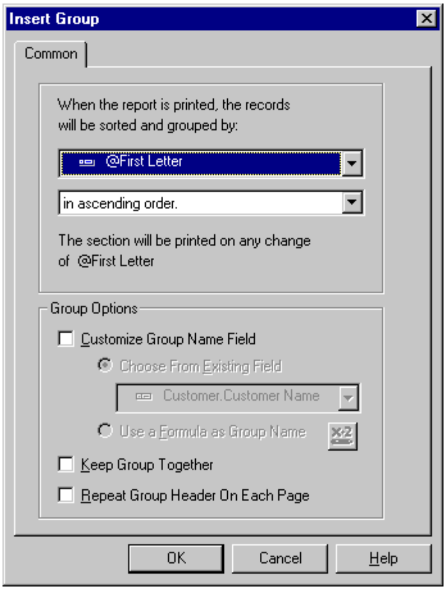
```
{Customer.Customer Name}[1]
```

8 Click the **Save and Close** button.

You return to the Field Explorer dialog box. The name of the formula is listed in the Field Objects directory.

9 On the Insert menu, click **Group**.

The Insert Group dialog box appears.



Insert Group

Common

When the report is printed, the records will be sorted and grouped by:

@First Letter

in ascending order.

The section will be printed on any change of @First Letter

Group Options

□ Customize Group Name Field

● Choose From Existing Field

Customer.Customer Name

○ Use a Formula as Group Name

□ Keep Group Together

□ Repeat Group Header On Each Page

10 Select the formula field as the field you want your data grouped by from the first drop-down list.

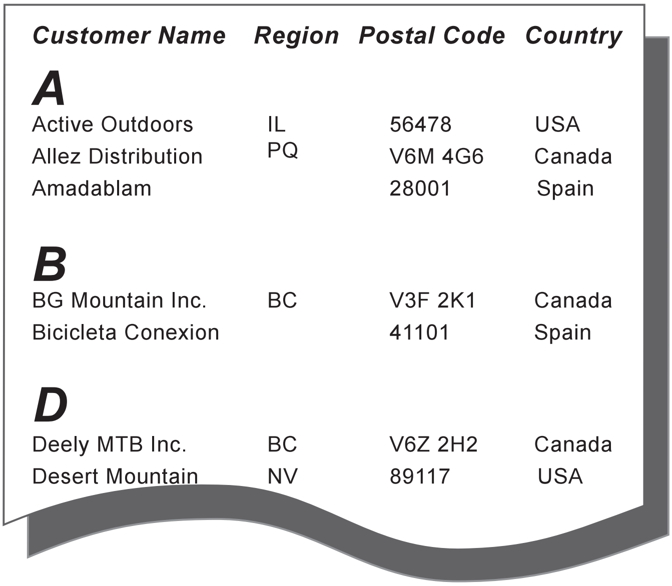
11 Select the sort direction from the second drop-down list.

12 Click **OK**.

You return to your report with the data grouped by the formula field as specified. The data is broken into groups based on the first letter in the customer's name. The formula provides a live header for every group.

For more information on live headers, see "Live headers" on page 166.

Your report should look similar to the following:



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

Related topics

"Formulas overview" on page 382.

"Grouping data" on page 140.

"Grouping data in intervals" on page 148.

"Creating group headers" on page 165.

Grouping data hierarchically

You have the option to group data in a report to show hierarchical relationships. When you group data hierarchically, you sort information based on the relationship between two fields.

For example, if you want to show the hierarchical structure of a department, you could group data by employee ID and specify the hierarchy using the data field listing whom the employee reports to.

To group data hierarchically

1. On the **Insert** menu, click **Group**.
2. In the Insert Group dialog box, select the field to be the basis of your hierarchy.

For example, if you want to view the hierarchical structure of a company's employee's, select the employee ID data field. If you want to view the hierarchical structure of region sales offices, select office names.

3 Select **in ascending order**.

4 By default, the group header of the report will display the value of the field you are grouping on. Select the Customize Group Name field check box if you want to show a different value in the group header.

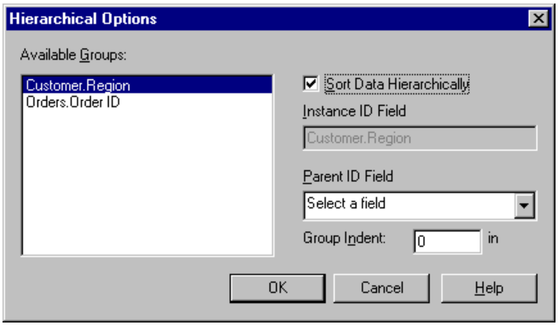
For example, if you grouped by {Employee.EmployeeID}, at each change of a group you will see the corresponding employee ID. If you want to display a different value (employee name instead of employee ID), customize the group name field by choosing an alternate data field, or creating a formula.

5 Click **OK**.

The group you created is added to the report.

6 On the **Report** menu, click **Hierarchical Grouping Options**.

The Hierarchical Options dialog box appears.



Hierarchical Options

Available Groups:

Customer.Region
Orders.Order ID

☑ Sort Data Hierarchically

Instance ID Field: Customer.Region

Parent ID Field: Select a field

Group Indent: 0 in

7 In the **Available Groups** list, select the group you want to organize hierarchically.

If you've created only one group on your report, then it will be selected automatically in the Available Groups list.

8 Select the **Sort Data Hierarchically** check box.

9 In the Parent ID field list, select the field by which you want the Instance ID field organized.

For instance, for a company hierarchical report, you might select the data field listing the supervisor to whom the employee reports.

Note: The Instance ID and Parent ID fields must be of the same data type. For example, if the Instance ID field holds string data, then the Parent ID field must also hold string data.

10 In the Group Indent field, enter the amount you want to indent for each subgroup.

11 Click **OK**.

The report data is now grouped hierarchically. If necessary, you can now calculate summary fields across your new hierarchical grouping. When inserting a Subtotal, Grand Total, or Summary in the usual manner, select the “Sum across hierarchy” option. For more information, see “Summarizing grouped data” on page 156 and “Subtotaling” on page 160.

Tip: If you want to suppress the group header, select the group header. On the Format menu, click Paragraph formatting. Click the Suppress check box.

Note:

- The top level of the hierarchy is determined by group instances that match the Instance ID and Parent ID.
- If a group instance is not connected to any Parent ID, it appears at the top of the hierarchy.

Editing groups

To edit a group

1 On the **Report** menu, click **Change Group Expert**.

2 In the Change Group dialog box, select the group you want to edit.

3 Click **Options**.

4 In the Change Group Options dialog box, edit the group as necessary.

5 Click **OK**.
The report reflects the changes you have made to the group.

Summarizing grouped data

One of the primary purposes for breaking data into groups is to run calculations on each group of records instead of on all the records in the report.

When the program summarizes data, it sorts the data, breaks it into groups, and then summarizes the values in each group. It does this all automatically.

The program includes a number of summarizing options. Depending on the data type of the field you plan 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, average, or Nth largest value
- calculate up to two kinds of standard deviations and variances.

For example:

- Customer list reports: determine the number of customers in each state. The summary would count the distinct customers in each state group.
- Order reports: determine the average order placed each month. The summary would calculate the size of the average order for each month group.
- Sales reports: determine the total sales per sales representative. The summary would sum or subtotal the order amounts for each sales representative group.

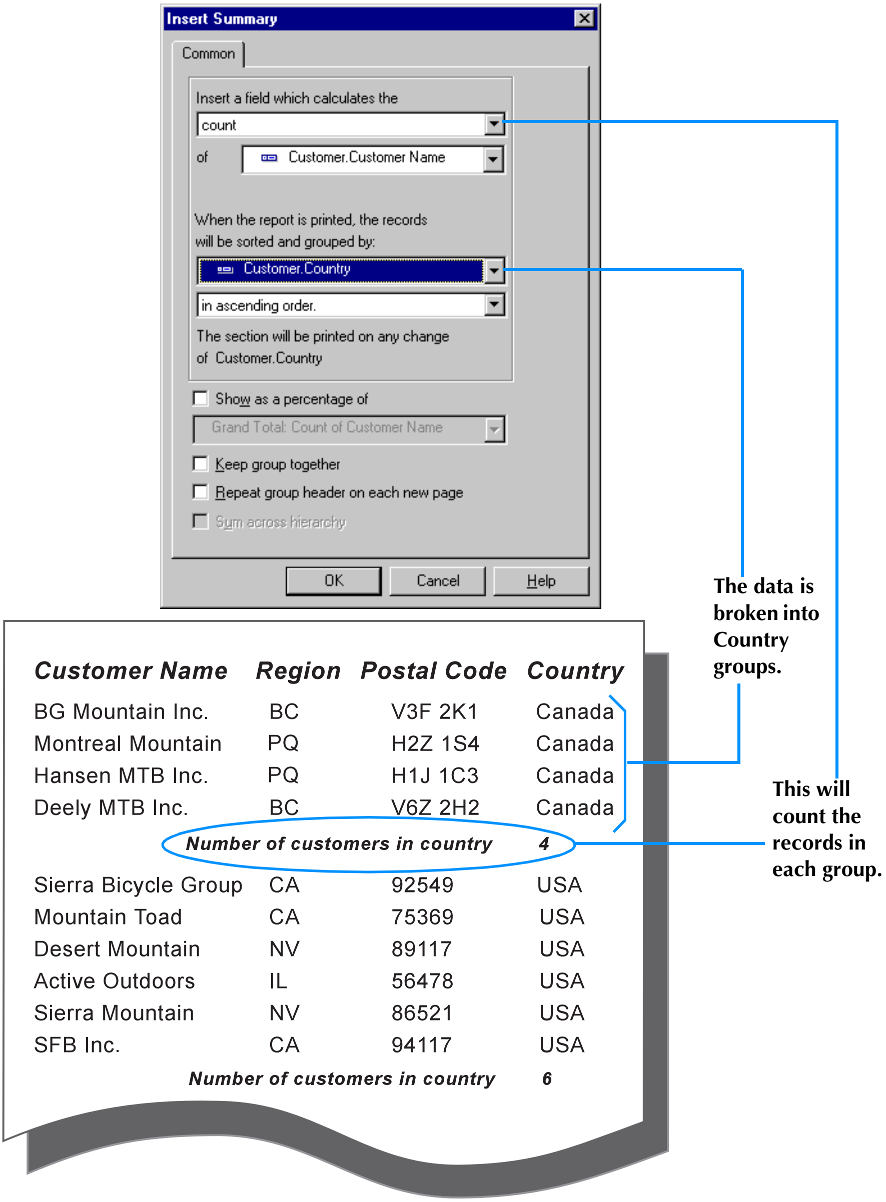
Note: You can also calculate summary fields across hierarchical groupings. To do so, select “Sum across hierarchy” in the Insert Subtotal, Insert Grand Total, or Insert Summary dialog box.

To summarize grouped data

1. On the Insert menu, click **Summary**.

   The Insert Summary dialog box appears.
2. Select the desired summary operation from the top drop-down list.
3. Select the field you want the data to be grouped by from the second drop-down list.
4. Select the sort direction from the third drop-down list.
5. Click **OK** when finished.

   For example, if you wanted to count the number of customers you had per country, you would create a count field based on customer, and group by country:



Insert a field which calculates the: count

of: Customer.Customer Name

When the report is printed, the records will be sorted and grouped by: Customer.Country

in ascending order.

The section will be printed on any change of Customer.Country

□ Show as a percentage of

Grand Total: Count of Customer Name

□ Keep group together

□ Repeat group header on each new page

□ Sum across hierarchy

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.

Related topics

"Subtotaling data" on page 160.

Ordering groups by summarized values

Groups can be organized in either ascending or descending order by summary value. For example, in an orders report, if you subtotal the order amount by state, you could order the groups:

- from lowest to highest order amount (ascending)
- from highest to lowest order amount (descending).

You order the groups in a report by summary values using the TopN/Sort Group Expert command on the Report Menu.

To order groups by summary value

1 Click **Top N Expert** on the Supplementary toolbar.

The Top N/Sort Group Expert dialog box appears with a tab for each of the groups in the report that has a summary.

2 Click the tab for the group you want to sort.

3 Select the **All** option from the drop-down list on the left.

4 Choose the summary that you want to base your selection on from the summary drop-down list on the right.

The summary drop-down list on the right is for those cases in which you have multiple summaries within a single group section. For example, in an orders report, you might sum and average the orders for each customer and then 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 list.

5 Specify the sort direction.

6 To select a second group sort, repeat Steps 2-5.

When you run the report, the program will order the groups based on the summary value(s) specified.

Selecting top or bottom N groups

At 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, etc. Because this kind of group selection is so popular, the program includes the Top N/ Sort Group Expert for setting it up easily.

There is one other element to consider when setting up Top N: what to do with all the records from other groups that do not fit the Top N or Bottom N criteria you have set. You need to decide whether to eliminate those records from your report entirely or to lump them all together into a single group. The program enables you to do choose either option.

Note: A report must contain a summary value in order to be able to perform a top N or bottom N selection. See “Summarizing grouped data” on page 156.

To select the top or bottom N groups

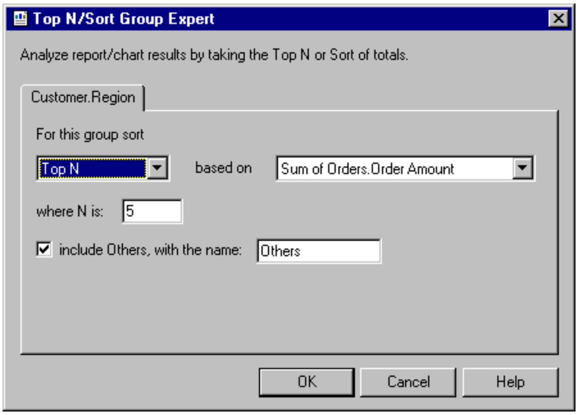
1 Create a report and summarize the data as desired. When you summarize the data, the program breaks the data into groups and summarizes each group.

- with top N grouping, you are instructing the program to display those groups that have the highest summary values (Top N)
- with bottom N grouping, you are instructing the program to display those groups that have the lowest summary values (Bottom N).

2 Click the **Top N Expert** button on the Supplementary toolbar.

The Top N/Sort Group Expert appears with a tab for your group.

Note: If you have multiple groups, the program will display a tab for each of the groups.


Top N/Sort Group Expert

Analyze report/chart results by taking the Top N or Sort of totals.

Customer.Region

For this group sort: Top N based on: Sum of Orders.Order Amount

where N is: 5

☑ include Others, with the name: Others

3 Choose **Top N**, **Bottom N**, or **All** from the first drop-down list.

4 Choose the summary that you want to base your selection on from the summary drop-down list on the right.

The summary drop-down list on the right is for those cases in which you have multiple summaries within a single group section. For example, in an orders report, you might sum and average the orders for each customer and then 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 list.

5 In the **where N is** text box, enter the number of groups you want to display. For example:

- to report on the three fastest selling product lines, select the top N option in the Top N Sort Group Expert and set *N* to be equal to three
- to report on the five least productive sales regions, select the bottom N option in the Top N Sort Group Expert and set *N* to be equal to five.

6 Click the **include Others, with the name** check box and enter a name if you want to group all the other records into a single group.

7 Click **OK** when finished.

When the program runs the report, it will include only those groups that you specified.

Subtotaling

A subtotal is a summary that totals or sums numeric values in a group. Crystal Reports groups and subtotals in a single step.

Note: If you are creating a subtotal using database tables that are grouped in a one-to-many linking relationship, you may need to use a running total instead of a subtotal. See "Creating running totals in a one-to-many linking relationship" on page 177.

Subtotaling data

In this example you will subtotal Last Year's Sales by Country.

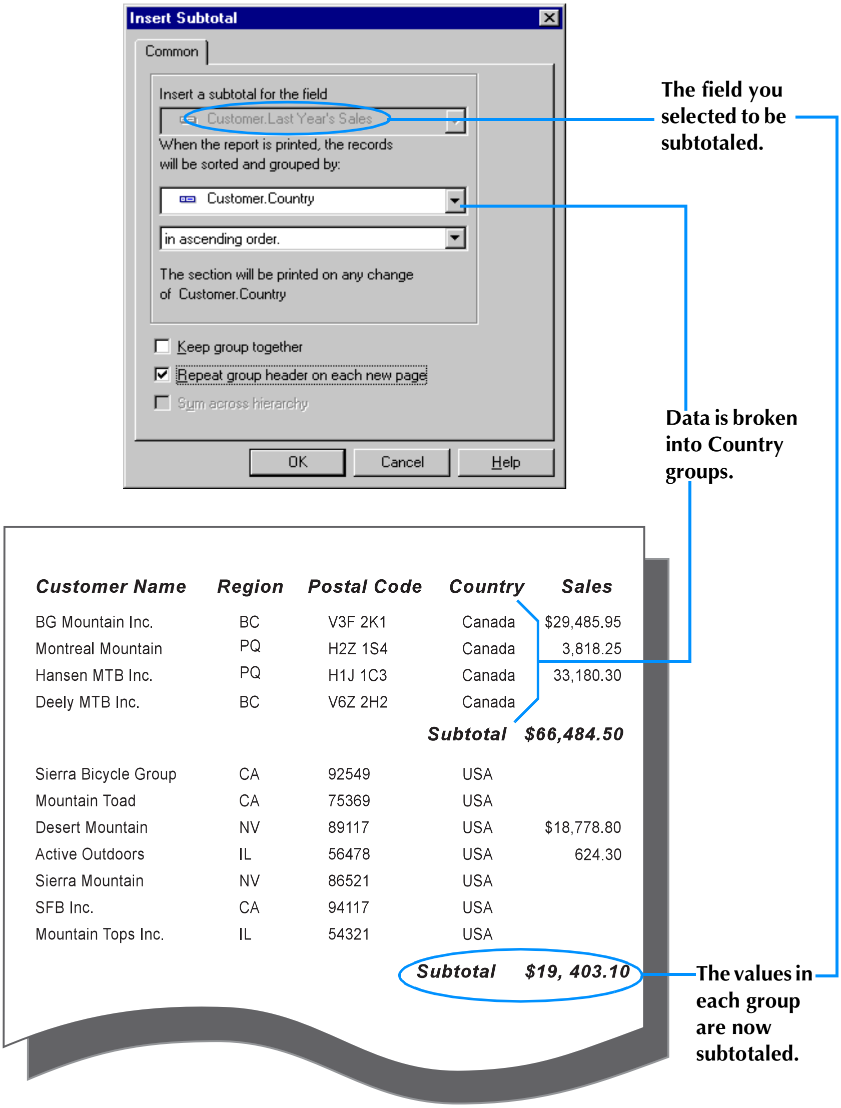
To subtotal data

1 Create a report using the sample data, **Xtreme.mdb**, and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}
{customer.REGION}
{customer.POSTAL CODE}
{customer.COUNTRY}
{customer.LAST YEAR'S SALES}
```

2 Right-click the Last Year's Sales field, point to **Insert** and choose **Subtotal** from the shortcut menu.

The Insert Subtotal dialog box appears with the chosen field listed at the top of the dialog box.



The field you selected to be subtotaled.

Data is broken into Country groups.

The values in each group are now subtotaled.

Insert Subtotal

Common

Insert a subtotal for the field

Customer.Last Year's Sales

When the report is printed, the records will be sorted and grouped by:

Customer.Country

in ascending order.

The section will be printed on any change of Customer.Country

□ Keep group together

☑ Repeat group header on each new page

□ Sum across hierarchy

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

3 Choose the field you want the data grouped by, specify a sort direction and then click **OK** when finished. The values in each group are now subtotaled.

Extending prices and subtotaling 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 by using a simple formula to extend the prices, and then you subtotal the formula field.

To extend the price and subtotal the extensions

1. Create a report using the sample data, **Xtreme.mdb**, and place the following fields from left to right in the Details section:

   ```
   {orders.CUSTOMER ID}
   {orders detail.PRODUCT ID}
   {orders detail.QUANTITY}
   {orders detail.UNIT PRICE}
   ```

2. To create the formula for extending the prices, choose **Insert** from the main menu and select **Field Object**.

   The Field Explorer dialog box appears.

3. Select **Formula Fields** and click **New**.

   The Formula Name dialog box appears.

4. Enter the name you want to identify the formula.

5. Click **OK**.

   The Formula Editor appears.

6. Enter the following formula into the Formula text box:

   ```
   {Orders_Detail.Quantity} * {orders detail.Unit Price}
   ```

7. Click the **Save and Close** button when finished.

   You return to the Field Explorer dialog box with the name of your formula listed in the Formula Fields directory.

8. Double-click your formula and place the formula field to the right of the Unit Price field in the Details section of the report.

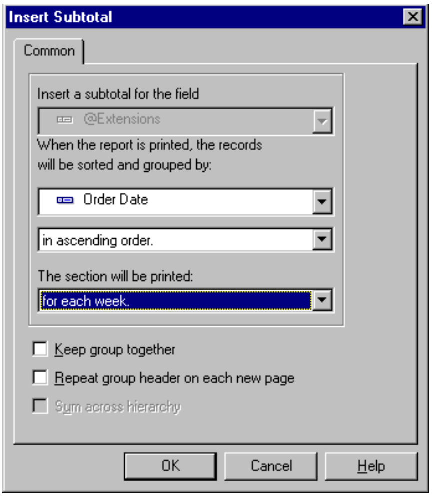
9. Click **Close** to close the Field Explorer dialog box.

10. To subtotal your extension formula, right-click the formula field, point to **Insert** and choose **Subtotal** from the shortcut menu.

    The Insert Subtotal dialog box appears.

11. To create a subtotal for weekly sales, select {orders.ORDER DATE} as the sort-and-group-by field, and choose **for each week** as the group interval.

Note: The "section will be printed" drop-down box will not be activated until you choose the Order Date field.



Insert a subtotal for the field: @Extensions

When the report is printed, the records will be sorted and grouped by: Order Date

in ascending order.

The section will be printed: for each week.

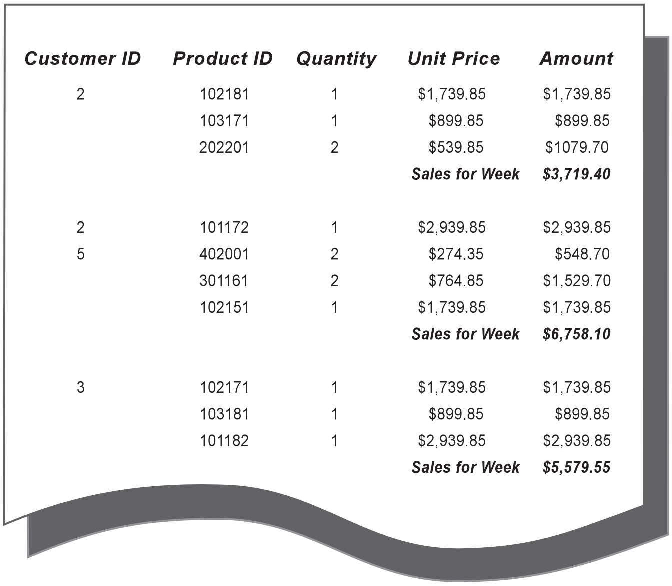
□ Keep group together

□ Repeat group header on each new page

□ Sum across hierarchy

12 Click **OK** to insert the subtotal in the report.

Your data will be sorted by date and grouped in one week intervals. Your final report should look similar to the following:



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***

Percentages

Calculating a percentage

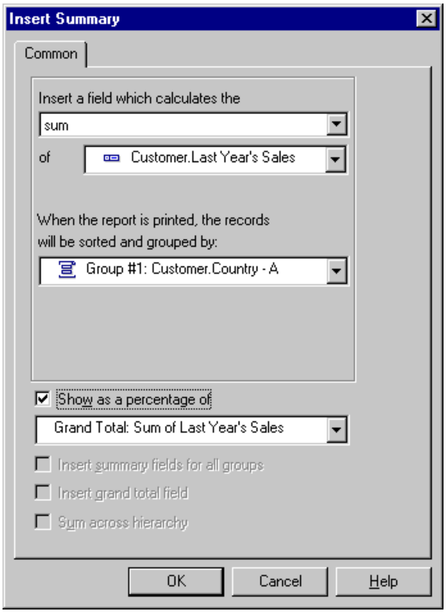
You can calculate the percentage of one group within a broader grouping. For example, you can show the percentage of sales in each city based on the total sales for each country. Or, you can see what percentage of the grand total of sales each country contributes.

To calculate a percentage

1 On the **Insert** menu, click **Summary**.

Tip: You can also add percentage fields on the Total tab in the Create Report Expert.

The Insert Summary dialog box appears.



Insert a field which calculates the: sum

of: Customer.Last Year's Sales

When the report is printed, the records will be sorted and grouped by: Group #1: Customer.Country - A

☑ Show as a percentage of: Grand Total: Sum of Last Year's Sales

□ Insert summary fields for all groups

□ Insert grand total field

□ Sum across hierarchy

2 Enter **sum** in the first drop-down list.

3 Select the field for which you want to calculate the sum.

For example, you may want to insert a field which calculates the sum of last year's sales.

4 Select how you want the data grouped.

5 Click the **Show as a percentage of** check box.

6 Select the group you want the percentage based on.

You can choose to show a percentage of a group within another group, or show a percentage of the grand total.

7 Click **OK**. The summary percentage field is added to your report.

Group headers

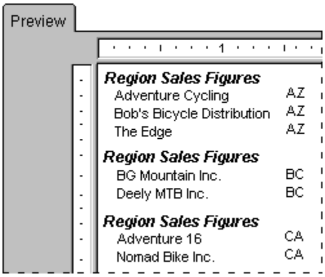
Creating group headers

Whenever you create a group, a subtotal, or a summary, the program creates both a Group Footer (GF) section (where it places any subtotal or summary value), and a Group Header (GH) section (where it automatically places the group name/header). Group Headers are useful, even necessary, if you want your report data to be clear and easily understood. Though the program creates a group header automatically, you may find that you would like to change or modify the header to suit your needs. You will learn how to create the four most common kinds of group headers:

- "Standard headers" on page 165.
- "Live headers" on page 166.
- "Live headers for groups based on a formula" on page 167.
- "Headers for custom groups" on page 168.

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.



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" means it is a regional group), you never know what region is in the group without first looking at the details in the group.

To create a standard header

1. Click the **Insert Text Object** button on the Standard toolbar.
2. When the object pointer appears, move the object frame into the Group Header section.
3. Enter the text you want to use for the 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.

Live headers

A live header is a header that changes based on the content of the group. If the data is 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 group as Arizona data, the California group would have a header identifying the group as California data, and so on.

Note: When creating 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 to manually insert such a section (if you do not 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 one based on identifying the value of the group field.

To create a live header by group name only

1. On the **Insert** menu, click **Field Object**.
2. In the Field Explorer dialog box, expand the **Group Name Fields** folder.
3. Double-click the Group Name field that matches the group you are working with and drag it into the Group Header section for that group.

   When you run the report, the group field value identifier will appear as the group header for each region group.

Group name with text

A more complex type of live header combines a field value and text. A typical group header of this kind for data broken down by region would be, “Sales for California” or “Customers in Postal Code 60606“. Creating these headers involves three steps:

- Insert a text object in the Group Header section.
- Type in the text you want to appear.
- Enter the Group Name field in the text field where you want it to appear in the Group 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:

To create a live header with group name using text

1 Click the **Insert Text Object** button on the Standard toolbar.

2 Place the object frame into the Group Header section for the group.

3 Type in the desired text with a space after it.

4 Click the **Insert Fields** button on the Standard toolbar.

5 In the Field Explorer dialog box, expand the **Group Name Fields** folder.

6 Double-click the Group Name field that matches the group you are working with and drag it into the text object, immediately after the text and the space you entered.

Tip: Expand the size of the text box to fit both the text and the group field.

7 Format the text object as you want it to appear.

When you run the report, the program will create a live header (with text) for each of your groups.

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 will have the letter “B”, and so on. For more information, see “Grouping by the first letter of a company name” on page 151 and “Grouping data in intervals” on page 148.

To create a more descriptive header such as “Customers beginning with the letter B”, see “Group name with text” on page 166.

Headers for custom groups

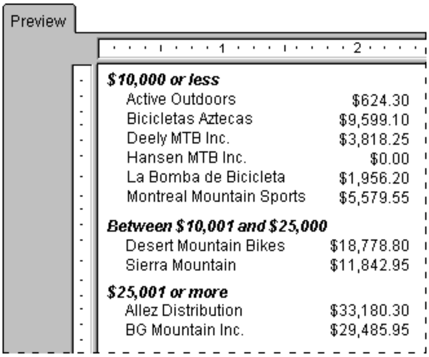
The final type of header is a header for the types of custom groups created when data is grouped in a specified order. When using specified order grouping, both the name for each group and the records that belong in it are specified. As in the other grouping situations, the program creates a group name field for each group based on the group names specified.

To create a header for custom groups

1. On the **Insert** menu, click **Field Object**.
2. In the Field Explorer dialog box, expand the **Group Name Fields** folder.
3. Double-click the Group Name field for the custom group and drag it into the Group Header section for that group.

The program automatically applies each of the group names you assigned to the appropriate groups.

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.



$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

Suppressing group headers

You have the option to hide group headers in your report.

To suppress group headers

1. Right-click the group header and select **Format Field**.
2. In the Format Editor, on the Common tab, click the **Suppress** check box.
3. Click **OK**.

   Tip: To show the group header again, clear the Suppress check box.

Drilling-down on group headers

To make report viewing easier, you can hide the details of your report and only have the group headers visible. When necessary, you can click on the group header to view the report details.

To drill-down on group headers

1. Right-click the grey Details section to the left of the report.
2. Select **Hide (Drill-Down OK)**.
3. Click **Refresh**.

   When you place the cursor over a group header, the cursor becomes a magnifying glass.
4. Double-click the group header to drill-down to the detail information.

Related topics

"Using the drill-down option on summarized data" on page 99.

Running Totals 10

Running totals are a flexible and powerful way to create specialized summaries and continually incrementing totals.

This chapter shows you how to add a basic running total and a running total within a group to your report. You will also learn how to create conditional running totals and running totals using formulas.

Understanding running totals

Running total fields are similar to summary fields but allow more control over how the total is calculated and when it is reset. Running total fields are specifically suited to perform the following totaling functions:

- show values of a total accumulate as it is calculated record by record
- total a value independent of the report's grouping
- total a value conditionally
- total a value after a group selection formula has been applied
- total a value from the driving table in a one-to-many linking relationship.

How running totals work

A running total field is created with the Running Total Expert. The Running Total Expert creates a running total field by asking you to select a field to summarize, the summary operation to use, a condition upon which to base the evaluation, and a condition upon which to reset the evaluation.

Note: A running total field can be used on database fields and first-pass formulas, but cannot be used on second-pass formulas or formulas that reference other second-pass formulas. For more information see"Multi-pass reporting flow chart" on page 660.

Placement of running total fields

The calculation of a Running Total field is determined by the settings selected in the Running Total Expert. However, where you place the running total affects the value that appears on the report. For example, if a Running Total field that evaluates every record and never resets (a grand total) in the Report Header, only the value of the first record will appear. Placing the same Running Total field in the Report Footer returns the desired value. The Running Total field is properly calculated in both cases, but it is displayed too soon in the first case.

The following is a list of the report sections and the records that the running total will use.

Report Header	Returns first record of the report
Page Header	Creates running total up to and including the first record of current page.
Group Header	Creates running total up to and including first record of current group.
Details	Creates running total for every record.
Group Footer	Creates a grand total for each group.
Page Footer	Creates running total up to and including the first record of the next page.
Report Footer	Creates a grand total that includes all records.

Creating running totals

Creating 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 on) up to and including the current record.

The most basic form of a running total is a single running total maintained throughout a list. In this tutorial, you will create this kind of report by setting up a running total for a list of order amounts.

Note: Running total fields are prefixed by the # sign.

To create a running total in a list

1 To get started, create a report using the sample database, **Xtreme.mdb**. Link the Customer and Orders tables, and then place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}
{orders.ORDER ID}
{orders.ORDER AMOUNT}
```

2 On the **Insert** menu, click **Field Object**.

The Field Explorer dialog box appears.

3 Select **Running Total Fields** and click **New**.

The Create Running Total Field dialog box appears.

4 Enter the name “TotalOrders” in the **Running Total Name** box.

5 Highlight {orders.ORDER AMOUNT} in the **Available Tables and Fields** box, and use the first arrow button to move it over to the **Field to summarize** box.

6 Select **sum** from the **Type of summary** list.

7 In the **Evaluate** section of the dialog box, click **On change of field**, and select {orders.ORDER ID} as the **On change of** field.

The running total will execute each time this field changes.

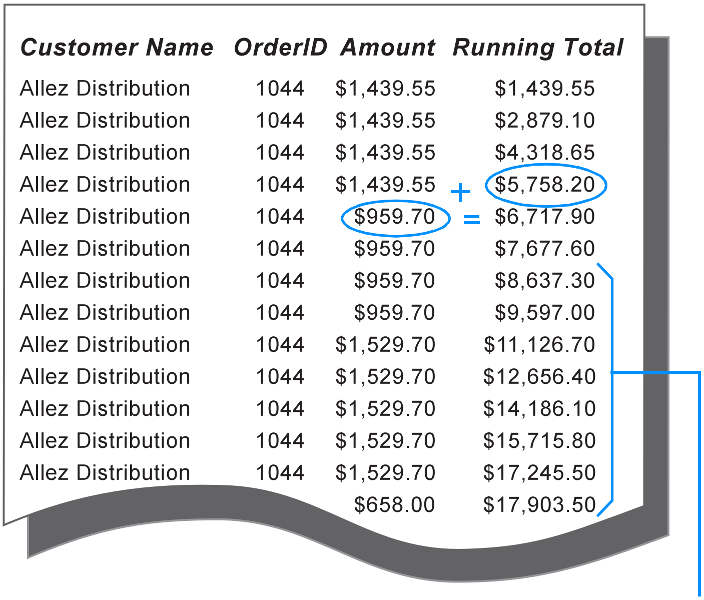
8 In the **Reset** section of the dialog box, click **Never** (this gives you a running total that never resets; that is, the running total continues throughout the report).

9 Click **OK** to save the running total field.

The program returns you to the Field Explorer dialog box.

10 Insert the running total field in the Details section of the report, just to the right of {orders.ORDER AMOUNT}.

The value in each record is added to the sum of the previous value in the report.



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 total continues, unbroken, throughout the running list.

Creating running totals for a group

Another common use for running totals is 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 tutorial, you will create a report that:

- maintains a running total of customer orders
- groups customer orders and resets the running total for each group
- displays the subtotal for each order (the last running total for that order).

To create a running total for a group

1 To get started, create a report using the sample data, **Xtreme.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. On the **Insert** menu, click **Group** and group on the {customer.CUSTOMER NAME} field.
3. On the **Insert** menu, click **Field Object**.

   The Field Explorer dialog box appears.
4. Select **Running Total Fields** and click **New**.

   The Create Running Total Field dialog box appears.
5. Enter the name “GroupRunningTotal” in the **Running Total Name** box.
6. Highlight {orders.ORDER AMOUNT} in the **Available Tables and Fields** box, and use the first arrow button to move it over to the **Field to summarize** box.
7. Select **sum** from the **Type of summary** list.
8. In the **Evaluate** section of the dialog box, click **For each record**.
9. In the **Reset** section of the dialog box, click **On change of group** and accept the default group name.
10. Click **OK** to save the running total field.

    You return to the Field Explorer dialog box.
11. Place the running total field in the Details section of your report, just to the right of the {orders.ORDER AMOUNT} field.

    Note: If you want to view a grand total of each group, place the running total field in the Group Footer section of your report.

Creating conditional running totals

There may be times when you have a list of ungrouped 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 U.S. 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
- you want a total of the values from just the Canadian records
- you also want a total of the values from just the U.S. records.

To accomplish this, create two running totals, one to keep a running total of the U.S. records, and one to keep a running total of the Canadian records.

- **USTotal**
  maintains a running total of the U.S. records
- **CanadaTotal**
  maintains a running total of the Canadian records.

To create a conditional running total

1 To get started, create a report using the sample data, **Xtreme.mdb**. Place the following fields from left to right in the Details section:

```
{customer.CUSTOMER NAME}
{customer.COUNTRY}
{customer.LAST YEAR'S SALES}
```

2 On the **Report** menu, click **Sort Records**.

Tip: Another way to do this is to click the Sort button on the Standard toolbar.

3 Sort the records based on the {customer.CUSTOMER NAME} field.

4 On the **Insert** menu, click **Field Object**.

The Field Explorer dialog box appears.

5 Select **Running Total Fields** and click **New**.

The Create Running Total Field dialog box appears.

6 Enter the name “USTotal” in the **Running Total Name** box.

7 Highlight {Customer.LAST YEAR’S SALES} in the **Available Tables and Fields** box, and use the first arrow button to move it over to the **Field to summarize** box.

8 Select **sum** from the **Type of summary** list.

9 In the **Evaluate** section of the dialog box, click **Use a formula** and then click the **Formula** button.

The Running Total Condition Formula Editor appears.

10 Enter the following formula in the formula box:
{Customer.Country} = "USA"

This tells the program to evaluate the running total each time it comes to a record where {Customer.COUNTRY} is equal to “USA.” The running total will ignore all other records (such as records for Canada).

11 When the formula has the correct syntax, click **Save and Close**.

You return to the Create Running Total Field dialog box.

12 In the **Reset** section of the dialog box, click **Never**.

13 Click **OK** to save the running total field.

The program returns you to the Field Explorer dialog box.

14 Place the running total field in the Details section of your report.

15 Now create the “CanadaTotal” running total field using the process outlined in steps 3-13. The only difference is that this time you will set the Evaluate formula to:

```
{Customer.Country} = "Canada"
```

16 When you are finished, place the **#CanadaTotal** field in the Details section of your report, just to the right of the {Customer.LAST YEAR'S SALES} field.

17 If you only want to see a grand total of the Canadian and American sales, place the two running total fields you created in the Report Footer section of your report.

Creating running totals in a one-to-many linking relationship

The term A to B, A to C report is used to refer to any report in which a primary table is linked to two lookup tables (see "Methods of looking up tables (direct access databases)" on page 524). 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 relationship. The Customer table is linked to the Credit table and again to the Orders table. Notice that Jones has only one Credit ID, but the credit and its amount are repeated, once for each of Jones' two orders.

Name	Credit ID	Credit Amount	Order ID	Order Amount
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

In this example, a standard subtotal is used for both the Order Amount field and the Credit Amount field. Thus, Jones' single credit is counted twice, and the subtotal displays an inaccurate value of -20.00.

Name	Credit ID	Credit Amount	Order ID	Order Amount
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 total Credit Amount for Jones is incorrect.

This problem also occurs in the Order Amount field if Jones had two different Credit Amounts and only one Order Amount. Avoid this problem by creating a running total for each field you want subtotaled.

Creating running totals using a formula

If you have suppressed data, or your data is based on a formula that occurs WhilePrintingRecords, you should create a running total formula rather than using the Running Total Select Expert.

When you create a running total manually, you need to create three formulas:

- a summary formula
- a reset formula to set a variable to zero
- a display formula to display the variable.

In the following procedure, you will create a report that performs the following functions:

- maintains a running total of customer orders
- groups customer orders and resets the running total for each group
- displays the subtotal for each order (the last running total for that order).

To create running totals using a formula

1. Create a report using the sample data, **Xtreme.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. On the **Insert** menu, click **Field Object.**
3. Select **Formula Fields** in the Field Explorer dialog box and click **New**.
4. Name the formula “RunningTotal” and click **OK**.
5. Enter the following into the Formula Editor:

```
WhilePrintingRecords;
CurrencyVar Amount;
Amount := Amount + {orders.ORDER AMOUNT};
```

6. Click **Save and Close**.
7. 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.

8 On the **Insert** menu, click **Group** and group the report on the {customer.CUSTOMER NAME} field.

9 In the Formula Editor, create "AmountReset":

```
WhilePrintingRecords;
CurrencyVar Amount := 0;
```

This formula says:

Set the value in the Amount variable to 0.

10 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.

11 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.

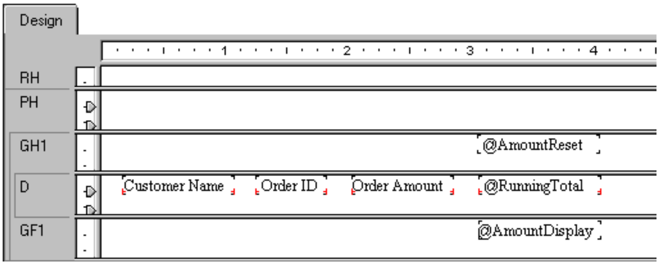
12 In the Formula Editor, create "AmountDisplay":

```
WhilePrintingRecords;
CurrencyVar Amount;
```

This formula simply displays the current value of the Amount variable at any time.

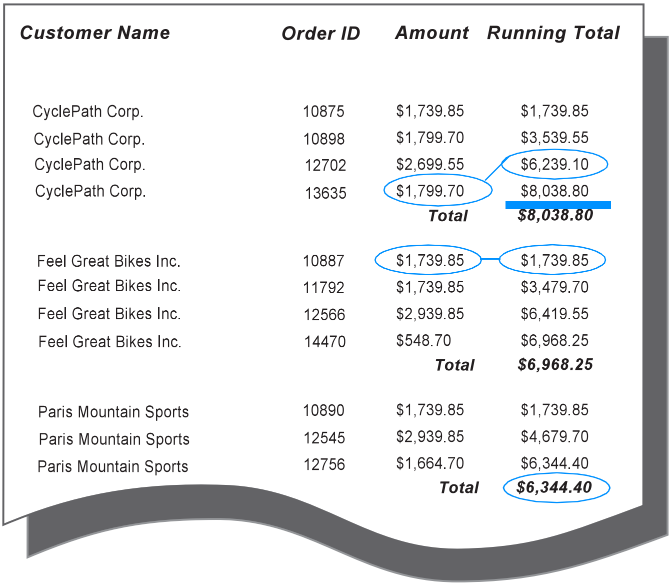
13 Place this formula in the Group Footer #1 section of your report.

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.

Your results should look similar to the following report:



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	$2,699.55	$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***

This report displays the running total within each group.

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).

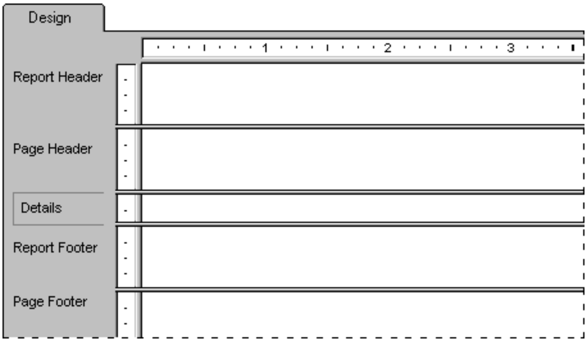
Multiple Section Reports 11

This chapter introduces the various types of sophisticated reports you can create using the multiple section reporting capabilities in Crystal Reports. These capabilities enable you to create reports that treat individual values differently based on sets of criteria you establish. These concepts are then applied to the creation of Form Letters.

About sections

Crystal Reports provides five design areas to use when building your report:

- Report Header
- Page Header
- Details
- Report Footer
- Page Footer.



Each area contains a single section when you first create a new report. You cannot 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.

Working with sections

Inserting a section

1 Click **Section Expert** on the Standard toolbar.

The Section Expert 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.

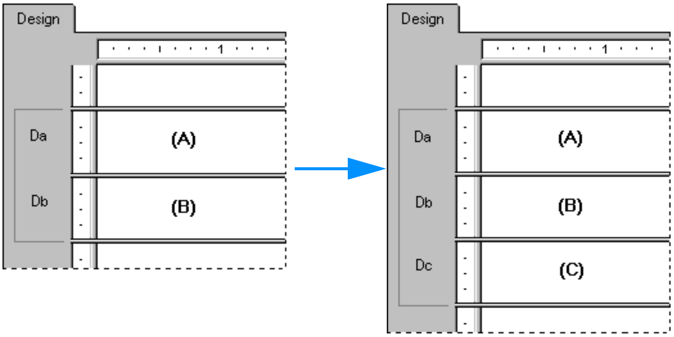
Note: The program enables only those options (free form, new page before, etc.) that apply to the highlighted section.

2 Highlight the section you want to insert a section after.

3 Click **Insert**.

A new section is inserted immediately below the highlighted section.

Note: Sections can also be inserted by right-clicking the shaded area to the left of any section in the Design or Preview tabs and choosing Insert Section Below from the shortcut menu.



Deleting a section

1. Click **Section Expert** on the Standard toolbar.

   The Section Expert 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: The program enables only those options (free form, new page before, etc.) that apply to the highlighted section.

2. Highlight the section you want to delete.
3. Click **Delete**.

   The program removes the highlighted section from the report.

Moving a section

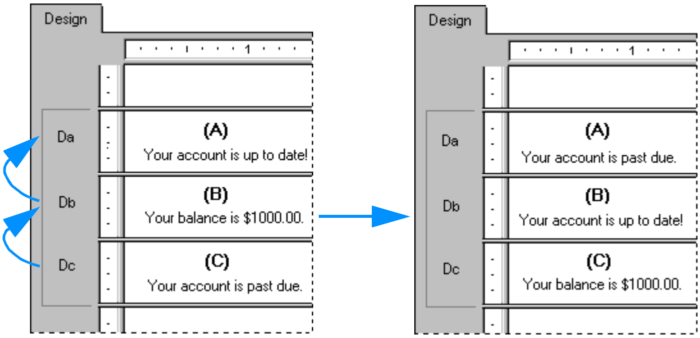
1. Click **Section Expert** on the Standard toolbar.

   The Section Expert 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: The program enables only those options (free form, new page before, etc.) that apply to the highlighted section.

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

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


Design
Da
Db
Dc
(A)
Your account is up to date!
(B)
Your balance is $1000.00.
(C)
Your account is past due.
Design
Da
Db
Dc
(A)
Your account is past due.
(B)
Your account is up to date!
(C)
Your balance is $1000.00.

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.
- 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.

Merging 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.

To merge related sections

1 Click **Section Expert** on the Standard toolbar.

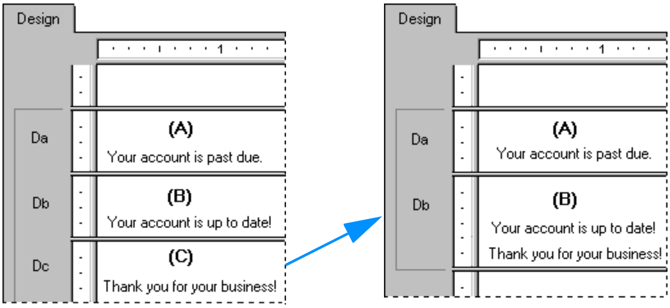
The Section Expert appears with a list of all the sections in the report. When there are more than one of any kind of section, the sections are lettered A, B, C, and so on.

Note: The program enables only those options (free form, new page before, etc.) that apply to the highlighted section.

2 Move the sections so the two sections you want to merge follow each other in the list.

3 Highlight the top section.

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



(A)
Your account is past due.
(B)
Your account is up to date!
(C)
Thank you for your business!

(A)
Your account is past due.
(B)
Your account is up to date!
Thank you for your business!

5 Rearrange the objects as needed.

Splitting and resizing sections

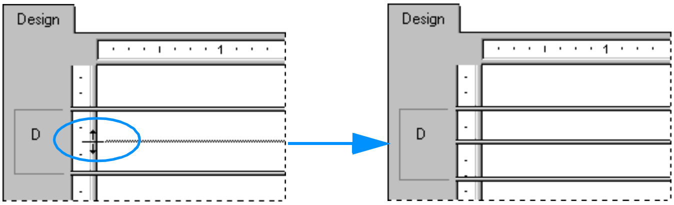
A section can be split into two or more sections and/or resized easily in the Design tab.

Splitting a section

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

2 When the pointer becomes a Section Splitting cursor, click the boundary.

3 When a horizontal line appears, drag it up or down to split the section the way you want it.

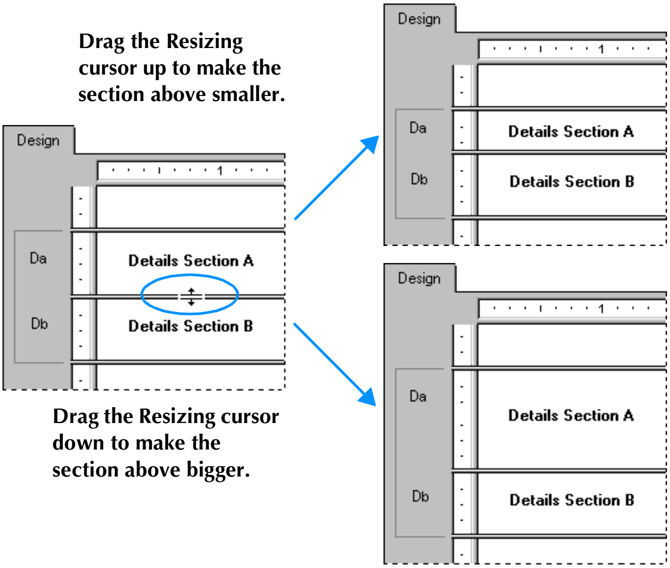


Resizing a section

1. Move the pointer over the bottom boundary of the section you want to resize.
2. When the pointer becomes a Resizing cursor, drag the boundary to make the section bigger or smaller as you wish.

Resizing a section to remove white space

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 in the Design and Preview tabs and choose Fit Section from the shortcut menu. 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 to make the section above smaller.

Drag the Resizing cursor down to make the section above bigger.

A section automatically expands vertically in two instances:

- when you place an object and the object is bigger (vertically) than the section you place it in
- when you expand an object (vertically) so it becomes bigger than the section it resides in.

Note: You cannot resize a section smaller than an object in the section.

Using multiple sections in reports

Certain reporting tasks are performed most efficiently by creating multiple sections within an area, such as:

- keeping variable length objects from overwriting each other (see “Keeping variable length objects from overwriting each other” on page 187)
- formatting objects conditionally (see “Formatting objects conditionally” on page 188)
- alternating background colors for rows (see “Alternating background colors for rows” on page 189)
- eliminating blank lines when fields are empty (see “Eliminating blank lines when fields are empty” on page 189)
- adding blank lines under specific conditions (see “Adding blank lines conditionally” on page 191).

Once you understand the power of multiple sections, you will discover even more ways to produce the effects you want.

Related topics

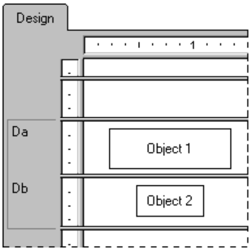
“Working with sections” on page 182.

Keeping variable length objects from overwriting each other

When subreports or other variable length objects are placed above other objects in one section of the report while the *Can Grow* option for the variable length object is toggled on in the Format Editor, that object may overprint objects positioned directly below it 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.

You can eliminate this overprinting problem by creating multiple sections in an area and placing objects below the variable length object in their own section(s).



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 “Combining two or more unrelated reports” on page 504.

Note: Memo and BLOB fields, as well as subreports, can cause overprinting.

Formatting objects conditionally

You may want to create a report that uses different formats depending on the field value. For example, you may want to create an international report that prints currency values for each country in the format that is common in that country. You can do that using multiple sections.

To format objects conditionally

1 Create a report. See “Creating the report” on page 40. Ensure you report contains the **Last Year’s Sales** field found in the **Customer** table.

2 Create one Details section for each country that requires a unique format. See “Working with sections” on page 182.

3 Make certain that each of the Details sections contains the same data; every field that you place in the Details A section, you must also place in the Details B section, and in the Details C section, etc.

4 In the Details A section, select the **Last Year’s Sales** field and then click **Format** on the Supplementary toolbar.

The Format Editor appears showing the Number tab.

5 Set the currency style you want to use for the first country.

6 Click **OK** to return to the report.

7 Click **Section Expert** on the Standard toolbar.

The Section Expert appears.

8 Select the Details A section in the **Sections** box. Select the **Suppress (No Drill-Down)** check box.

9 Click the **Conditional Formula** button to the right of the **Suppress (No Drill-Down)** check box and use the Format Formula Editor to create a formula that specifies the conditions under which the section should be suppressed.

For example, if the Details A section contains currency values for the UK, you would create a formula that specifies that the country value is not equal to UK. In other words, you create a formula that suppress0

es the section when the country value is not UK. This will make the section print only when the record contained a UK value.

10 Repeat Steps 4 through 9 for each additional section.

Now, when the report is printed, the date and currency data for each country appears in the format that is expected for that country. See “Working with conditional formatting” on page 237.

Alternating background colors for rows

Another typical use of multiple sections is to vary the background color in alternating lines for the Details section of a report in order to improve readability (a greenbar-paper effect).

To alternate background colors for rows

1. Use the Section Expert to insert a second Details section. You should now have Details A and Details B sections. See “Working with sections” on page 182.
2. While in the Section Expert, highlight Details A in the **Sections** list, click the **Color** tab, and set the background color to White.
3. Click the **Common** tab, click the **Conditional Formula** button to the right of **Suppress (No Drill-Down)**, and type the following formula in the Format 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. Click **Save and Close**.
5. Now select Details B and set the background color to Green.
6. Use the technique from Step 3 to set **Suppress (No Drill-Down)** conditionally for this section using the following formula:

   ```
   Remainder(RecordNumber,2) = 0
   ```

   This formula divides the record number by 2 and if the remainder is equal to zero, it tells the program to suppress the section.
7. Create the report and make certain that the information and layout of each of the Details sections is identical. In other words, whatever objects you place in Details A should be placed into Details B as well.

Now when you run the report, the program will print every even numbered line with a white background and every odd numbered line with a green background.

Eliminating blank lines when fields are empty

It is very common to have two address lines in a customer table, one for street address (Address 1), and one that can be used for a suite number or mail stop (Address 2). 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 either by using multiple sections, or by suppressing blank lines.

To eliminate blank lines by using multiple sections

1. Use the Section Expert to create two new Details sections so that you have a total of three. See “Working with sections” on page 182.
2. Place the **Address 2** field in the middle section and the other data in the sections above and below it as you want it to appear in the report.
3. In the Section expert, highlight the middle section.
4. On the **Common** tab, select the **Suppress Blank Section** check box.

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 the report.

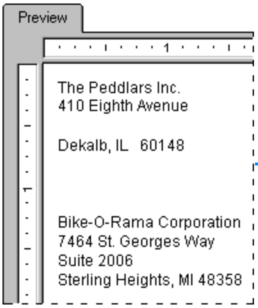
To suppress blank lines in embedded fields

1. Open your report in the **Design** tab, and click the desired text object—that is, the text object that causes blank lines to show for some records.

   Tip: To ensure that you have clicked a text object, look for the word Text in the status bar at the bottom left corner of the screen.

2. Right-click the text object and, on the shortcut menu, click **Format Text**.
3. Select the **Suppress Embedded Field Blank Lines** option in the Format Editor, and then click **OK**.

Now, when the report prints, unwanted blank lines will no longer appear in place of empty embedded fields. You can confirm your changes in the Preview tab.

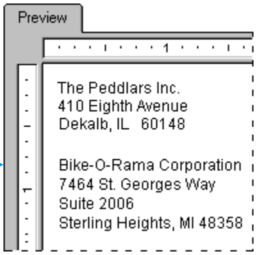


The Peddlars Inc.
410 Eighth Avenue

Dekalb, IL 60148

Bike-O-Rama Corporation
7464 St. Georges Way
Suite 2006
Sterling Heights, MI 48358

Before suppressing blank lines, the Address2 field prints a blank line if it is empty.



The Peddlars Inc.
410 Eighth Avenue
Dekalb, IL 60148

Bike-O-Rama Corporation
7464 St. Georges Way
Suite 2006
Sterling Heights, MI 48358

After suppressing blank lines, the Address2 field does not print if it is empty.

Adding blank lines conditionally

Use multiple sections to print a blank line on your report under specific conditions. For example, you may want to insert a blank line after every fifth record in the report.

To add blank lines conditionally

1. Use the Section Expert to create two Details sections. See "Working with sections" on page 182.
2. Place the report detail data in the top section.
3. Leave the second section empty.
4. In the Section expert, highlight the second section.
5. On the **Common** tab, select the **Suppress (No Drill-Down)** check box then click the conditional formatting button to its right.
6. Enter the following formula in the Format Formula Editor:

   ```
   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 for every fifth record printed), the program prints the second section, thus inserting a blank line.

Note: To insert a blank line under different conditions, you can modify your formula appropriately. See "Working with conditional formatting" on page 237.

Form letters

While form letters themselves are not necessarily multi-section reports, they are often used in multi-section reports to generate custom mailings. The topic "Printing conditional messages in form letters" on page 197, explains how to use multiple form letters or multiple versions of the same form letter for custom mailings.

Form letters often use text objects to hold the content of the report. The following sections provide an introduction to text objects and demonstrate how to use them in form letters.

Working with text objects

You will use many of the capabilities of text objects when creating form letters. A brief discussion of text objects should make it easier for you to create the form letter in the next section. Some things to consider are:

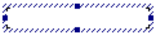
- a text object can contain both text and fields; you will use both in this example
- you can resize text objects; you will be resizing the text object so that it prints as a letter.

Text objects operate in two modes:

- Move/Resize mode
- Edit mode.

Move/Resize 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 the object by dragging any of the resizing handles, or you can 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
- by clicking the frame when the object is in edit mode.

Edit mode

When the object is in the edit mode, it appears as a broken line frame without sizing handles, and an in-place ruler appears at the top of the tab.

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 when it's inactive or in the move/resize mode.

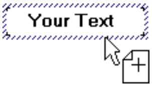
Each text object contains word processor capabilities, including the ability to change the fonts for individual characters and fields, and automatic word wrap. In the edit mode you can insert text and such non-text objects as database fields and formulas. Whenever the object is in edit mode, it contains an insertion point, a flashing vertical line that indicates the position at which 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 the cursor position when you click anywhere within the existing text.

As you work through the tutorials for multiple sections, it is always expected that you will type or place fields at the existing insertion point, unless otherwise indicated.

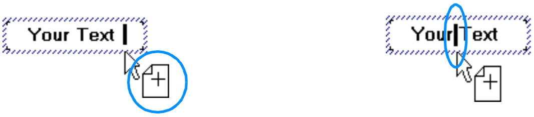
- To select text within a text object (to delete it, to change a font, etc.), position the cursor over the text and, when the I-beam cursor appears, drag the cursor to highlight the text you want to select.

- To select a field which is inside a text object, position the cursor over the field and, when the I-beam cursor appears, right-click.
- To insert text, type in the desired text and it will appear at the insertion point.



Note: It is critical that you see the Drag and drop cursor before placing the field. If it does not appear, you might place the field so that it overlays the text object instead of being inserted into 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 Drag and drop cursor. If you have existing text or fields in the text object, the insertion point moves as you move the Drag and 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.**

Related topics

"Working with text objects" on page 191.

Creating a form letter using a text object

The following section shows you how to create a form letter.

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 will be customized with company information from a different record.

If you have difficulty performing any of the steps, please see "Working with text objects" on page 191.

To create a form letter

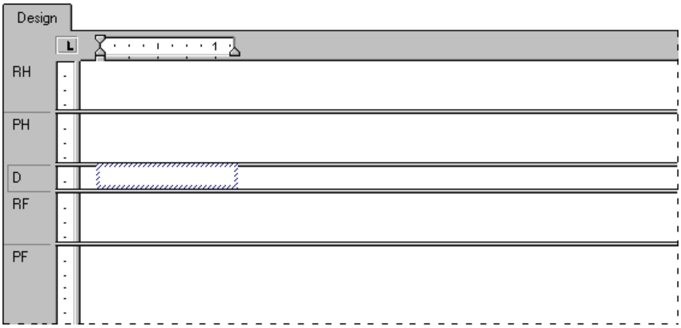
The letter will consist of a date, an inside address, a salutation, a one paragraph letter body, and a closing section.

1. Create a blank report. Use the **Customer** table of **Xtreme.mdb**.

   The Design tab appears with "Database Fields" highlighted in the Field Explorer dialog box.

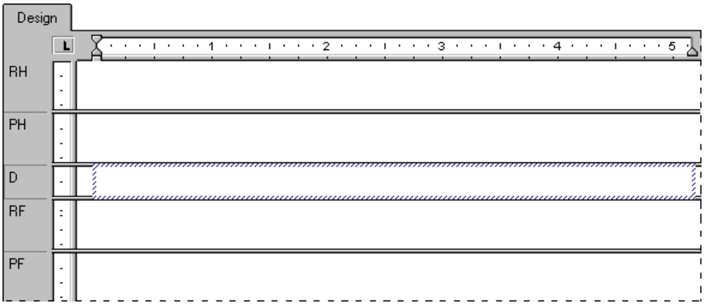
2. Since you do not want field titles to appear above the fields that you insert into the letter, clear the **Insert Detail Field Titles** check box on the **Layout** tab of the Options dialog box.

3 Insert a text object into the **Details** section of the 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 Double-click inside the text object to place it in edit mode. It is now ready for you to begin your work. The insertion point appears at the extreme left, inside the object.

Inserting a date

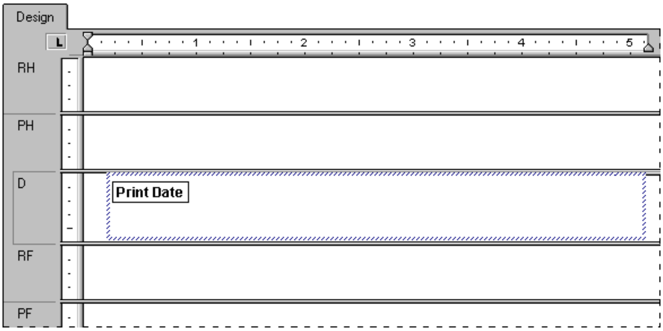
1 To insert a date into the letter, double-click **Special Fields** in the Field Explorer dialog box and scroll until you find **Print Date**.

2 Click **Print Date** and drag the placement frame into the text object and place it at the insertion point.

Note: To change the way the date is formatted in the letter, double-click the border of the text object to select it. Then right-click the Print Date field and choose Format [Print Date] from the shortcut menu. Make your modifications on the Date/Time tab of the Format Editor when it appears.

3 Press Enter twice to insert some white space between the date and the inside address and to move the insertion point down within the text object.

Tip: You might have to resize the Details section in order to see the resized text object.



Creating an inside address

1 To create the inside address, drag database fields into the text object from the **Customer** table in the Field Explorer dialog box.

2 Drag in the **Address 1** field and place it at the insertion point, and press Enter. The insertion point moves down to the line below.

3 Drag in the **City** field and place it at the insertion point.

4 Type a comma, followed by a space.

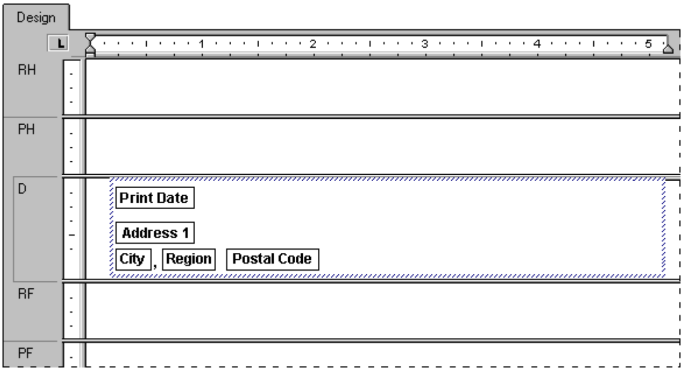
5 Drag in the **Region** field and place it at the insertion point.

6 Type in two spaces.

7 Finally, drag in the **Postal Code** field, place it at the insertion point, and press Enter. The insertion point moves down to the line below.

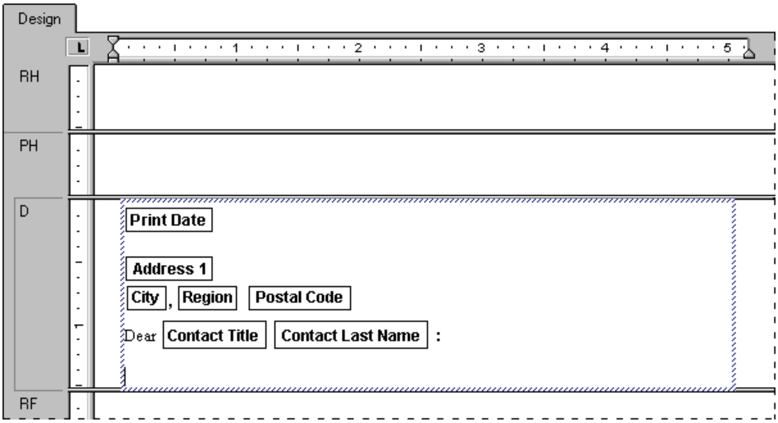
8 Press Enter one more time to move the insertion point down one more line to the position where the salutation should start. This completes the inside address.

Note: When a field is placed within a text object, it is automatically trimmed on both the left and right sides, so that it contains no extra white space.



Creating a salutation

1. Press Enter four times to move the insertion point down.
2. Type in the word “Dear” and a space (do not include the quotation marks).
3. In the Field Explorer dialog box, highlight the **Contact Title** field from the **Customer** table and drag it into the text object, placing it immediately after the space.
4. Insert a space. The program positions the insertion point immediately after the space.
5. In the Field Explorer 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.
6. Type a colon “:” at the insertion point (without the quotation marks) and press Enter twice to move the insertion point down two lines.



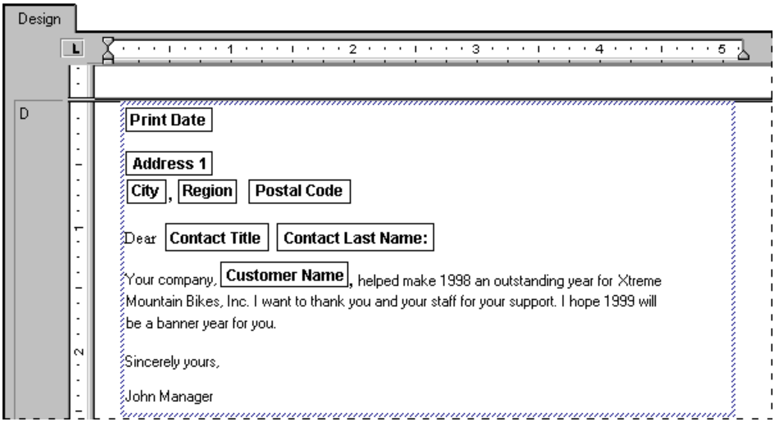
Creating the letter body

1. Now type “Your company” (do not include the quotation marks) and type a comma followed by a space.
2. Drag the **Customer Name** field into the text object and place it at the insertion point, just after the space.
3. Type a comma, followed by a space.
4. Type the following text (do not include the quotation marks): “helped make this year an outstanding year for Xtreme Mountain Bikes, Inc. I want to thank you and your staff for your support. I hope 2000 will be a banner year for you.”
5. Press Enter twice.

6 Type “Sincerely yours” (without the quotation marks), followed by a comma, and then press Enter four times.

7 Finally, to complete the form letter, type your name.

The Design tab should look similar to this:



Print Date

Address 1
City , Region Postal Code

Dear Contact Title Contact Last Name:

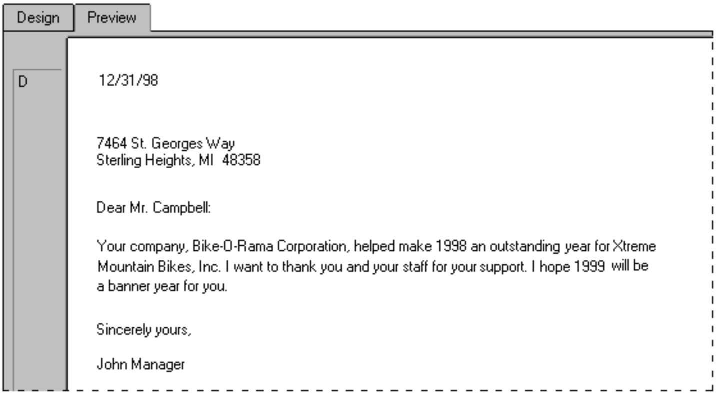
Your company, Customer Name, helped make 1998 an outstanding year for Xtreme Mountain Bikes, Inc. I want to thank you and your staff for your support. I hope 1999 will be a banner year for you.

Sincerely yours,

John Manager

8 Click **Print Preview** on the Standard toolbar to preview the form letter.

It should look similar to this:



12/31/98

7464 St. Georges Way
Sterling Heights, MI 48358

Dear Mr. Campbell:

Your company, Bike-O-Rama Corporation, helped make 1998 an outstanding year for Xtreme Mountain Bikes, Inc. I want to thank you and your staff for your support. I hope 1999 will be a banner year for you.

Sincerely yours,

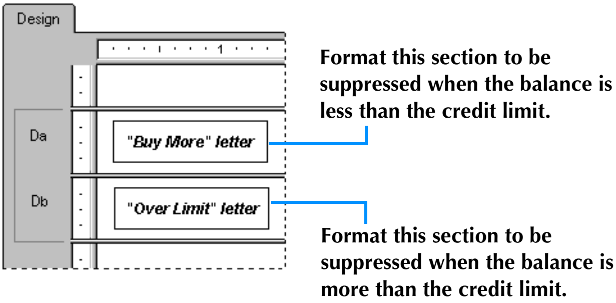
John Manager

Printing conditional messages in form letters

It is likely that you will 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 create both of these letters within a single report.

To create a conditional message

1. Use the Section Expert to insert a second Details section in the report. See "Working with sections" on page 182.
2. Create two form letters. Place a letter that encourages customers to buy more in the Details A section of the report; place a letter that encourages customers to bring their balance down in the Details B section of the report. See "Creating a form letter using a text object" on page 193.
3. Use the Section Expert to format the Details sections so that each is suppressed under certain conditions. For example:



Format this section to be suppressed when the balance is less than the credit limit.

Format this section to be suppressed when the balance is more than the credit limit.

Now, when a record indicates available credit, the *buy more* letter will print. When the account is over the credit limit, the *over limit* letter will print. And when the customer is right at the credit limit, nothing will print at all.

Related topics

"Formatting objects conditionally" on page 188.

"Working with conditional formatting" on page 237.

Formatting 12

Formatting refers to changes in the layout and design of a report, as well as the appearance of text, objects, or entire report sections. This chapter details methods you can use to draw attention to data, change the presentation of dates, numbers, and other values, hide unwanted sections, and perform a variety of other formatting tasks to give a report a professional appearance.

Formatting concepts

This section explains how to format a report. Formatting refers to changes you can make to the layout and design of a report, as well as the appearance of text, objects, or entire report sections.

You can use formatting to do many things, including:

- dividing sections of a report
- calling attention to certain data
- changing the presentation of dates, numbers, Boolean values, currency values, and text strings
- hiding unwanted sections
- giving the report a professional appearance.

The following topics describe the types of formatting you can do with Crystal Reports, giving step-by-step instructions for performing a variety of formatting tasks.

Note: There are many date formats you can choose to use on an English report, but if you send the report to a Japanese system, there may be some formatting irregularities. Not all English date formats are viewable on a Japanese system, and the same is true going from Japanese to English. For instance, abbreviated English months do not appear on a Japanese system and Japanese eras in short format do not appear on an English system.

Using the Report Design Environment

Design solutions

There are several things to keep in mind when designing reports that are distributed to different environments. For the best results, consider:

- Section characteristics
- Making an object underlay a following section
- Pre-printed forms
- Hiding report sections
- Hiding report objects
- Placing text-based objects
- Placing multi-line, text-based objects
- Importing text-based objects from a file
- Spacing between text-based objects
- Overflow Field Representation
- Selecting multiple objects

- Free-form placement
- Vertical placement
- TrueType fonts
- Page margins
- Default printer
- Printer drivers.

Section characteristics

A report consists of several sections, including the Report Header, Page Header, Group Header, Details, Group Footer, Page Footer, and Report Footer.

Each report section is made up of a series of lines. When a text-based object is placed in a section, it is placed on a line in such a way that the text is aligned to the baseline. The line's height is then adjusted by the printer driver so that it is high enough to accommodate the object.

- if you place another text-based object on the same line with a font size larger than that of the first object, the line's height extends to accommodate the second object
- if you place another text-based object on the same line with a font size even larger than the previous two objects, the line's height extends to accommodate the third object.

A line's height is determined by the text-based object with the largest font size on the line.

As you add text-based objects to a report, either in the same section or other sections, the line height adjusts to accommodate the various fonts. Since this vertical spacing is set up by the printer driver, it is difficult to create reports designed for pre-printed forms when they are printed in various environments.

When designing reports, you should do the following:

- always print a test page
- keep all font sizes the same
- be sure to print pre-printed forms on the same machine.

Making an object underlay a following section

Using this example, you can make the Xtreme logo (Xtreme.bmp) underlay multiple sections. This procedure is similar to the one for inserting a company watermark to serve as a report background.

To make an object underlay a following section, first place the object in the section above the section you want it to underlay. Then select the Underlay Following Sections check box in the Section Expert for the section in which the object is placed.

To create a basic report

1. Create a report using the Customer table in **Xtreme.mdb**.

   The Xtreme.mdb file is located in the `\Program Files\Seagate Software\Crystal Reports\Samples\En\Databases` directory, or the directory in which the program resides.

2. Place {customer.CUSTOMER NAME} and {customer.LAST YEAR'S SALES} side-by-side in the Details section of the report.

3. To remove unnecessary objects from this example, delete the field titles that the program places in the Page Header section of each field.

4. On the **Insert** menu, click **Group** to break the data into regional groups.

5. On the **Common** tab of the Insert Group dialog box, select {customer.REGION} as the sort and group by field.

6. Click **OK**.

To insert a picture onto the report

1. On the **Insert** menu, click **Picture**.

   Tip: Another way to do this is to click the Insert Picture button on the Supplementary toolbar.

2. Select the **Xtreme.bmp** file, then place it in the Page Header section, to the right of the report body.

   The Xtreme.bmp file is located in the `\Program Files\Seagate Software\Crystal Reports\Samples\En\Databases` directory, or the directory in which the program resides.

   Note: In this example, the picture is placed to the right of the fields to avoid underlaying the text. When you are working with a watermark, a subdued picture designed to be nearly invisible, place it directly above the text.

3. On the Standard toolbar, click **Print Preview**.

   The picture prints first (entirely in the Page Header section), then the body of the report follows.

4. Once you are finished previewing the report, return to the **Design** tab.

To make the picture underlay the following section

1. On the **Format** menu, click **Section**.

   The Section Expert dialog box appears.

2. In the **Sections** list, click **Page Header**, then select the **Underlay Following Sections** check box.

3 Click **OK** to preview the report again.

   The picture now prints in both the Group Header and Details sections, next to (instead of above) the text in the body of the report.

   Note: Using the technique of placing a picture to the right of the body of the report, you can set up a chart or an employee picture to print beside the details pertaining to that chart or employee.

4 Once you are finished previewing the report, return to the **Design** tab.

5 Resize the object vertically to make it two or three times larger, then preview the report again.

   The bitmap file now underlays more sections.

The area in which the picture underlays depends on the following conditions:

- the size of the picture
- the section in which the picture was originally placed
- the position of the picture in the section.

By modifying size and placement of an object, you can create a variety of visual effects, using the underlay feature.

Pre-printed forms

If you print on pre-printed forms, you will be able to:

- scan a form
- place it in the report as a bitmap
- use the underlay feature to line up the bitmap and report, as well as move objects anywhere you want them to appear
- eliminate the need to print the forms separately by printing the report and the form as a single unit.

Hiding report sections

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

Hide (Drill-Down OK)

The Hide property hides a section whenever you run the report. For example, in a summary report, the Hide property can be used to display only the summaries, but not the details behind the summaries. When the Hide property is applied to a section, it becomes visible when the Drill-down cursor is used to drill down on the section contents. This property is absolute; it cannot be conditionally applied using 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, then drill down to reveal the section contents. This property can be applied absolutely, or conditionally using a formula. This is useful for writing form letters. For example, in a form letter, you might create two Details sections: one to suppress when sales are over $X and one to suppress when sales are under $X.

Suppress Blank Section

The Suppress Blank Section property hides a section whenever there is nothing in it. If something is placed within the section, then it becomes visible.

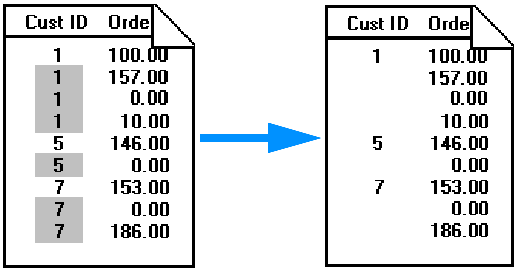
Hiding report objects

Crystal Reports has three formatting options in the Format Editor for hiding individual objects.

Suppress If Duplicated (Common tab)

The Suppress If Duplicated property 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

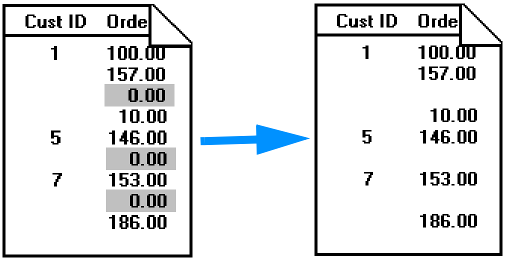
Duplicated values are suppressed and do not print.

Suppress If Zero (Number tab)

Tip: To find this option, click the Number tab of the Format Editor, then click the Customize button.

The Suppress If Zero property 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, select the Suppress Blank Section check box in the Section Expert.

Note: This will only work if there are no other objects in the section.Use the Section Expert to eliminate blank lines in this situation, then clear the Suppress Blank Section check box for the section that the field is in. This will eliminate the lines as long as there are no other objects in the section.



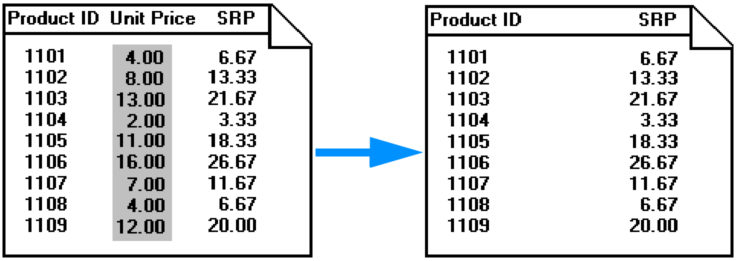
Cust ID	Orde
1	100.00
	157.00
	0.00
	10.00
5	146.00
	0.00
7	153.00
	0.00
	186.00

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.

Suppress (Common tab)

The Suppress property hides an object when you run the report. For example, it is common 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 you select this property, the selected object does 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.

Note: You can click the Conditional Formula button for any of these properties and create a formula that will make the setting conditional on some event. See "Working with conditional formatting" on page 237.

To set these properties, select the object, then click Format from the Supplementary toolbar to open the Format Editor dialog box. When the Format Editor appears, set the properties.

Placing text-based objects

When a text-based object is placed on a report, the object is represented by an object frame. The height of the object frame is based on the height of the font. The width, however, is determined differently, depending on the object you are working on.

- For database fields that are not memo fields, the width is initially determined by the width of the field as defined in the database, and by the average character width as provided by the selected font and font size.

  For example, you have a database field called {customer.LAST NAME} and your database defines this field as a text field with a length of 35 characters. When you place this field on your report, the width of the boundary is 35 times the average character width of the font and font size that the database field is formatted to. Remember that this is the initial default boundary width. The width can always be resized to increase or decrease as you see fit.
- For text-based objects, the default width is approximately 17 average character widths wide. Objects are different from database fields in that their width automatically expands as you enter in text and/or database fields into the object. As with all other text-based objects, the width can be resized by the user.
- For different number fields (double, single, integer, long integer, and byte) the default widths are all different. As with all text-based objects, the width can be resized by the user.

Preventing the truncation of text inside an object

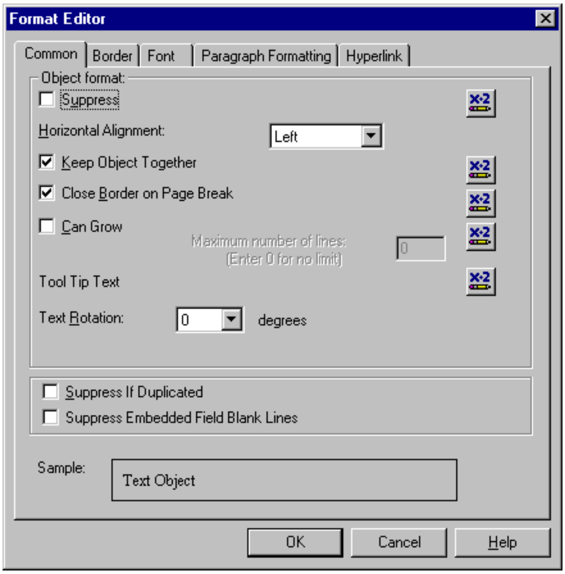
Whether the default widths are accepted or the text-based objects are resized, a problem could arise if the text inside the object prints right to the edge of the object frame. While the report may look fine on the machine it was designed on, when the report is printed using another printer driver that measures the font wider, the length of the text grows, but the object frame remains fixed. The resulting text is cut-off or truncated.

To prevent the truncation of text inside an object

1. Right-click the text object you want to format to bring up the shortcut menu.

   Tip: Another way to do this is to click the Format button on the Supplementary toolbar.
2. On the shortcut menu, click **Format Text**.

The Format Editor dialog box appears.



Object format:

☐ Suppress

Horizontal Alignment: Left

☑ Keep Object Together

☑ Close Border on Page Break

☐ Can Grow

Maximum number of lines: (Enter 0 for no limit) 0

Tool Tip Text

Text Rotation: 0 degrees

☐ Suppress If Duplicated

☐ Suppress Embedded Field Blank Lines

Sample: Text Object

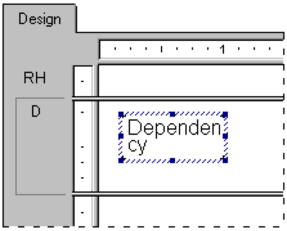
3 On the **Common** tab, select the **Can Grow** check box.

4 Click **OK** to save your changes.

The object is then formatted to print on multiple lines. If the text prints wider than the object, the text wraps onto additional lines.

Preventing breaks in non-spacing text inside an object

For text strings that do not contain spaces, such as single words, the text string is broken at the edge of the object frame before the line starts to wrap.



To prevent the breaks in non-spacing text inside an object

1. Select the object you want to format.
2. Expand the object frame to make it wider than the widest block of text inside the frame.

There are many times when the actual text in a database field is far less than the maximum amount the field can contain. For example, a {table.LAST NAME} field is designed with a field size of 80 and the longest name in the database is 28 characters. In this case, when you first place the field in your report, the field is 80 times the average character width. Reduce the width of the field, but include enough space to account for growth.

While each of these options offers an effective solution when dealing with a single text-based object in a section, there are still design considerations to take into account when placing more than one object in a section. When sizing an object, consider its placement with regard to other objects in the section.

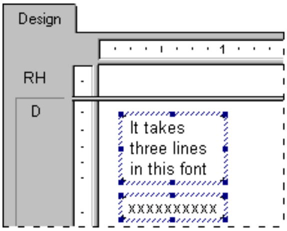
Avoid designing reports where the space between each object is very tight. Leave room for growth by expanding the width of the object by approximately 5 per cent. Or, if this is not possible, consider reducing the size of the font.

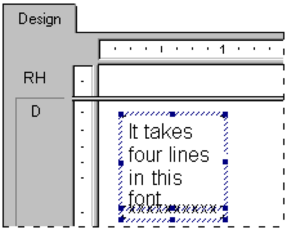
Placing multi-line, text-based objects

While text-based objects that are formatted to print on multiple lines follow the same design rules as other objects, they have one additional characteristic to be considered. If the printer driver expands or contracts the spacing of the text, word wrapping may differ, changing the number of lines necessary to print the object in order to accommodate growth or shrinkage.

When placing multi-line, text-based objects, you could encounter problems if other objects in the same section are placed directly below them.

Unlike single-line, text-based objects, expanding the object frame of a multi-line, text-based object to accommodate growth is not a viable option. When you do this, the line width increases according to the expanded boundaries.





So, when possible, place multi-line, text-based objects at the bottom of a section. If they require more lines to print, the section expands downward to accommodate the growth, and they do not endanger other objects.

Importing text-based objects from a file

Using Crystal Reports, you can import a formatted, text-based object from an existing file onto your report.

To import text-based objects from a file

1 Double-click the text-based object you want to format, then right-click it to bring up the shortcut menu.

2 On the shortcut menu, click **Insert from file**.

3 In the Open dialog box that appears, select the file in which your text-based object is stored, then click **Open**.

The object is imported from the file onto your report, replacing the object you selected to format in Step 1.

Spacing between text-based objects

Use the grid and guidelines options to help evenly align text-based objects.

You can select the Snap To Grid option, set the grid to a maximum of one inch, and make the grid visible or invisible on the Design tab, Preview tab, or both. For more information on working with grids, see "Using the grid" on page 209.

You can also work without a grid, placing objects wherever you want them on a report. You may want to work in a free-form environment while retaining the ability to align objects, or to move or resize them as a group. You can do this using guidelines. See "Designing with guidelines" on page 211.

Using the grid

The grid is a series of row and column coordinates. When the grid is selected, Crystal Reports enables you to place text-based objects only at these coordinates, not between them. You can then space data on your report and align objects as needed. If you attempt to place an object between grid coordinates, the object "snaps" to the grid; that is, the object automatically moves to the nearest set of row and column coordinates.

Each report section contains a design grid. You can select the grid on or off, as well as set it to different sizes when required. By default, the grid is not selected. See "Selecting the grid" on page 210.

Once set, the grid remains the same size for all sections. It is measured from the upper-left corner of each section and continues down and to the right, until the end of the section. A new grid of the same size then starts from the upper-left corner of the next section, and so on, through the end of the report.

If you select the Snap To Grid option, the following conditions occur:

- The upper-left corner of all newly placed, text-based and OLE objects snap to a grid point.
- Objects placed on a report before the Snap To Grid option is selected do not snap to the nearest grid point. They remain in the same place.
- If you resize an object, the side or sides that you are resizing snap to the nearest grid point.

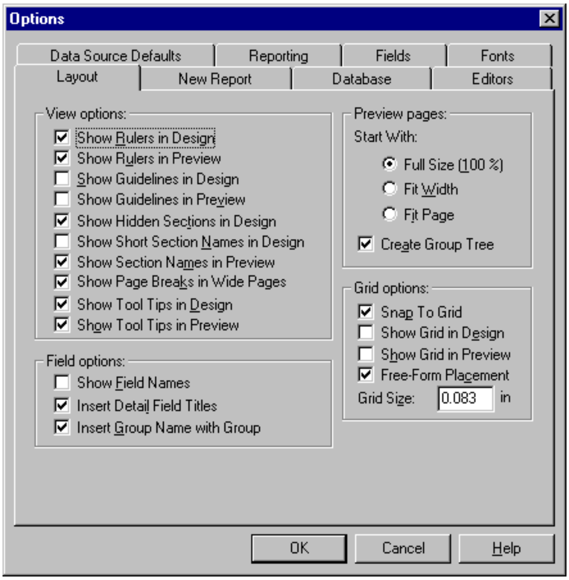
Selecting the grid

The Design and Preview tabs have an underlying grid structure that you can activate on the Layout tab in the Options dialog box.

To select the grid

1 On the **File** menu, click **Options**.

The Options dialog box appears.



View options:
☑ Show Rulers in Design
☑ Show Rulers in Preview
□ Show Guidelines in Design
□ Show Guidelines in Preview
☑ Show Hidden Sections in Design
□ Show Short Section Names in Design
☑ Show Section Names in Preview
☑ Show Page Breaks in Wide Pages
☑ Show Tool Tips in Design
☑ Show Tool Tips in Preview

Field options:
□ Show Field Names
☑ Insert Detail Field Titles
☑ Insert Group Name with Group

Preview pages:
Start With:
● Full Size (100 %)
○ Fit Width
○ Fit Page
☑ Create Group Tree

Grid options:
☑ Snap To Grid
□ Show Grid in Design
□ Show Grid in Preview
☑ Free-Form Placement
Grid Size: 0.083 in

2 On the **Layout** tab, in the Grid options area, you can show the underlying grid structure on the Design or Preview tab, or both, select free-form placement, and specify grid size.

3 Click **OK** to save your changes.

Designing with guidelines

Crystal Reports provides guidelines to help you align and size report objects with accuracy. Guidelines are non-printing lines that you can place anywhere on the Design and Preview tabs to aid in alignment. They have a snap property that automatically snaps objects to them.

Viewing guidelines

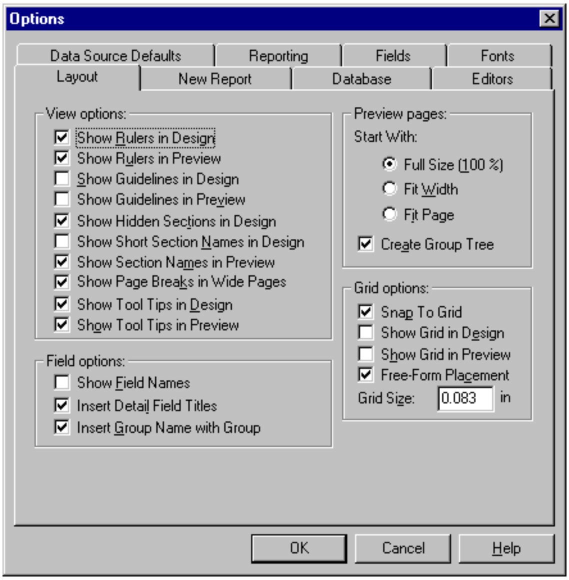
You can view guidelines on the Design and Preview tabs by selecting the view options in the Options dialog box.

To view guidelines

1 On the **File** menu, click **Options**.

Tip: Another way to do this is to select the Guidelines commands from the View menu.

The Options dialog box appears.



View options:

☑ Show Rulers in Design
☑ Show Rulers in Preview
☐ Show Guidelines in Design
☐ Show Guidelines in Preview
☑ Show Hidden Sections in Design
☐ Show Short Section Names in Design
☑ Show Section Names in Preview
☑ Show Page Breaks in Wide Pages
☑ Show Tool Tips in Design
☑ Show Tool Tips in Preview

Field options:

☐ Show Field Names
☑ Insert Detail Field Titles
☑ Insert Group Name with Group

Preview pages:

Start With:

● Full Size (100 %)
○ Fit Width
○ Fit Page

☑ Create Group Tree

Grid options:

☑ Snap To Grid
☐ Show Grid in Design
☐ Show Grid in Preview
☑ Free-Form Placement

Grid Size: 0.083 in

2 On the **Layout** tab, in the View options area, select the **Show Guidelines in Design** check box and/or the **Show Guidelines in Preview** check box.

3 Click **OK** to save your changes.

Inserting guidelines

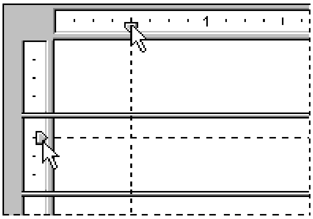
Although you can and should insert manual guidelines whenever necessary, Crystal Reports will automatically insert guidelines in certain situations:

- whenever you add a field or formula field to a report, the program creates a guideline at the left edge of the field frame and snaps the field and field title to it
- when a field is summarized, the program snaps the summary to the same guideline to ensure proper alignment
- when you right-click the shaded areas to the left of a section, then select the Arrange Lines option from the shortcut menu, the program creates one or more horizontal guidelines in the section and snaps the fields to them.

To insert, move, and remove guidelines manually

1 In the Design or Preview tab, click the ruler at the top to activate a vertical guideline; click the ruler on the left to activate a horizontal guideline.

Notice that each guideline is attached to an arrowhead on its originating ruler.



Note: If guidelines don’t appear, ensure that “Guidelines in Design” or “Guidelines in Preview” is selected on the View menu.

2 To position a guideline, drag its arrowhead along the ruler to the desired location.

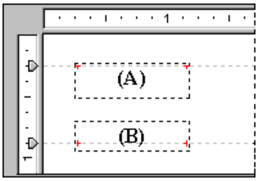
3 To delete a guideline, drag its arrowhead away from the ruler.

Note: If you select the Snap To Grid option, you can only insert or move guidelines in grid increments.

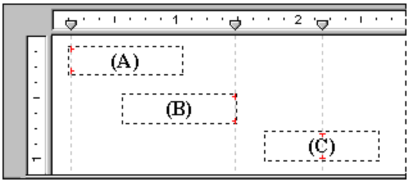
Snapping objects to guidelines

To snap an object to a guideline, drag the report object onto the guideline until the object’s edge sits atop the guideline. Snapping enables you not only to align report objects accurately, but also to re-position and re-size multiple objects together. Once several objects are snapped to a guideline, you can move all the objects by moving the guideline.

You can snap either the top or the bottom of an object to a horizontal guideline.



You can snap an object's left side, right side, or vertical midline (the invisible line that bisects an object vertically) to a vertical guideline.



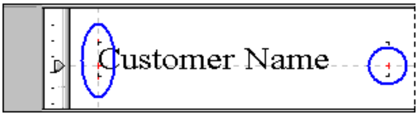
To snap objects to a guideline

1. Insert a guideline by clicking one of the rulers.
2. Drag the report object onto the guideline, so that one of the object's edges is on the guideline.

   Note: The guideline will usually darken or change color to show when the object is touching it correctly.

The snap property of guidelines works differently for text-based objects than for other objects, such as OLE objects. When a single-line text-based object snaps to a guideline, it is the baseline of the text, not the object frame, that snaps to the guideline. When a multi-line text-based object snaps to a guideline, either the baseline of the text or the object frame can be snapped to the guideline.

You can tell if a text-based object is snapped to a horizontal guideline by looking for the special indicators positioned on either side of the object directly at the baseline (as circled in the image below). If the object is snapped to a vertical guideline, the special indicators appear along the side of the object.



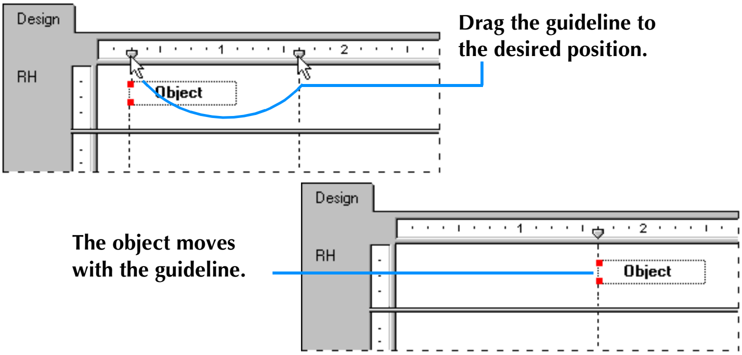
To place several text objects of different font sizes on one line with their baselines lined up, snap each object's baseline to the same horizontal guideline.

If you select the Snap To Grid option, and you do not select the Free-Form placement option, the following conditions occur:

- all objects snap to the guidelines for vertical placement
- the left boundary snaps to both vertical guidelines and grid points.

Positioning objects using guidelines

Once you've snapped one or more objects to a guideline, you can move all the objects at once by moving the guideline. To move the guideline, drag its arrowhead along the ruler.



Drag the guideline to the desired position.

The object moves with the guideline.

Note: When a guideline is moved, any object that is snapped to it is moved as well. But, if you move an object that is snapped to a guideline, the guideline does not move.

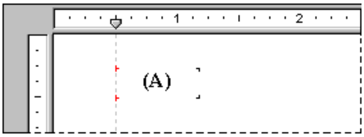
Resizing objects using guidelines

Use two guidelines to resize two or more objects that are either the same size (height or width) or different sizes. The process for resizing with two guidelines is the same in either case.

To resize objects using guidelines

1 Create a guideline.

2 Snap one side of the object to that guideline.

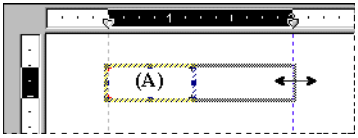


3 Create a second guideline to the right of the object.

Note: The guideline should not be touching the object.

4 Click the object to activate the sizing handles.

5 Drag the resizing handle over to the second guideline so that the object snaps to the guideline.



6 For each additional object you want to snap to both guidelines, repeat Steps 2 through 5.

7 If the objects are not the desired size, drag one or both of the guidelines until the objects are the correct size.

Indenting lines

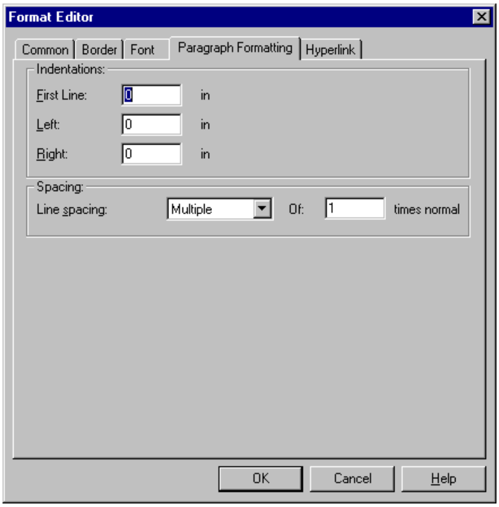
Using Crystal Reports, you can control line indentation for memo fields, string fields, and text-based objects. For objects, you have the option of indenting lines for a particular paragraph by positioning the cursor at the start of that paragraph. Or, if you select an object in its entirety, you can apply the same indenting specifications to all the paragraphs within that object.

Keep in mind that any line following a carriage return will be considered the first line of a new paragraph.

To indent lines

1 Right-click the field or object you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Paragraph Formatting**. The Format Editor dialog box appears with the Paragraph Formatting tab open.



Indentations:

First Line: 0 in

Left: 0 in

Right: 0 in

Spacing:

Line spacing: Multiple Of: 1 times normal

3 In the Indentations area, you can indent the first line of the paragraph; indent every paragraph line from the left margin; and indent every paragraph line from the right margin.

Note: Only indentation values within the range of the field or object width are accepted.

4 Click **OK** to save your changes.

Overflow Field Representation

Crystal Reports uses Overflow Field Representation to assist users when working with numeric or currency values in report cells. Normally, if a numeric or currency value is larger than the field containing it, that value is truncated, or “clipped.” For example, values like 100,000,000 might appear on the report as 1,000, or as 000 (depending on the properties you have set). This could potentially cause confusion when the report is read.

When the Allow Field Clipping option is cleared, numeric/currency field values that exceed the field size will be represented by number signs (######) in the Preview tab, letting you know immediately when the field is too small.

To allow for overflow field representation

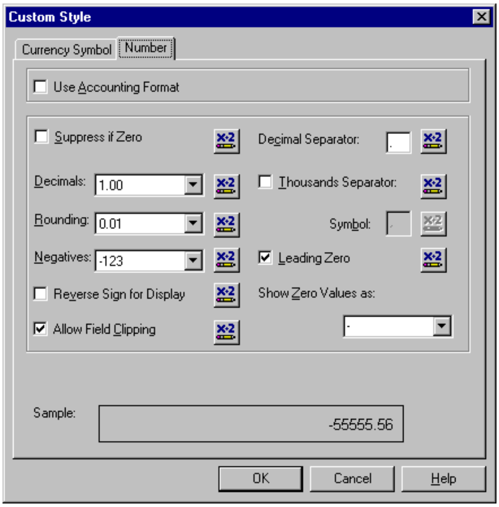
1 Right-click the currency field or number field you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Format Field**.

The Format Editor dialog box appears with the Number tab open.

3 Click the **Customize** button.

The Custom Style dialog box appears with the Number tab open.



☐ Use Accounting Format

☐ Suppress if Zero

Decimal Separator: .

Decimals: 1.00

☐ Thousands Separator:

Rounding: 0.01

Symbol: ,

Negatives: -123

☑ Leading Zero

☐ Reverse Sign for Display

Show Zero Values as: -

☑ Allow Field Clipping

Sample: -55555.56

4 To allow overflow field representation, clear the **Allow Field Clipping** check box.

   Note: You also have the option to click the Format button to enter a formula in the Format Formula Editor. In the Format Formula Editor, you can specify that field clipping will be disabled only when certain conditions are met.

5 Click **OK** to save your changes.

To view the results, refresh the report. If you disabled field clipping, any numeric/currency field values that are larger than the fields containing them will be represented by number signs (######).

Selecting multiple objects

You can select multiple objects, including text, field, chart, map, bitmap, OLAP grid, Cross-Tab and OLE objects, to format them together.

Once you have selected multiple objects, you can move, align, size, cut, and copy and paste them as a group. You can also change their font, color, and paragraph style.

Objects are moved, aligned, and sized based on a “main” object, which is usually the last object you select. You can change the main object to another by right-clicking the desired object.

Note: When moving multiple objects, if the new location does not accommodate all of the selected objects, but can accommodate the main object, then the main object and those objects that can be accommodated will be moved. The other objects will remain in their original position.

To select multiple objects

1 Click one object, and Ctrl-click the other objects you want to select.

2 Right-click the main object.

3 On the short-cut menu, select the appropriate formatting option.

Free-form placement

You may also want to use Crystal Reports without a grid, in a free-form environment similar to that of a drawing program. Free-form means that you can place objects anywhere you want them to appear on your report. Your only restriction is that the program does not allow you to place chart and Cross-Tab objects on the Page Header, Page Footer, or Details section.

To work in a free-form environment, select the Free-Form Placement option on the Layout tab in the Options dialog box.

Free-form placement on

In order to create dynamic reports and reduce printer driver dependency as much as possible, it is recommended that all sections of the report be formatted with the Free-Form Placement option turned on. This is especially true if your report includes OLE objects such as charts, boxes, lines, and bitmapped images (e.g. company logos).

When a section is formatted with free-form placement, *all* objects can be placed *anywhere* on that section. Crystal Reports places objects within a section based on their absolute coordinates. These coordinates determine the vertical placement of objects on your report. What this means is that you control the vertical placement of single-line, text-based objects rather than the printer driver. In doing so, you can better protect your reports from printing inconsistencies caused by using different printer drivers.

However, while the printer driver no longer controls the vertical spacing of text-based objects within the sections, it still determines horizontal spacing of text within the text-based objects, as well as the inter-line spacing of multi-line, text-based objects. So, while free-form placement gives you better control, you must still take into account these considerations when designing your reports. See "Placing multi-line, text-based objects" on page 208.

Free-form placement off

If a section has the Free-Form Placement option turned off, the program no longer references the object's absolute coordinates to determine where it prints, except under the following conditions.

- the absolute *x* coordinate is still referenced to determine where each object begins printing horizontally (left and right placement)
- the *y* coordinate is still referenced for vertical placement of the object, but the coordinates may be adjusted by the Crystal Reports Designer when the printer driver changes.

So, if the report is printed using a printer driver that measures inter-line spacing *greater* than the original printer driver, the *y* coordinate increases and the text-based object prints farther down the page. With the Free-Form Placement option turned off, the user no longer controls the vertical placement of text-based objects; the printer driver determines that instead.

However, the placement of OLE objects such as graphics, boxes, and lines is *not* controlled by the printer driver. Therefore, when placing and printing, the Crystal Reports Designer always references the absolute coordinates of these objects. Due to the various placement methods, problems can arise when both text-based objects and OLE objects are placed on a report.

For example, a box (OLE object) is placed around a database string field (text-based object). Everything looks great and everything is aligned as it should be. However, if the report is printed using a printer with an especially high measurement for inter-line spacing, then the following conditions occur.

- the placement of the box does not change relative to the section in which it is placed (the *x* and *y* coordinates do not change)
- the vertical placement of the text-based object changes because the *y* coordinate is adjusted upward.

Note: The value is a measurement from the upper-left corner of the section; so, the greater the value, the farther down the page the object prints.

If you are distributing reports that have sections formatted with the Free-Form Placement option turned off, and you want the objects in those sections to be surrounded by a border or formatted with lines, it is better to modify the objects' border properties than to insert lines and boxes. This way, the borders always stay with the objects. See "Adding borders, color, and shading to a field" on page 227.

Remember, *any* section can be formatted with the Free-Form Placement option turned on or off. While it is better to turn the option on in some cases and not in others, it is highly recommended that every section of your report be formatted in the *same* manner.

Free-form placement using guidelines

You may want to work in a free-form environment while retaining 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 tabs. Guidelines have a snap property. When you move an object within a guideline's magnetic range, the object snaps, or attaches itself, to the guideline.

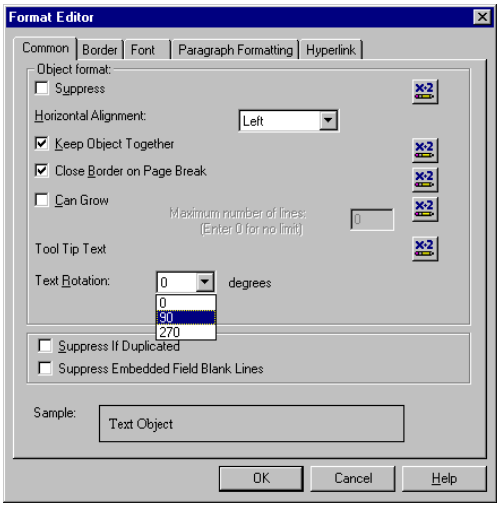
- Once an object is snapped to a guideline, if the guideline is moved, the object also moves.
- If you have several objects snapped to a guideline, they all move when the guideline is moved.
- If you have several objects snapped to a guideline on two sides (right and left, or top and bottom) and one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

Using guidelines in a free form environment provides flexibility with control. See "Designing with guidelines" on page 211.

Vertical placement

On the Common tab of the Format Editor, you can use the text rotation options to vertically align the fields and text-based objects on your report.

When you select a text rotation of 90 degrees, the text shifts 90 degrees in a counter-clockwise direction.



Object format:

☐ Suppress

Horizontal Alignment: Left

☑ Keep Object Together

☑ Close Border on Page Break

☐ Can Grow

Maximum number of lines: (Enter 0 for no limit) 0

Tool Tip Text

Text Rotation: 0 degrees

0
90
270

☐ Suppress If Duplicated

☐ Suppress Embedded Field Blank Lines

Sample: Text Object

When you select a text rotation of 270 degrees, the text shifts 270 degrees in a counter-clockwise direction.

Note:

- If text rotation is left at 0 degrees, your report is horizontally formatted, left to right.
- For text rotation of text-based objects, the Can Grow option that prevents the truncation of text inside an object is automatically cleared. For more information on the Can Grow option, see "Preventing the truncation of text inside an object" on page 206.
- Vertically formatted text that spans over the edge of the page cannot be displayed as part of your report.

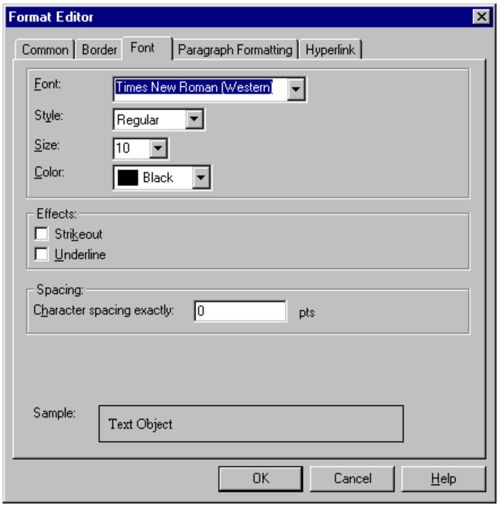
Inserting character and line spacing

With Crystal Reports, you can specify the amount of spacing between characters or lines for memo fields, string fields, and text-based objects.

To insert character and line spacing

1. Right-click the field or object you want to format to bring up the shortcut menu.
2. On the shortcut menu, click **Font**.

   The Format Editor dialog box appears with the Font tab open.

   Font: Times New Roman (Western)
   Style: Regular
   Size: 10
   Color: Black

   Effects:
   □ Strikeout
   □ Underline

   Spacing:
   Character spacing exactly: 0 pts

   Sample: Text Object

   You will use this tab to set up the character spacing values.
3. In the Spacing area, in the **Character spacing exactly** field, specify the value *n* that each character occupies.

   The value *n* is defined as the distance in number of points measured from the start of one character to the start of the next. When you change the character spacing, you change only the spacing between adjacent characters, not the font size of the characters.

   For example, if you specify a 14-point font with a character spacing of 14 points, each character will remain as a 14-point font size, occupying a space that is 14 points wide.
4. Click the **Paragraph Formatting** tab.

   You will use this tab to set up the line spacing values.
5. In the Spacing area, in the **Line spacing** field, specify the line spacing as a multiple of the font size you are using, or as an exact number of points.
6. Click **OK** to save your changes.

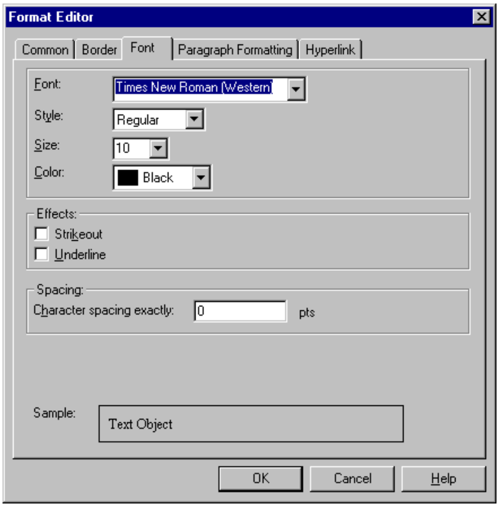
Setting fractional font sizes

On the Font tab of the Format Editor, you can select a fractional font size for database fields and text-based objects on your report.

To set fractional font sizes

1 Right-click the field or object you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Font**. The Format Editor dialog box appears with the Font tab open.



Font: Times New Roman (Western)
Style: Regular
Size: 10
Color: Black

Effects:
□ Strikeout
□ Underline

Spacing:
Character spacing exactly: 0 pts

Sample: Text Object

3 In the **Size** list, type the desired fractional font size for the field or object.

Note: The number you type must be between 1 and 1638. Crystal Reports will automatically round all fractional entries to the nearest 0.5. Consequently, in your report, you can use the fractional font sizes 1.5, 2.5, 3.5, and so on, up to 1637.5.

4 Click **OK** to save your changes.

Note: When setting fractional font sizes for individual database fields and text-based objects that you've already placed on your report, you must make your changes manually—that is, by following these procedures. (This is because the existing font settings of objects in your report will override your default Options.) However, you can use the File > Options > Fonts tab to adjust your default font settings: these default Options will affect the new reports that you create, along with any new objects that you add to an existing report.

TrueType fonts

Designing your report with printer-specific fonts can lead to problems when using different printers. The fonts may not be supported by the other printers, or if they are supported, the fonts may not be installed on the printers.

During the printing process, if you encounter printer-specific fonts that are unrecognizable to the printer driver, Crystal Reports substitutes the fonts, creating inconsistent results. To avoid this situation, only common TrueType fonts should be used when designing reports.

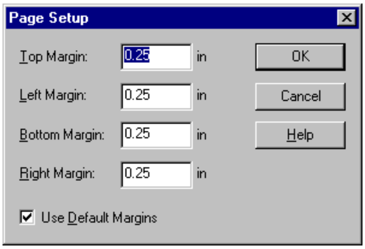
Page margins

Setting specific margins

Crystal Reports gives you the option of setting margins to meet your specifications.

To set specific margins

1 On the **File** menu, click **Page Setup**.
  The Page Setup dialog box appears.



Top Margin: 0.25 in
Left Margin: 0.25 in
Bottom Margin: 0.25 in
Right Margin: 0.25 in
☑ Use Default Margins

2 Change the default page margins to fit your needs.

3 Click **OK** to save your changes.

Note: 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.

Using default margins

If you decide to design your report using the default margins, the following problems may occur.

- When printing a report in another environment where the printer's default margins are *greater* than its setting, the report objects on the right side of the report print off the page.
- When printing a report in another environment where the printer's default margins are *smaller* (allowing a larger printing area), the entire report moves to the left side of the page.

It is recommended that you always set your own margins. Even if the margins you want to use are the same as the default margins, be sure that the Use Default Margins option in the Page Setup dialog box is not selected, and you set your margins manually using Page Setup.

Default printer

In general, it is a good idea not to choose a specific printer. Even though the printer may be identical to the default printer, how the printer is recognized can still vary for different operating systems.

For example, an HP Laser III printer is being installed on three different operating systems.

- With Microsoft Windows 95 and 98, the printer name can be changed so that HP Laser III is Front Reception Printer, but the printer driver will be listed as HPPCL5MS.DRV.
- With Microsoft Windows NT, the printer name is also referenced and can be changed by the user, but the printer driver is always WINSPOOL.

If you select a specific printer, Crystal Reports looks for that printer by name. If the printer you selected cannot be found, the default printer is chosen, resulting in the possibility of printing inconsistencies.

When selecting a specific printer, such as a label printer or a printer dedicated to printing invoices, the printer name must be the same as the name of the printer the report was designed on. Be aware that anyone printing the report must use that same printer or they could encounter problems.

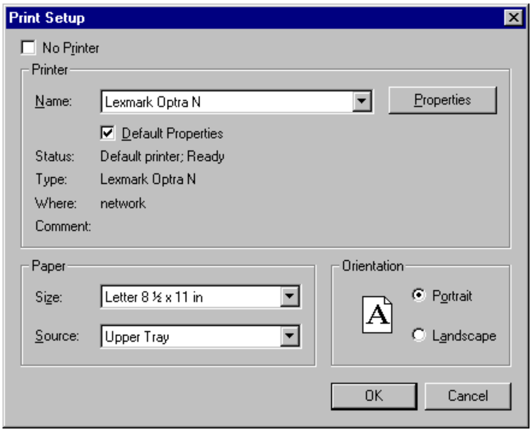
If your report is part of an application that you are distributing, you can provide a Select Printer dialog box. Using this dialog box, users of your report can choose the correct printer or rename the printer accordingly.

Setting page orientation and paper size

You can print your reports using either portrait or landscape orientation, and in a variety of paper sizes. You can specify these options using the **Printer Setup** command from the File menu.

To set page orientation and paper size

1 On the **File** menu, click **Printer Setup**.
The Print Setup dialog box appears.



Print Setup

□ No Printer

Printer

Name: Lexmark Optra N

☑ Default Properties

Status: Default printer; Ready

Type: Lexmark Optra N

Where: network

Comment:

Paper

Size: Letter 8 ½ x 11 in

Source: Upper Tray

Orientation

● Portrait

○ Landscape

2. In the Printer area, in the **Name** list, select the printer you want to use, if it is not already the default printer.
3. In the Paper area, in the **Size** and **Source** lists, select the desired paper.

   The paper size options are directly related to the printer you select. 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.
4. In the Orientation area, click **Portrait** or **Landscape**.
5. Click **OK** to save your changes.

Printer drivers

Updating printer drivers

In order to maintain performance, Crystal Reports queries the printer driver for each of the font elements (font metrics), such as average character height, character width, height of the ascenders and descenders, internal leading, and so on. A problem may develop if using an older printer driver that does not return the font metrics accurately. If you are experiencing problems when printing (missing fields, incorrect formatting, and so on), it is recommended that you obtain and install the most recently updated drivers for your printer. In many cases, the newer printer drivers provide accurate font metrics and any printing issues are quickly resolved.

Inconsistencies due to printer drivers

When printing, inconsistencies may occur if different printer drivers are used to create and print your reports. These inconsistencies are a result of the various methods that individual printer drivers use to measure text metrics such as font size. When printed, text-based objects may be misaligned, truncated, or overprint each other. Examples of text-based objects include string or character fields, text objects, memo fields, numeric fields, and formula fields.

Problems such as these may arise when you have:

- two identical printers, but each one is using a different printer driver
- two different printers using the same printer driver
- two different printers using different printer drivers
- one printer driver that uses the TrueType font and a second printer driver that maps TrueType fonts to PostScript fonts
- two identical printers using the same printer driver, but each one is printing from a different version of Microsoft Windows
- two identical printers using the same printer driver, but the printer drivers are different versions

- two identical printers, two identical printer drivers, and two identical operating systems, but the resolution of the video drivers is different.

Therefore, while a document using one printer driver may require six full lines to display a block of text:

- using a second printer driver that measures fonts narrower could result in the same block of text requiring less than six full lines
- using a third printer driver that measures fonts wider could require more than six full lines.

For the most part, this situation cannot be avoided. The goal of the report distributor is to design reports that accommodate printer driver dependency and still print consistently using different printer drivers. To do this, Crystal Reports provides several design solutions. If taken into account when creating your report, these solutions can ensure proper printing and distribution for your report in almost any environment.

Formatting properties

You can set formatting properties using the Format Editor for objects and the Section Expert for report sections. In most cases, you can set one of two types of properties:

- absolute (always apply the property)
- conditional (apply the property only if certain criteria are met).

Working with absolute formatting

Absolute formatting is formatting that applies under any condition. This type of formatting property always follows a *select, then apply* procedure. For example, you *select* what it is that you want to format (object or section), *then* you *apply* the formatting to the selection using property settings.

You can use the following dialog boxes to format your reports:

- Format Editor to format field values
- Section Expert to format entire sections
- Highlighting Expert to conditionally format currency and number fields.

Each of these dialog boxes contains a number of different formatting properties, as well as the tools for turning the properties on or off and specifying attributes.

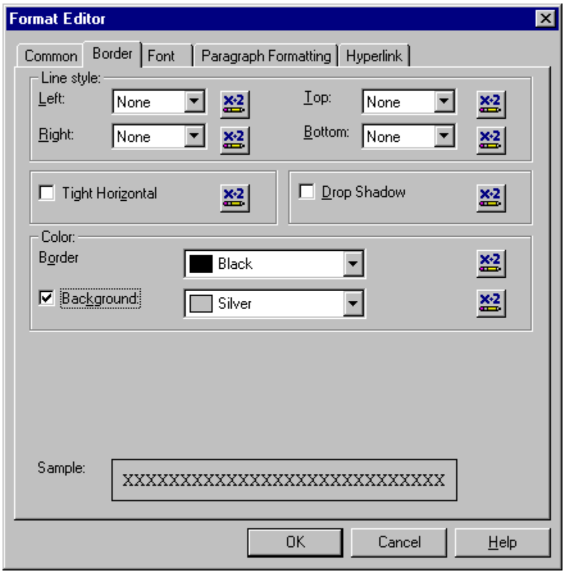
Adding borders, color, and shading to a field

Crystal Reports allows you to add borders, color, and shading to fields on your report in order to emphasize important data and create professional-looking reports.

To add borders, color, and shading to a field

1 Right-click the field you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Borders and Colors**.
The Format Editor dialog box appears with the Border tab open.



Format Editor

Common | Border | Font | Paragraph Formatting | Hyperlink

Line style:

Left: None   Top: None

Right: None   Bottom: None

□ Tight Horizontal   □ Drop Shadow

Color:

Border: Black

☑ Background: Silver

Sample: XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

3 Select the line style, color, and background color of the field.

4 Click **OK** to save your changes.

Changing your default field formats

Crystal Reports allows you to display database fields in almost any format on your report. This section describes how to use the Options command to control the default format settings that Crystal Reports uses when you add a field to any report. In the Options dialog box, you can set the default formats for database fields of the following type: String, Number, Currency, Date, Time, Date/Time, and Boolean.

Note: When you change default field formats, your new settings affect only the objects that you subsequently add to a report. To format fields that you've already added to a report, you must right-click the field in the report and select Format Field from the shortcut menu.

To specify default formats for fields

1. On the **File** menu, click **Options**.
2. In the Options dialog box, click the **Fields** tab.
3. Click the button appropriate to the type of field you want to format (String, Number, Currency, Date, Time, Date/Time, or Boolean).

   The Format Editor appears.
4. Use the Format Editor's tabs to specify the formats you want.
5. Click **OK**.

Setting default formats for Date, Time, and Date/Time fields

The following procedures first describe how to specify standard formats for Date, Time, and Date/Time fields, and then describe how to customize the formats for such fields.

Note: These default settings will affect only the objects that you subsequently add to a report. To format fields that you've already added to a report, you must right-click the field in the report and select Format Field from the shortcut menu.

To set standard default formats for Date, Time, and Date/Time fields

1. On the **File** menu, click **Options**.
2. In the Options dialog box, click the **Fields** tab.
3. To open the Format Editor, click the button appropriate to the field you want to format (**Date**, **Time**, or **Date/Time**).

   Note: If you click the Date/Time button in the Format Editor, then any subsequent changes will affect Date/Time fields only. You must click Date or Time to format independent date fields or time fields.
4. In the Format Editor dialog box, click the **Date / Time** tab.
5. Select a predefined format from the list (or click **Customize** to create your preferred format). When you click a new format, you can preview the results in the Sample area of the Format Editor.

   Note: From the list of predefined formats, you can choose the System Default settings to ensure that Crystal Reports uses the formats dictated by Windows. You can alter your system's settings in the Regional Settings Properties dialog box, located in the Control Panel.
6. Once you've selected a format, click **OK** in the Format Editor dialog box.
7. Click **OK** in the Options dialog box.

Now, when you add Date, Time, or Date/Time fields to a report, Crystal Reports should use the format you specified.

To customize formats for Date, Time, and Date/Time fields

1 On the **File** menu, click **Options**.

2 In the Options dialog box, click the **Fields** tab.

3 Open the Format Editor by clicking the button appropriate to the field you want to format (**Date**, **Time**, or **Date/Time**).

4 In the Format Editor dialog box, click the **Date / Time** tab.

5 Click **Customize** to open the Custom Style dialog box.

Note: If you chose to format Date/Time fields at Step 3, then you will see three tabs in the Custom Style dialog box (Date/Time, Date, and Time). The formats specified in these tabs apply only to the two elements of Date/Time fields, and will not affect the formats specified for independent date fields or time fields.

6 Create your preferred format by adjusting the various options in the Custom Style dialog box.

7 Once you've finished designing your format, click **OK** in the Custom Style dialog box.

8 Click **OK** in the Format Editor dialog box.

9 To format another type of field, click the appropriate button in the Options dialog box. Otherwise, click **OK** to return to Crystal Reports.

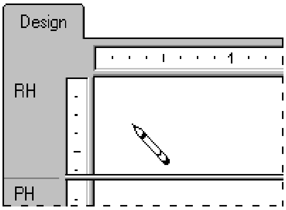
Now, when you add Date, Time, or Date/Time fields to your reports, Crystal Reports should use the customized format that you created.

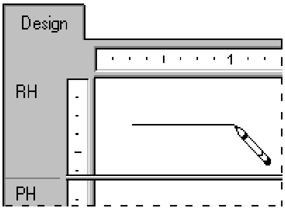
Adding and editing lines

Crystal Reports allows you to add lines to a report to emphasize important data and create professional-looking reports.

To add lines to a report

1 On the Supplementary toolbar, click **Insert Line**.





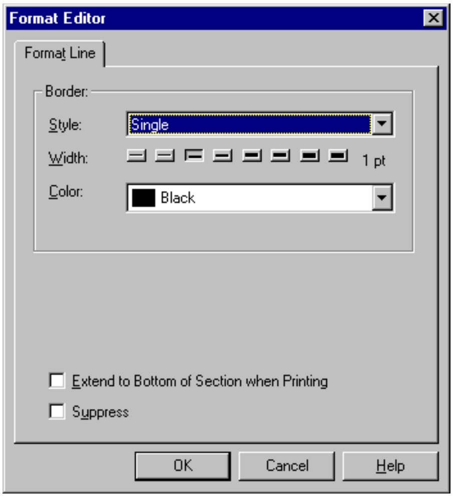
2 Use the pencil cursor to draw the line where desired.

To edit lines on a report

1 Right-click the line you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Format Line**.

The Format Editor dialog box appears.



Border:

Style: Single

Width: 1 pt

Color: Black

□ Extend to Bottom of Section when Printing

□ Suppress

3 On the **Format Line** tab, make the desired changes to the line.

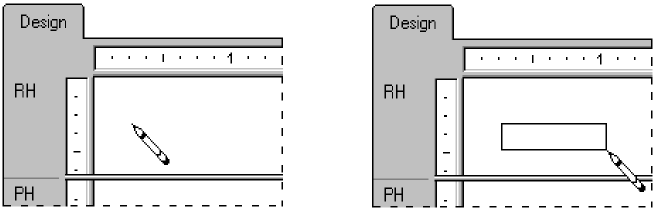
4 Click **OK** to save your changes.

Adding and editing boxes

Crystal Reports allows you to add boxes to a report to emphasize important data and create professional-looking reports.

To add boxes to a report

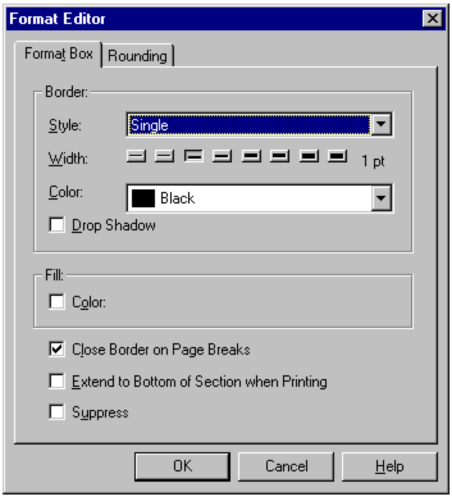
1 On the Supplementary toolbar, click **Insert Box**.



2 Use the pencil cursor to draw the box where desired.

To edit boxes on a report

1. Right-click the box you want to format to bring up the shortcut menu.
2. On the shortcut menu, click **Format Box**.
   The Format Editor dialog box appears.



Border:

Style: Single

Width: 1 pt

Color: Black

□ Drop Shadow

Fill:

□ Color:

☑ Close Border on Page Breaks

□ Extend to Bottom of Section when Printing

□ Suppress

3. On the **Format Box** tab, make the desired changes to the box.
4. Click **OK** to save your changes.

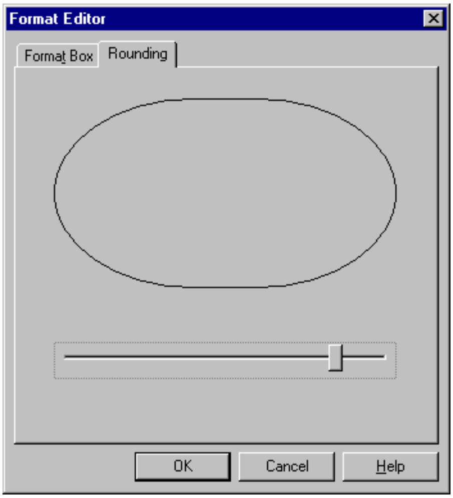
Adding shapes to a report

When designing report formats in Crystal Reports, you can insert a variety of shapes such as circles, eclipses, and boxes with rounded corners, as part of your report. This is especially useful for formatting reports in languages that require these shapes to effectively communicate.

To add shapes to a report

1. Add a box to your report.
   See "Adding and editing boxes" on page 230.
2. Right-click the box to bring up the shortcut menu.
3. On the shortcut menu, click **Format Box**.

4 In the Format Editor that appears, click the **Rounding** tab.



5 Move the slider to the right to increase the curvature of the box corners.

The box that you started with gradually changes to an ellipse or circle, depending on how far you move the slider to the right.

6 Once the appropriate shape is created, click **OK** to save your changes.

Using conventional accounting formats

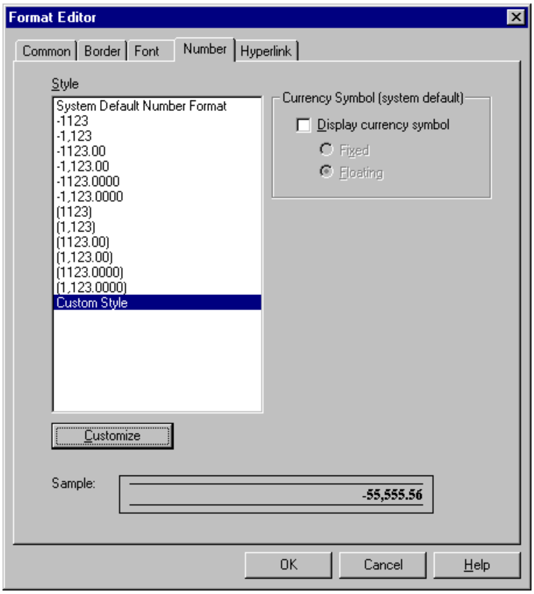
As a way of supporting the conventions used in the accounting profession, Crystal Reports lets you decide on how to display the currency symbol, negative values, and zero values on your financial reports. You can also set up your report to reverse the signs for credit and debit amounts.

To use accounting conventions in a report

1 Right-click the currency field or number field you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Format Field**.

The Format Editor dialog box appears with the Number tab open.



Style

- System Default Number Format
- -1123
- -1,123
- -1123.00
- -1,123.00
- -1123.0000
- -1,123.0000
- (1123)
- (1,123)
- (1123.00)
- (1,123.00)
- (1123.0000)
- (1,123.0000)
- Custom Style

Currency Symbol (system default)

□ Display currency symbol

○ Fixed

● Floating

Sample: -55,555.56

3 In the Style area, select how you want the system number format to appear for either positive or negative values.

4 In the Currency Symbol (system default) area, specify how you want the currency symbol to appear with the values on your report.

5 Click **OK** to save your changes.

To customize the accounting conventions for a report

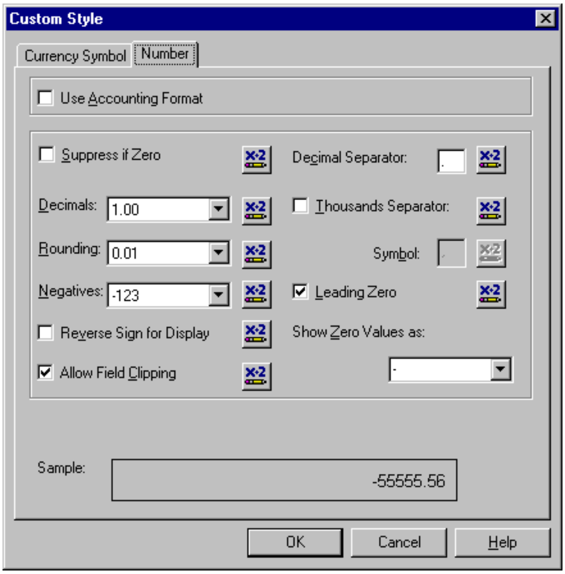
1 Right-click the currency field or number field you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Format Field**.

The Format Editor dialog box appears with the Number tab open.

3 In the Style area, select **Custom Style**, then click the **Customize** button.

The Custom Style dialog box appears with the Number tab open.



☐ Use Accounting Format

☐ Suppress if Zero

Decimal Separator: .

Decimals: 1.00

☐ Thousands Separator:

Symbol: ,

Rounding: 0.01

Negatives: -123

☑ Leading Zero

☐ Reverse Sign for Display

Show Zero Values as: -

☑ Allow Field Clipping

Sample: -55555.56

4 Select the **Use Accounting Format** check box.

Once you select this option, the following conditions occur:

- In the Negatives list, how the negative values appear on your report are determined by the Windows locale settings. The negative values are represented by either a minus sign or brackets.
- In the Show Zero Values as list, the dash symbol is automatically selected to represent zero values on your report.
- On the Currency Symbol tab of the Custom Style dialog box, the currency symbol is positioned on the left-side of the currency and numeric values.

Note: Changes made to the Windows locale settings are implemented only after you exit and restart Crystal Reports.

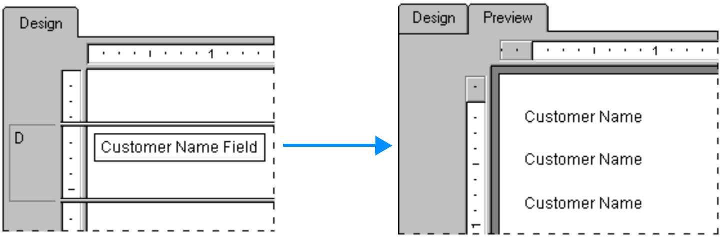
5 Select the **Reverse Sign for Display** check box to reverse the signs for debit and credit amounts in your financial reports.

6 Click **OK** to save your changes.

7 Click **OK** again to return to your report.

Using white space between rows

The height of a section in relation to the objects within it affects the amount of white space that appears between rows on the report.



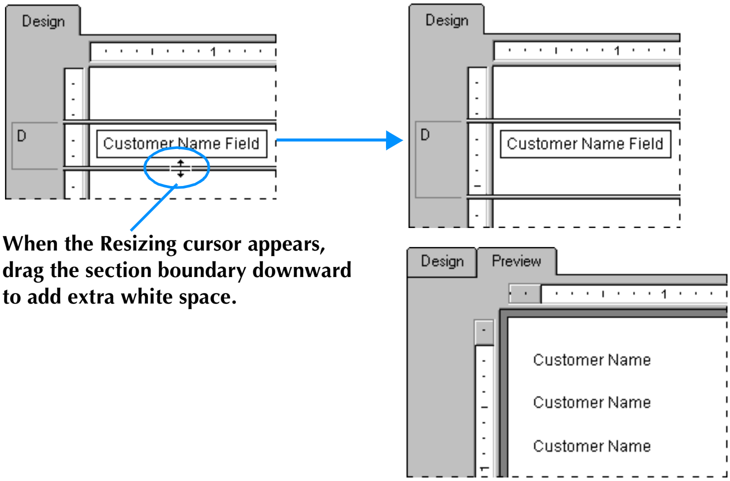
The free-form Design tab lets you add and delete white space in two ways:

- using the Resizing cursor to resize the area on the Design tab
- changing the option in the Section Expert.

Note: You can also right-click the shaded area to the left of the section and select **Fit Section** from the shortcut menu. The program automatically resizes the section so that the bottom boundary is even with the baseline of the bottom object in the section.

Adding white space by resizing

To add extra white space between rows in the report, move the pointer over the lower section boundary line. The pointer changes to a Resizing cursor.

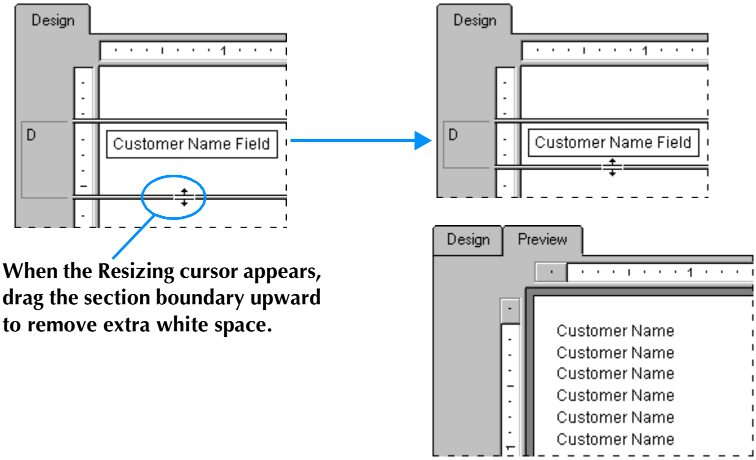


When the Resizing cursor appears, drag the section boundary downward to add extra white space.

Note: White space can also be added to a section by right-clicking the shaded area to the left of the section and selecting **Insert Line** from the shortcut menu. The program resizes the section automatically, adding the amount of space necessary to hold a line of typical database fields.

Deleting white space by resizing

To delete unnecessary white space within a section, move the pointer over the lower section boundary line. The pointer changes to a Resizing cursor.



When the Resizing cursor appears, drag the section boundary upward to remove extra white space.

Deleting white space by suppressing a section

If an entire section is blank (for example, if you are not putting anything into the Page Footer section of the report), you can eliminate the unnecessary white space by suppressing the section in the Section Expert.

To delete white space by suppressing a section

1 On the **Format** menu, click **Section**.

The Section Expert dialog box appears.

Tip: Another way to do this is to click the Section Expert button on the Standard toolbar.

2 In the Sections area, click the section you want to suppress.

3 On the **Common** tab, select the **Suppress** (**No Drill-Down**) check box.

4 Click **OK** to return to your report.

The blank section will no longer be printed.

Working with conditional formatting

Conditional formatting is formatting that applies only under certain conditions. For example, in a report 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
- background color to appear on every other line.

Crystal Reports makes it easy to apply conditional formatting in these and hundreds of other situations.

With absolute formatting, you follow 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 these conditions using simple formulas. For more information on creating formulas using Crystal syntax, see "Crystal syntax fundamentals" on page 440. Or, for more information on creating formulas using Basic syntax, see "Basic syntax fundamentals" on page 401.

When a conditional formatting formula is set up, the formula overrides any fixed settings you have made in the Format Editor. For example, if you select the *Suppress* option, then set up a conditional formula for the *Suppress* option, the property will still apply only if the condition in the formula is met.

Crystal Reports enables you to set both on and off properties and set attribute properties conditionally. However, 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.

Crystal syntax example

```
condition
```

Basic syntax example

```
formula = 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 is 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, then it applies the red attribute; if it is not, then it applies the black attribute.

Use an If-Then-Else formula for this kind of conditional formatting. For Crystal syntax, see "If expressions" on page 465. Or, for Basic syntax, see "If statements" on page 427.

Crystal syntax example

```
If Condition A Then
    crRed
Else
    crBlack
```

Basic syntax example

```
If Condition A Then
    formula = crRed
Else
    formula = crBlack
End If
```

When conditional attribute properties are set up, Crystal Reports loads a selection of attributes into the Functions list in the Formula Editor. Double-click any of these attributes to add them to a formula. For example, if you are setting horizontal alignment conditionally, the Functions list contains attributes such as DefaultHorAligned, LeftAligned, and Justified. If you are setting borders conditionally, the Functions list contains attributes such as NoLine, SingleLine, and DashedLine.

Note: Always include the Else keyword in conditional formulas; otherwise, values that don't meet the If condition may not retain their original format. To retain the original format of values that don't meet your If condition, use the DefaultAttribute function.

Crystal syntax example

```
If Condition A Then
    crRed
Else
    DefaultAttribute
```

Basic syntax example

```
If Condition A Then
    formula = crRed
Else
    formula = DefaultAttribute
End If
```

You can take this kind of property one step further. You can specify a list of conditions and a property for each; you are not limited to two 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, the program should apply the Country A attribute; if it is from Country B, the Country B attribute; if it is from Country C, the Country C attribute, and so on.

With more than two alternatives, use this kind of formula:

Crystal syntax example

```
If Condition A Then
    crRed
Else If Condition B Then
    crBlack
Else If Condition C Then
    crGreen
Else
    crBlue
```

Basic syntax example

```
If Condition A Then
    formula = crRed
ElseIf Condition B Then
    formula = crBlack
ElseIf Condition C Then
    formula = crGreen
Else
    formula = crBlue
End If
```

Use a multi-condition If-Then-Else formula for this kind of conditional formatting.

Changing conditional fonts

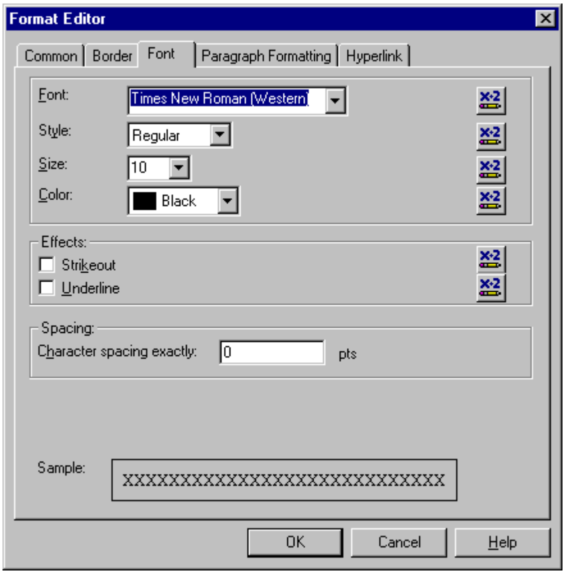
For memo or string fields that are based on conditions such as a parameter value, you can change the font, font style, size, and color for these fields using the Format Editor.

To change conditional fonts

1 Right-click the field you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Font**.

The Format Editor dialog box appears with the Font tab open.



Font: Times New Roman (Western)

Style: Regular

Size: 10

Color: Black

Effects:

□ Strikeout

□ Underline

Spacing:

Character spacing exactly: 0 pts

Sample: XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

3 To change any of the font options, click the appropriate **Formula** button, located on the right side of the dialog box.

4 In the Format Formula Editor dialog box, you can specify that conditional fonts will change only when certain conditions are met.

5 Click **Save**.

Note:

- If there is an error in the formula, a message box appears, asking whether to exit without examining the error. If you click No, a second message box will be displayed, detailing the error.
- If there is no error in the formula, you are returned to the Format Editor. Note that the Formula button has changed. This indicates that a formula has been entered for that property.

6 Click **OK** to return to your report.

Creating footers after the first page

You may choose to print a page footer on all pages except the first page. You can do this by formatting the Page Footer section conditionally, using an on or off property.

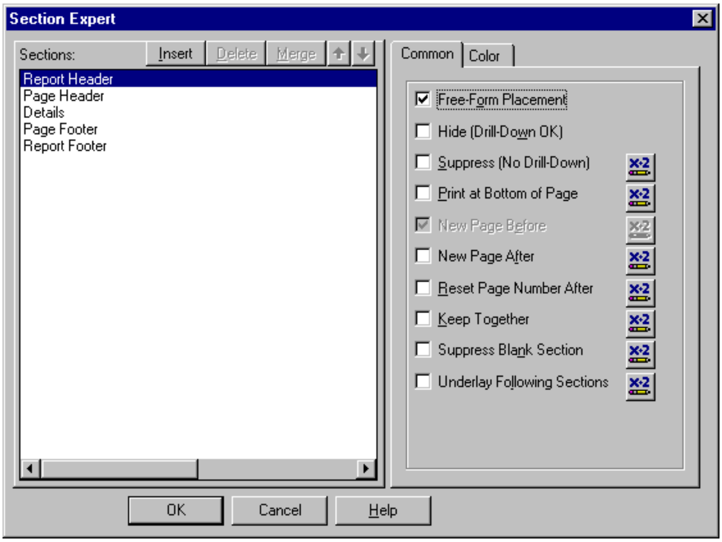
To create footers after the first page

1 Place the field you want displayed as a page footer in the Page Footer section of the report.

2 On the **Format** menu, click **Section**.

Tip: Another way to do this is to click the Section Expert button on the Standard toolbar.

The Section Expert dialog box appears.



Report Header
Page Header
Details
Page Footer
Report Footer

☑ Free-Form Placement
☐ Hide (Drill-Down OK)
☐ Suppress (No Drill-Down)
☐ Print at Bottom of Page
☑ New Page Before
☐ New Page After
☐ Reset Page Number After
☐ Keep Together
☐ Suppress Blank Section
☐ Underlay Following Sections

3 In the Sections area, click **Page Footer**.

4 To open the Format Formula Editor dialog box, click the **Formula** button, located to the right of the **Suppress** (**No Drill-Down**) check box.

5 Enter the following formula in the Format Formula Editor:

Crystal syntax example:

```
PageNumber = 1
```

Basic syntax example:

```
formula = PageNumber = 1
```

This formula suppresses the page footer on the first page, but not on any of the other pages.

6 Click **Save**.

   Note:

   - If there is an error in the formula, a message box appears, asking whether to exit without examining the error. If you click No, a second message box will be displayed, detailing the error.
   - If there is no error in the formula, you are returned to the Section Expert. Note that the Formula button has changed. This indicates that a formula has been entered for that property.

7 On the Standard toolbar, click **Preview** to preview the report and ensure that the page footer appears on all pages but the first.

Note:

- if you have a multi-line page footer and have inserted the lines into separate Page Footer sections, you will need to suppress each section conditionally, using the formula above
- to create a page header that appears on all pages but the first, place the 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.

Using the Highlighting Expert

Crystal Reports includes the Highlighting Expert that allows you to efficiently apply conditional formatting to currency and number fields. The Highlighting Expert is most commonly used for highlighting field values that are in some way distinguished from the other values in the report. For example, if you wanted to highlight the {customer.LAST YEAR'S SALES} field in red whenever the sales are greater than $25,000, you could specify in the Highlighting Expert that when the value is greater than $25,000, a red background color is applied to the field. You can also use the dialog box to set conditional font color and border style.

Think of the Highlighting Expert as an alternative to the Formula Editor. The Highlighting Expert essentially runs the following equation: `If Condition is True, Then Apply These Formatting Specifications`. For this purpose, the dialog box is divided into two areas: the Item list area which displays the formula, and the Item editor area which allows you to set the formula. The Item editor includes the Sample field that lets you view the formatting specifications being applied.

Some of the benefits to using the Highlighting Expert include:

- using the Highlighting Expert for numeric/currency fields in Cross-Tabs
- the ability to undo highlighting.

Setting highlighting priorities

The Priority buttons in the Item list area of the Highlighting Expert allow you to set the priorities for your formulas. This is useful when you have two or more formulas that could offer conflicting results in some situations.

For example, suppose you are highlighting the Unit Price field on the report. You assign to this field a highlighting formula that shows a yellow background when a unit price is greater than $100. Imagine then, that on this same report, you create another highlighting formula that shows a red background when a unit price is greater than $200. Considering that 100 is a subset of 200, you could have Unit Price fields with yellow backgrounds when, in fact, those fields should have red backgrounds. In other words, a unit price of $300 could receive either a red or a yellow background, depending on which formula has been assigned priority.

To set the priority for a formula

1 On the **Format** menu, click **Highlighting Expert**.

2 Highlight the formula.

3 In the Item list area, click the **Priority** arrows to move the selected formula to a position above or below the formula(s).

Note: A formula has priority over another formula when it is higher in the Items list area.

4 Click **OK**.

5 Click the **Preview** tab, or refresh the report to see the highlighting changes.

Note: The Remove and Remove All buttons in the Item list area can be used to delete formulas.

Conditionally formatting fields using the Highlighting Expert

Using the Highlighting Expert, you can conditionally format only currency and number fields.

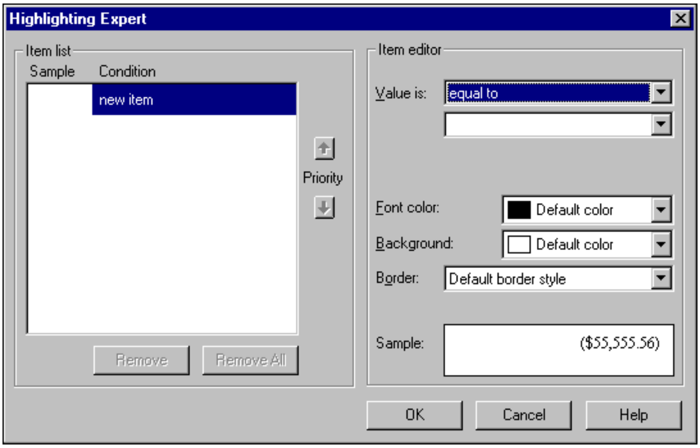
To conditionally format fields using the Highlighting Expert

1 Right-click the currency or number field you want to format to bring up the shortcut menu.

2 On the shortcut menu, click **Highlighting Expert**.

Tip: Another way to do this is to click the Highlighting button on the Formatting toolbar.

The Highlighting Expert dialog box appears.



Value is: equal to

Font color: Default color

Background: Default color

Border: Default border style

Sample: ($55,555.56)

3 In the Item editor area, select a comparison from the **Value is** list, and enter a numeric value in the adjacent box.

4 In the **Font color**, **Background**, and **Border** lists, specify your formatting changes to the selected field.

5 Repeat steps 3 and 4 if you want multiple highlighting settings for the selected field.

6 Click the **Priority** arrows to specify the priority for each additional setting you apply. See "Setting highlighting priorities" on page 243.

7 Click **OK** to return to your report.

Undo/Redo activities

Crystal Reports includes multiple levels of undo. With multiple levels of undo, you can undo any number of changes to an object, in reverse order, 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 the latest change.

The Undo and Redo buttons have lists that allow you to undo or redo a number of changes at one time.

- To undo an action, click Undo on the Standard toolbar.

  The first time the button is clicked, it reverses the most recent change made to the report. Each additional time the button is clicked, it reverses the next most recent change.

To undo several actions at once, click the arrow button to display the list of actions. Select the series of actions you wish to undo.

- To redo a change after you have undone it, click Redo on the Standard toolbar.

  The program disables the Undo button and the Undo/Redo commands whenever there is nothing to undo/redo or when you have made a change that cannot be reversed.

  To redo several actions at once, click the arrow button to display the list of actions. Select the series of actions you wish to redo.

Note: You can only undo or redo actions in order from the most recent backward. You cannot undo an action without undoing more recent actions.

Charting 13

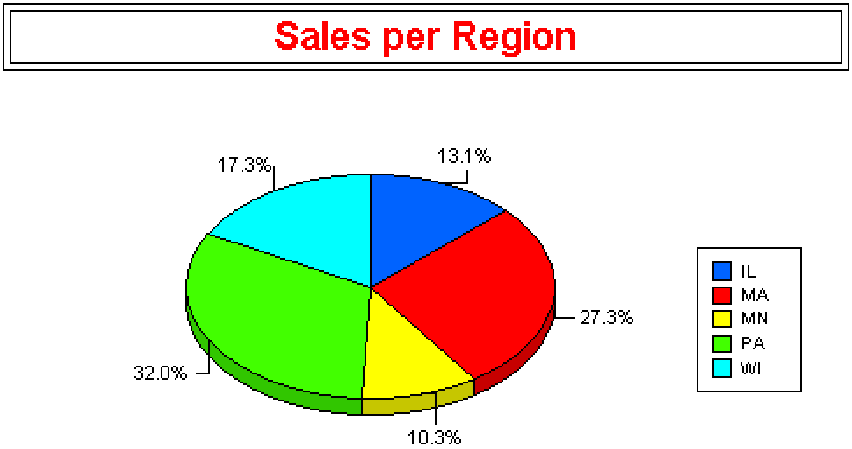
Crystal Reports enables you to present summarized data in colorful, easy-to-read charts. This chapter demonstrates how to create charts and how to use them in reports to make report data more meaningful and easier to understand. You can choose from a number of chart layouts and types, as well as drill down to see the details behind the graphical summaries and format chart objects.

Charting concepts

Charting overview

Crystal Reports enables you to include sophisticated, colorful charts in your reports. You can use charts any time you want to improve the usefulness of a report.

For example, if you have a sales report grouped by Region with a subtotal of Last Year's Sales for each region, you can quickly create a chart that will display Sales per Region.



You can chart on the following:

- summary and subtotal fields
- detail, formula, and Running Total fields
- Cross-Tab summaries
- OLAP data.

You will typically chart on summary and subtotal information at the group level. However, depending on the type of data you are working with, you can create an Advanced, Cross-Tab, or OLAP grid chart for your report.

Chart layouts

The Chart Expert provides four layouts that correspond to certain sets of data.

You can create charts with any of the following layouts, and depending on the data you are using, you can change the chart from one layout to another.

Advanced

Use the Advanced layout when you have multiple chart values or when you do not have any group or summary fields in the report.

The Advanced chart layout supports one or two condition fields: with these condition fields, you can create a 2-D or 3-D chart. Other specific functions with the Advanced layout include:

- values can be grouped in ascending, descending, or specified order, as well as by Top N or Sort totals
- values can be plotted for each record
- values can be plotted as a grand total for all records
- charts can be based on formula and Running Total fields.

Group

The Group layout is a simplified layout in which you show a summary on change of field for topics such as Country.

Note: In order to create a chart using the Group layout, you must have at least one group and at least one summary field for that group.

Cross-Tab

Use the Cross-Tab layout to chart on a Cross-Tab object. A Cross-Tab chart uses the fields in the cross-tab for its condition and summary fields.

OLAP

Use the OLAP layout to chart on an OLAP grid. An OLAP chart uses the fields in the OLAP grid for its condition and summary fields.

Chart types

Different sets of data are particularly suited to a certain chart type. The following is an overview of the main chart types and their most common uses.

Bar

A bar chart (also known as a column chart) displays or compares several sets of data. Two useful bar charts are the Side-by-Side bar chart and the Stacked bar chart.

- Side-by-Side bar chart
  A Side-by-Side bar chart displays data as a series of vertical bars. This type of chart is best suited for showing data for several sets over a period of time (for example, last year's sales figures for AZ, CA, OR, and WA).
- Stacked bar chart
  A Stacked bar chart displays data as a series of vertical bars. This type of chart is best suited for representing three series of data, each series represented by a color stacked in a single bar (for example, sales for 1997, 1998, and 1999).

Line

A line chart displays data as a series of points connected by a line. This type of chart is best suited for showing data for a large number of groups (for example, total sales over the past several years).

Area

An area chart displays data as areas filled with color or patterns. This type of chart is best suited for showing data for a limited number of groups (for example, percentage of total sales for AZ, CA, OR, and WA).

Pie

A pie chart displays data as a pie, split and filled with color or patterns. Pie charts are typically used for one group of data (for example, the percentage of sales for the entire inventory); however, you have the option to choose multiple pie charts for multiple groups of data.

Doughnut

A doughnut chart is similar to a pie chart, displaying data as sections of a circle or doughnut. If, for example, you charted sales by region on a particular report, you would see the total number of sales (the figure) in the center of the doughnut and the regions as colored sections of the doughnut. As with the pie chart, you have the option to choose multiple doughnut charts for multiple groups of data.

3-D Riser

A 3-D Riser chart displays data in a series of 3-dimensional objects, lined up side-by-side, in a 3-dimensional plane. The 3-D Riser chart shows the extremes in your report data. For example, the differences between sales by customer by country are visually dynamic when presented in this chart.

3-D Surface

3-D Surface charts present a topographic view of multiple sets of data. If, for example, you need a chart to show the number of sales by customer by country, in a visually dynamic and relational format, you might consider using the 3-D Surface chart.

XY Scatter

An XY Scatter chart is a collective of plotted points that represent specific data in a pool of information. The XY Scatter chart allows the user to consider a larger scope of data for the purpose of determining trends. For example, if you input customer information, including sales, products, countries, months, and years, you would have a collective of plotted points that represents the pool of customer information. Viewing all of this data on an XY Scatter chart would allow you to speculate as to why certain products were selling better than others or why certain regions were purchasing more than others.

Radar

A radar chart positions group data, such as countries or customers, at the perimeter of the radar. The radar chart then places numeric values, increasing in value, from the center of the radar to the perimeter. In this way, the user can determine, at a glance, how specific group data relates to the whole of the group data.

Bubble

A bubble chart displays data as a series of bubbles, where the size of the bubble is proportional to the amount of data. A bubble chart would be very effective with the number of products sold in a certain region; the larger the bubble, the greater the number of products sold in that region.

Stock

A stock chart presents high and low values for data. It is useful for monitoring financial or sales activities.

Where to place a chart

The placement of a chart determines which data is displayed and where it is printed. For example, if you place a chart in the Report Header section, the chart includes data for the entire report. If you place it in a Group Header or Group Footer section, it displays group specific data.

Note: If your report contains subreports, you can place charts in those subreports as well. See "Subreports" on page 493.

Drill-down with charts

Not only is charting a means of presenting data — it is also an analysis tool. You can activate the drill-down mode by right-clicking the group chart on the Preview tab and click Drill-down from the shortcut menu. When you are in drill-down mode, move the pointer over a section of the group chart on the Preview tab, so that the pointer becomes a Drill-down cursor, then double-click to view the underlying details for that section of the chart.

Drill-down with legends

If the chart consists of one or more group fields, you can use the chart legend to drill down on individual groups. Double-click the drill-down cursor on the markers and text in the legend to view the details about that section of the chart.

Creating charts

Charting on details or formula fields (Advanced layout)

The Advanced layout allows you to create a chart based on specific values. Since charts are a good way to display summarized information, they are often based on a summary field in your report. With an Advanced layout, you can create a chart without the need for a summary field by using values that appear in the Details section of your report.

To create a chart based on the Advanced layout, you must specify two things:

- conditions (there can be two)
- values (there can be multiple values).

Condition

The condition is used to indicate when to plot the point. For example, a chart showing last year's sales for your customers uses the Customer Name field as the condition. Each time the condition changes (the customer name changes), a point is plotted.

You also have the option of plotting a point for each record, or plotting one point for all records.

Value

The Advanced chart uses the value to indicate what information is plotted as the points on your chart. For example, to create a chart showing last year's sales for your customers, the Last Year's Sales field would be the value.

To chart on a details or formula field

1 On the **Insert** menu, click **Chart**.

Tip: Another way to do this is to click the Insert Chart button on the Standard toolbar.

The Chart Expert dialog box appears.

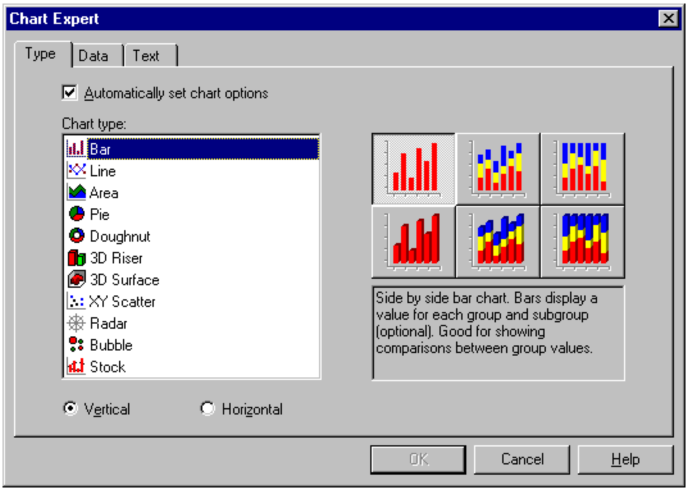


Chart Expert

Type | Data | Text

☑ Automatically set chart options

Chart type:

- Bar
- Line
- Area
- Pie
- Doughnut
- 3D Riser
- 3D Surface
- XY Scatter
- Radar
- Bubble
- Stock

Side by side bar chart. Bars display a value for each group and subgroup (optional). Good for showing comparisons between group values.

● Vertical ○ Horizontal

2 On the **Type** tab, in the **Chart type** list, select a chart type. Then click the chart subtype that best illustrates your data. See "Chart types" on page 249.

3 Select **Automatically set chart options** if you want the Chart Expert to use the default options on the Axes and Options tabs.

Note: If you want to select options on the Axes and Options tabs manually, ensure that "Automatically set chart options" is not selected.

4 Click the **Data** tab.

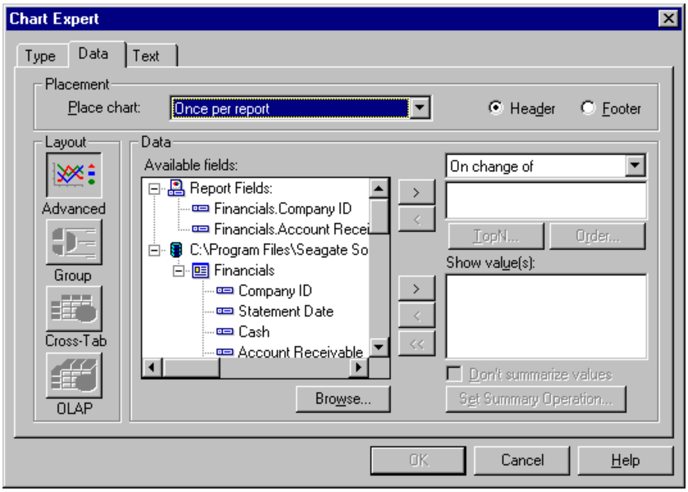


Chart Expert

Type | Data | Text

Placement

Place chart: Once per report ● Header ○ Footer

Layout

Advanced

Group

Cross-Tab

OLAP

Data

Available fields:

- Report Fields:
  - Financials.Company ID
  - Financials.Account Recei
- C:\Program Files\Seagate So
  - Financials
    - Company ID
    - Statement Date
    - Cash
    - Account Receivable

On change of

Show value(s):

□ Don't summarize values

5 In the Layout area, click **Advanced**, if it is not already selected.

6. In the Placement area, in the **Place chart** list, specify how often your chart appears on the report, then click **Header** or **Footer** to specify where to place your chart.
7. In the Data area, specify the database fields you want to use as conditions.

   You can select **On change of** from the list, then add up to two database fields in the box underneath the list.

   The arrow buttons on the Chart Expert dialog box allow you to move fields from one list to the other. Single arrows move only the selected field; double arrows move all fields at the same time.
8. Add the database fields you want to use as values to the **Show value(s)** list.
9. If you do not want Crystal Reports to automatically summarize the chart values, select the **Don't summarize values** check box.
10. If the **Axes** and **Options** tabs appear, you can customize some of the chart's properties, such as the scaling of the axes, the legend, and the data points.
11. Click the **Text** tab.
12. Accept the default title information or add new titles to your chart.
13. Click **OK**.

Your chart is placed in the Header or Footer section of the report, depending on your selection in Step 7. You may have to refresh your report to preview the finished chart.

Note: When your chart is inserted, it may cover a portion of the report. Move and resize the chart so that it fits properly within the report.

Charting on summary or subtotal fields (Group layout)

Many of the charts you create are based on summary or subtotals within your report. In order to create these charts, you must have a summary or subtotal already inserted into your report. For more information on inserting summaries and subtotals, see "Summarizing grouped data" on page 156 and "Subtotaling" on page 160.

To chart on a summary or subtotal field

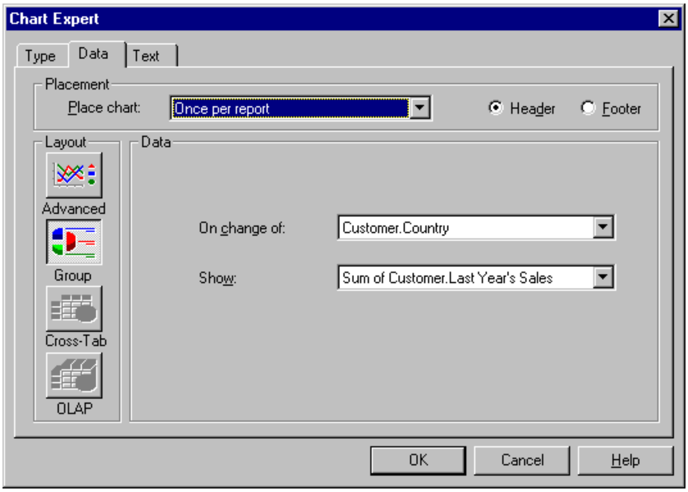
1. On the **Insert** menu, click **Chart**.

   The Chart Expert dialog box appears.

   Tip: Another way to do this is to click the Insert Chart button on the Standard toolbar.
2. On the **Type** tab, in the **Chart type** list, select a chart type. Then click the chart subtype that best illustrates your data. See "Chart types" on page 249.
3. Select **Automatically set chart options** if you want the Chart Expert to use the default options on the Axes and Options tabs.

Note: If you want to select options on the Axes and Options tabs manually, ensure that "Automatically set chart options" is not selected.

4 Click the **Data** tab.



Place chart: Once per report

● Header ○ Footer

On change of: Customer.Country

Show: Sum of Customer.Last Year's Sales

5 In the Layout area, click **Group**, if it is not already selected.

Note: When summarizing or subtotaling a field, the data is automatically grouped. See "Sorting, Grouping, and Totaling" on page 137.

6 In the Placement area, in the **Place chart** list, specify how often your chart appears on the report, then click **Header** or **Footer** to specify where to place your chart.

Note: The "Place chart" list includes all groups in your report that have summary fields except for the inner-most group.

7 In the Data area, in the **On change of** list, click the group field you want to base your chart on; then, in the **Show** list, click the summary field you want to display on your chart.

8 If the **Axes** and **Options** tabs appear, you can customize some of the chart's properties, such as the scaling of the axes, the legend, and the data points.

9 Click the **Text** tab.

10 Accept the default title information or add new titles to your chart.

11 Click **OK**.

Your chart is placed in the Header or Footer section of the report, depending on your selection in Step 6. You may have to refresh your report to preview the finished chart.

Note: When your chart is inserted, it may cover a portion of the report. Move and resize the chart so that it fits properly within the report.

Charting on Cross-Tab summaries (Cross-Tab layout)

Crystal Reports allows you to include a chart based on summary values in your Cross-Tab report. For example, with a Cross-Tab that shows the amount of a certain product sold in each region of the United States, you may want to include a chart showing the percentage of total sales provided by each region for that product.

To create a Cross-Tab chart, you must first have a Cross-Tab in your report. For more information, see "Cross-Tab Objects" on page 299.

To chart on a Cross-Tab summary

1 Select the Cross-Tab on which you want to chart.

2 On the **Insert** menu, click **Chart**. The Chart Expert dialog box appears.

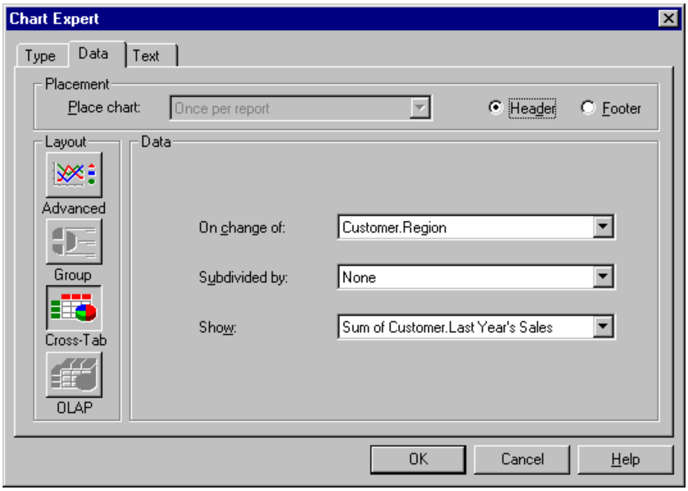
Tip: Another way to do this is to click the Insert Chart button on the Standard toolbar.

3 On the **Type** tab, in the **Chart type** list, select a chart type. Then click the chart subtype that best illustrates your data. See "Chart types" on page 249.

4 Select **Automatically set chart options** if you want the Chart Expert to use the default options on the Axes and Options tabs.

Note: If you want to select options on the Axes and Options tabs manually, ensure that "Automatically set chart options" is not selected.

5 Click the **Data** tab.



Place chart: Once per report
● Header ○ Footer

On change of: Customer.Region
Subdivided by: None
Show: Sum of Customer.Last Year's Sales

6 In the Layout area, click **Cross-Tab**, if it is not already selected.

7. In the Placement area, click **Header** or **Footer** to specify where to place your chart.

   Note: How often your chart appears on the report depends on where the Cross-Tab summary field has been placed.

8. In the Data area, in the **On change of** list, click the group field you want to base your chart on.
9. If necessary, in the **Subdivided by** list, click a secondary row or column you want to base your chart on.
10. In the **Show** list, click the summary field you want to display on your chart.
11. If the **Axes** and **Options** tabs appear, you can customize some of the chart's properties, such as the scaling of the axes, the legend, and the data points.
12. Click the **Text** tab.
13. Accept the default title information or add new titles to your chart.
14. Click **OK**.

Your chart is placed in the Header or Footer section of the report, depending on your selection in Step 7. You may have to refresh your report to preview the finished chart.

Note: When your chart is inserted, it may cover a portion of the report. Move and resize the chart so that it fits properly within the report.

Charting on an OLAP cube (OLAP layout)

The OLAP layout lets you chart on an OLAP grid. In order to create an OLAP chart, you must first have an OLAP grid in your report. For more information, see "Creating an OLAP report" on page 318.

To chart on an OLAP cube

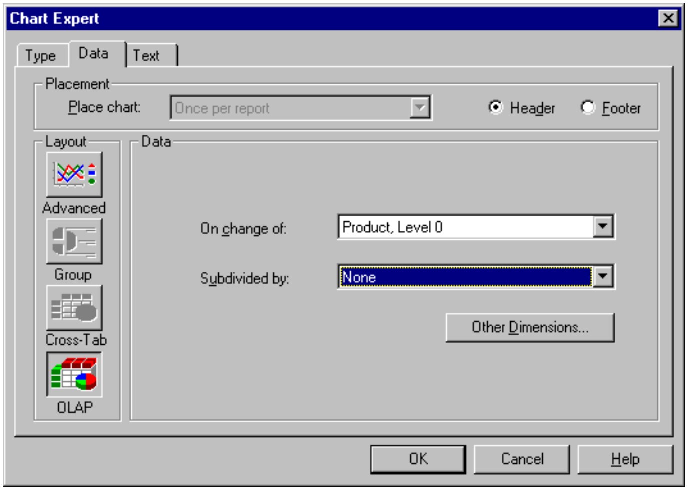
1. Select the OLAP grid on which you want to chart.
2. On the **Insert** menu, click **Chart.** The Chart Expert dialog box appears.

   Tip: Another way to do this is to click the Insert Chart button on the Standard toolbar.

3. On the **Type** tab, in the **Chart type** list, select a chart type. Then click the chart subtype that best illustrates your data. See "Chart types" on page 249.
4. Select **Automatically set chart options** if you want the Chart Expert to use the default options on the Axes and Options tabs.

   Note: If you want to select options on the Axes and Options tabs manually, ensure that "Automatically set chart options" is not selected.

5 Click the **Data** tab.



On change of: Product, Level 0

Subdivided by: None

6 In the Layout area, click the **OLAP** button, if it is not already selected.

7 In the Placement area, click **Header** or **Footer** to specify where to place your chart.

Note: The frequency with which your chart appears in the report depends on the report section that the OLAP grid has been placed in.

8 In the Data area, in the **On change of** list, click the field you want to base your chart on.

9 If necessary, in the **Subdivided by** list, click a secondary row or column you want to base your chart on.

Note: Be sure that the chart type selected in Step 3 supports a secondary charting field.

10 If the **Axes** and **Options** tabs appear, you can customize some of the chart's properties, such as the scaling of the axes, the legend, and the data points.

11 Click the **Text** tab.

Accept the default title information or add new titles to your chart.

12 Click **OK**.

Your chart is placed in the Header or Footer section of the report, depending on your selection in Step 7. You may have to refresh your report to preview the finished chart.

Note: When your chart is inserted, it may cover a portion of the report. Move and resize the chart so that it fits properly within the report.

Working with charts

Editing charts using the Chart Expert

Once you have created a chart, you may want to add a new title, headings, or a legend, change fonts, or even change the type of chart. Modifying charts is easy with the use of the Chart Expert.

To edit a chart using the Chart Expert

1. Right-click the chart to bring up the shortcut menu.
2. On the shortcut menu, click **Chart Expert**.
3. In the Chart Expert dialog box, make the desired changes.
4. Click **OK**.

Editing charts on the Analyzer tab

Crystal Reports provides the Analyzer tab for advanced charting. Use this tab to organize your charts for more efficient analysis.

To launch the Analyzer tab

1. On the **Preview** tab, right-click the chart to bring up the shortcut menu.
2. On the shortcut menu, click **Chart Analyzer**.

   Crystal Reports opens the Analyzer tab.

When viewing a chart on the Analyzer tab, you can perform the following operations by using the commands on the shortcut menu:

- zoom in and out on bar and line charts
- save the current template to a file
- apply a new template to the chart
- change the template specifications of the chart
- change the chart titles
- change the numeric axis grids and scales of the chart
- perform any additional formatting operations specific to the chart type
- auto-arrange the appearance of the chart.

Many of the operations listed here can be found in the Advanced Chart Settings menu on the Design, Preview, and Analyzer tabs. If you select an option from this menu on either the Design or Preview tab, your changes are applied to all instances of the chart on the report. Or, if you select an Advanced Chart Settings option on the Analyzer tab, you set the properties of an individual instance of the chart independently from the rest of the report.

With the Analyzer tab, up to two chart objects can be viewed simultaneously. In this case, the Analyzer tab becomes a split-screen tab, with each chart appearing in a separate section. To resize a section, move the cursor over the section boundary until the cursor becomes a Section Sizing cursor, and then drag the boundary line to expand or reduce the size of the section, as needed.

The Analyzer tab can also be used to work with chart objects that have been placed within a subreport. The program opens a separate Analyzer tab for each subreport that contains a chart object you are analyzing. You can view up to two chart objects from the same subreport simultaneously in the Analyzer tab.

Note: Changes performed in the Analyzer tab are applied to the currently selected chart or charts. However, you can apply changes to all instances of that chart or charts by selecting the Apply Changes option. You can have two charts selected in the analyzer at one time. When you try to select a third chart and bring it into the analyzer, you will be asked to discard one of the previously selected charts.

Formatting chart objects on the Analyzer tab

When viewing a chart on the Analyzer tab, you can move or resize the following chart objects by using the drag procedure:

- chart
- title
- subtitle
- footnote
- legend
- X axis title
- Y1 axis title
- Y2 axis title.

Using the zooming features with bar and line charts

On the Preview and Analyzer tabs, you can find commands for zooming bar charts and line charts within your report. You have the ability to zoom in and out on these chart types at any time, with each time being referred to as *instance-specific*. If you decide to save the instance of the chart that has been zoomed in or out, you must save the data with the report.

To zoom in and out on a bar or line chart

1 On the **Preview** or **Analyzer** tab, right-click the bar or line chart to bring up the shortcut menu.

2 On the shortcut menu, click **Zoom In**.

3 With the Zoom In cursor, click the chart.
The chart zooms in one level of magnification.

4 Click the chart again to zoom in further.

Note: If you want to view a section of the chart in greater detail (e.g. a riser on your bar chart), you can drag the Zoom In cursor around the section to enclose it within the tracking rectangle.

5 To zoom out on a chart, right-click the chart to bring up the shortcut menu.

6 On the shortcut menu, click **Zoom Out**.

7 With the Zoom Out cursor, click the chart.
The chart zooms out one level of magnification.

8 Click the chart again to zoom out further.

Auto-arranging charts

If you move or resize chart objects on the Analyzer tab, select the Auto-Arrange Chart feature to reset the chart.

To auto-arrange a chart

1 On the **Analyzer** tab, right-click the chart to bring up the shortcut menu.

2 On the shortcut menu, click **Auto-Arrange Chart**.
Crystal Reports resets the chart on the Analyzer tab.

Changing chart size and position

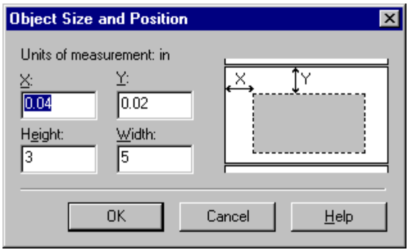
You can use the Object Size and Position dialog box to specify the height, width, and position of the chart, in inches.

To change chart size and position

1 On the **Design** or **Preview** tab, right-click the chart to bring up the shortcut menu.

2 On the shortcut menu, click **Object Size and Position**.

The Object Size and Position dialog box appears.



Object Size and Position

Units of measurement: in

X: 0.04

Y: 0.02

Height: 3

Width: 5

3 In the **X** text box, enter the distance in inches from the left edge of the chart object to the left margin of the report section.

The default distance is 0.04 inches.

4 In the **Y** text box, enter the distance in inches from the upper edge of the chart object to the upper margin of the report section.

The default distance is 0.02 inches.

5 In the **Height** text box, enter the vertical height of the chart, in inches.

6 In the **Width** text box, enter the horizontal height of the chart, in inches.

7 Click **OK** to save your changes.

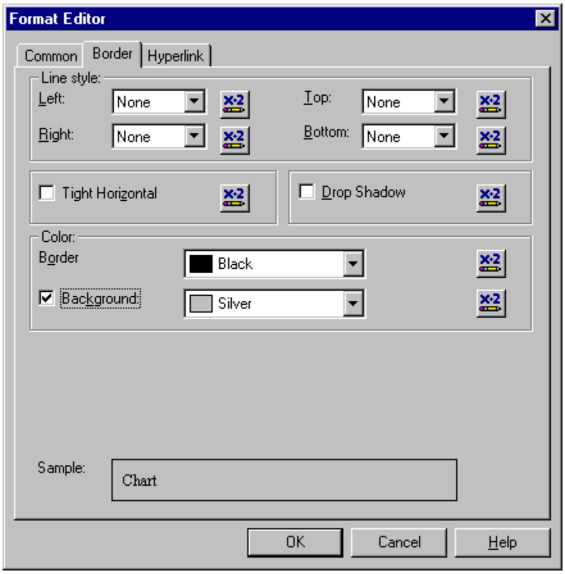
Crystal Reports returns you to the report and implements your changes.

Changing the border of a chart

1 On the **Design** or **Preview** tab, right-click the chart to bring up the shortcut menu.

2 On the shortcut menu, click **Border and Colors**.

The Format Editor dialog box appears with the Border tab open.



Line style:

Left: None

Top: None

Right: None

Bottom: None

□ Tight Horizontal

□ Drop Shadow

Color:

Border ■ Black

☑ Background: □ Silver

Sample: Chart

3 Change the line style, color, background color, and add or remove a drop shadow from the chart border.

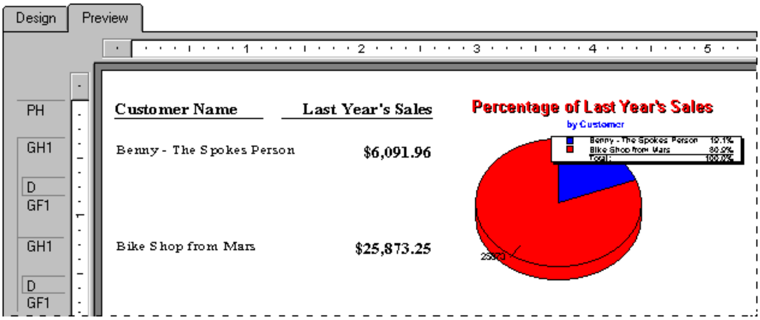
4 Click **OK** to save your changes.

Crystal Reports returns you to the report and implements your changes.

Using the underlay feature with charts

Since charts can print only in certain sections of your report, the underlay feature gives you more control in the overall look of your report. Instead of having a chart print ahead of the data it represents, it can appear next to the data for a more comprehensible report.

This is how your report looks when you underlay a chart with report data.


Design
Preview
PH
GH1
D
GF1
GH1
D
GF1
Percentage of Last Year's Sales
by Customer
Benny - The Spokes Person 19.1%
Bike Shop from Mars 80.9%
Total: 100.0%
25873

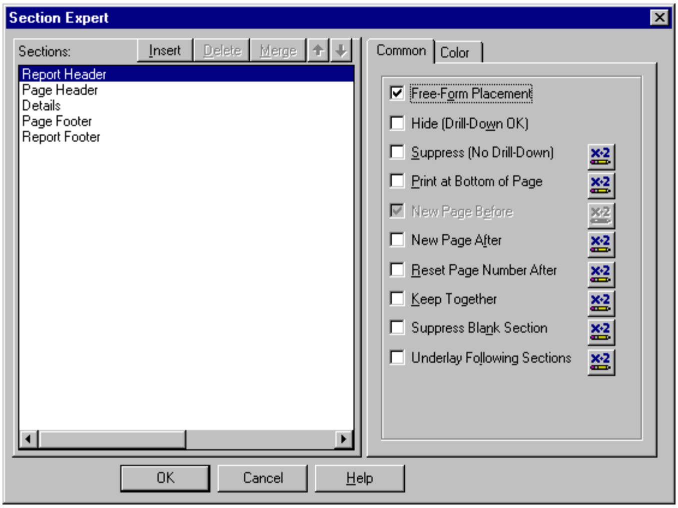
Customer Name	Last Year's Sales
Benny - The Spokes Person	$6,091.96
Bike Shop from Mars	$25,873.25

To underlay a chart

1 Create your chart and place it in the Report Header section. See "Creating charts" on page 252.

2 On the **Insert** menu, click **Section**. The Section Expert dialog box appears.

Tip: Another way to do this is to click the Section Expert button on the Standard toolbar.



Sections:

- Report Header
- Page Header
- Details
- Page Footer
- Report Footer

☑ Free-Form Placement
☐ Hide (Drill-Down OK)
☐ Suppress (No Drill-Down)
☐ Print at Bottom of Page
☑ New Page Before
☐ New Page After
☐ Reset Page Number After
☐ Keep Together
☐ Suppress Blank Section
☐ Underlay Following Sections

3 In the Sections area, click **Report Header**, then select the **Underlay Following Sections** check box.

4 Click **OK**.

Crystal Reports returns you to the report. Your chart will now underlay the sections below it.

5 If necessary, move or resize the chart.

Mapping 14

Crystal Reports enables you to include maps with reports made up of geographic data. This chapter explains how to use maps in reports to make report data more meaningful and easier to interpret. You can customize and rearrange the appearance of a map and activate the drill-down mode to view the details behind the graphical summaries.

Mapping concepts

Mapping overview

With Crystal Reports, you can include geographic maps in reports. Maps help you analyze report data and identify trends more efficiently. For example, you could create a map that shows sales by region. You would then be able to:

- use one of the five map types to analyze the data
- use the Analyzer tab to adjust the appearance and organization of the map (allowing you to better identify trends)
- drill down on the map regions to view underlying data.

In order to place a generic, group-based map on a report, you must first have a group and a summary or subtotal field present for that group. However, the program does provide several specialized map layouts that do not require groups and summaries.

Note: When a field is summarized or subtotaled, the program automatically groups the data. For more information, see “Sorting, Grouping, and Totaling” on page 137.

Map layouts

The Map Expert provides four layouts that correspond to certain sets of data.

You can create maps with any of the following layouts, and depending on the data you are using, you can change the map from one layout to another.

Advanced

Use the Advanced layout when using multiple map values or when you do not have any groups or summaries in the report.

Group

The Group layout is a simplified layout in which you show a summary on change of a geographic field (such as Region).

Note: In order to create a map using the Group layout, you must have at least one group and at least one summary field for that group.

Cross-Tab

Use the Cross-Tab layout when mapping on a Cross-Tab object. A Cross-Tab map does not require groups or summary fields.

OLAP

Use the OLAP layout when mapping on an OLAP grid. An OLAP map does not require groups or summary fields.

Note: If there is no map associated with the data you specify, then an empty block will appear unless the section that the map is placed in has been formatted to suppress if blank.

Map types

The Map Expert also provides five basic map types, each suitable for a different strategy of data analysis. When deciding which map type best fits your report, you should consider the type of data you want to analyze. For example, if you want the map to display one data item for each geographic division (city, state, country, and so on), then you might use a Ranged, Dot Density, or Graduated map. On the other hand, if you want the map to display more than one value for each new geographic division, then you could use a Pie Chart map or a Bar Chart map. The following is an overview of the main map types and their most common uses.

Ranged

A Ranged map breaks the data into ranges, assigns a specific color to each range, then color codes each geographic area on the map to display the range. For example, you could create a map that displays Last Year's Sales by Region. If you have sales ranging from zero to 100,000, you might give the map five ranges, with equal intervals of 20,000 each. You could use shades of red (going from dark to light red) to color code each region according to those sales figures. Then you could use this map to see where sales are the highest.

With equal intervals, you might end up with all your regions ranging between zero and 20,000, except perhaps one region (for example, California) that might have exceptionally high sales (such as 98,000). This map would be a very distorted representation of the data. A more useful map would have ranges like 0-5000, 5001-10000, 10001-15000, 15001-20000, and over 20000. It is important to carefully define your ranges.

There are four different distribution methods for ranged maps.

- **Equal count**
  This option assigns intervals in such a way that the same number of regions (or as close to the same number of regions as possible) appear in each interval. In other words, this option would assign intervals so that each color on the map is assigned to the same number of regions. The numeric quantity of the summary values in each interval may or may not be equal, depending on the individual regions and their summary values.
- **Equal ranges**
  This option assigns intervals in such a way that the summary values in each interval are numerically equal. The number of regions in each interval may or may not be equal, depending on the individual regions and their summary values.

- **Natural break**
  This option assigns intervals using an algorithm that attempts to minimize the difference between the summary values and the average of the summary values for each interval.
- **Standard deviation**
  This option assigns intervals in such a way that the middle interval breaks at the mean (or average) of the data values, and the intervals above and below the middle range are one standard deviation above or below the mean. Standard deviation is the square root of the variance. It projects how various values in a set of values deviate from the mean for that set.

Dot Density

A Dot Density map displays a dot for each occurrence of a specified item. For example, you might create a United States map that shows one dot for each shipbuilder in the nation. In states like Tennessee, there would be no dots. However, in some coastal states, such as South Carolina, you might be able to count the dots on the map, since their dispersal would be fairly wide.

The purpose of a Dot Density map is to provide an overall impression of the distribution of the specified item. A Dot Density map is much like a nighttime satellite photo of the United States, where you can see the lights of all the cities. Such a map is not a very accurate means of communicating information (particularly if you have a large number of items), but it is a good way to give an overview of the distribution.

Graduated

A Graduated map is much like a Ranged map; it shows one symbol per instance of a specified item. This symbol is a circle by default, but you can choose a different symbol if you prefer. Each symbol is proportional in size to the value of the item it represents (within a range of three sizes).

A Graduated map communicates the same information as the Ranged map, but you would usually create a Ranged map for a case in which the geographic areas have distinct geographic boundaries (as in the case of Regions), while you would use a Graduated map for displaying data that is linked to points rather than precise areas (as in the case of Cities).

For example, a map of an individual region could use graduated circles to represent the sales for each office. The size of each circle would be proportional to the sales (or to the sales range) of the office it represents. On this map, an office with a sales figure of $70,000 might have a large circle, and an office with a sales figure of $20,000 might have a small circle. So, a Graduated map provides a more efficient representation of point data (e.g. Cities) than a Ranged map does, and it uses sized symbols rather than colors to distinguish variations in the values of the items it represents.

Pie Chart

A Pie Chart map displays a pie chart over each geographic area. The pie charts represent data items that make up a whole. Each slice of the pie represents an individual data item and shows that item’s percentage in the whole. For example, you could create a Pie Chart map showing heating fuel types by region. You might have four types of heating fuel (four slices in each pie): electricity, gas, wood, and solar. Each region would then have a pie chart showing the breakdown of heating fuel types within that region. Washington state would probably use a high percentage (a large slice of the pie) of electricity because of the hydropower in that region, while Idaho would probably use a high percentage (a large slice of the pie) of natural gas.

You can use this map type to compare the distribution of several items within a particular region. You can also specify that the pie charts be sized proportionately so that, as with the symbols in a Graduated map, the pie charts will appear in various sizes, depending on the underlying data values. This will allow you to compare the totals between regions.

Bar Chart

A Bar Chart map works like a Pie Chart map, but may be more useful for certain sets of data. Typically, you would use a Bar Chart map for items that do not total 100%; that is, for data items that do not make a whole, or for data items that are unrelated. For example, you could create a Bar Chart map that displays use of heating fuel by region. You might choose to analyze use of three types of fuel: electricity, gas, and solar. Each bar chart on the map could contain individual bars for each of these types. In this example, the data items (electricity, gas, and solar) do not comprise a whole. There may be other types of fuel used in these regions, such as wood, but this map only focuses on three of them. Also, the purpose of the map is to compare each region’s use of each fuel type with that of every other region. In a Pie Chart map, you could show these three fuel types as percentages of the entire fuel use within each region, and though you could compare the percentages for each region, you would probably not be able to compare the actual values for each region because each region would have the same total value (100%).

Where to place a map

When you choose where to place the map, you determine the amount of data that will be included in the map. For example, if you place the map in the Report Header section, the map includes data for the entire report. If you place it in a Group Header or Group Footer section, it displays group-specific data. This choice will also determine whether the map prints once for the entire report, or many times (once for each instance of a given group).

Note: If your report contains subreports, you can place maps in those subreports as well. See “Mapping on subreports” on page 278.

Drill-down with maps

Not only is mapping a means of presenting data — it is also an analysis tool. You can activate the drill-down mode by right-clicking the group map on the Preview tab and clicking Drill-down from the shortcut menu. When you are in drill-down mode, move the pointer over a section of the map on the Preview tab, so that the pointer becomes a Drill-down cursor, then double-click to view the underlying details for that section of the map.

Note: If you drill down on a region that has no data associated with it, you will get a message saying, "There are no detail records for that {Region Name}."

Creating maps

The process for creating a map varies depending on the data you are mapping on. The following sections detail the mapping process for each map layout.

Mapping on details fields (Advanced layout)

Using the Advanced layout, you can map on specific values rather than summaries. This layout also allows you to use multiple fields as map values. Use this layout to use multiple map values or when you do not have any groups or summaries in a report.

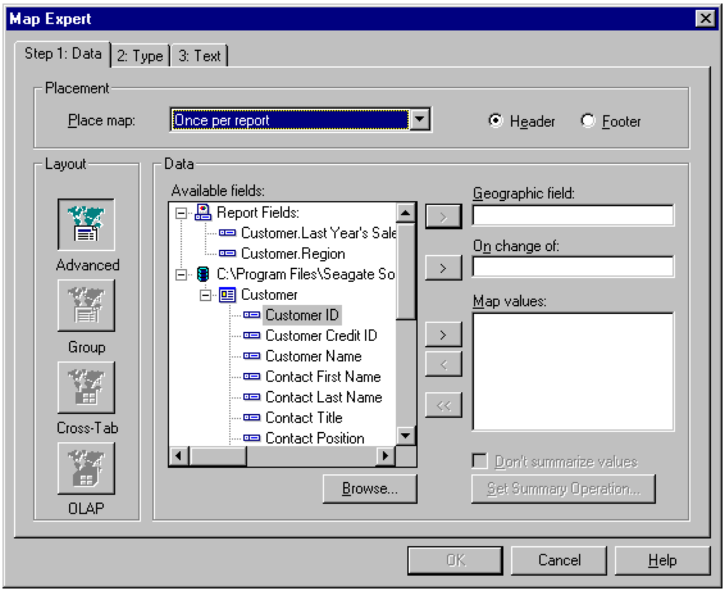
Note: You must map on a geographic field, such as Country, Region, or City. Crystal Reports cannot generate a map based on a non-geographic field.

To map on a details field

1. On the **Insert** menu, click **Map**.

   Tip: Another way to do this is to click the Insert Map button on the Standard toolbar.

The Map Expert dialog box appears.



Place map: Once per report

● Header ○ Footer

2 On the **Data** tab, in the Layout area, click **Advanced**, if it is not already selected.

3 In the Placement area, in the **Place map** list, specify how often your map appears on the report, then click **Header** or **Footer** to specify where to place your map.

4 In the Data area, add the database field you want to base your map on to the **Geographic field**.

The arrow buttons on the Map Expert dialog box allow you to move fields from one list to the other. Single arrows move only the selected field; double arrows move all fields at the same time.

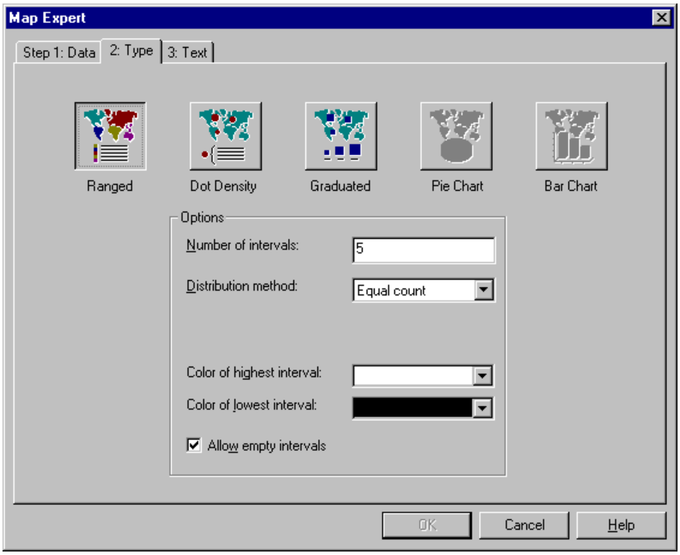
Note: You must map on a geographic field, such as City, Region, or Country; otherwise, Crystal Reports cannot generate your map.

5 Add the database field you want to appear with each change to the **On change of** field.

6 Add the database fields you want to use as values to the **Map values** list.

7 If you do not want Crystal Reports to automatically summarize the map values, select the **Don't summarize values** check box.

8 Click the **Type** tab.



Map Expert

Step 1: Data | 2: Type | 3: Text

Options

Number of intervals: 5

Distribution method: Equal count

Color of highest interval:

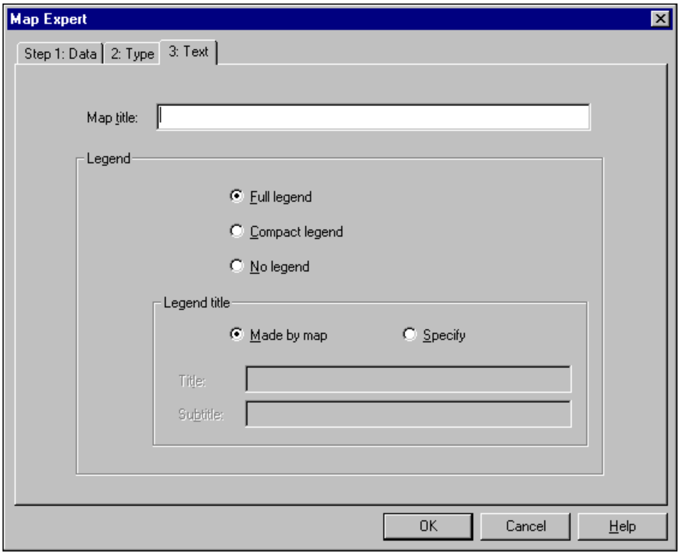
Color of lowest interval:

☑ Allow empty intervals

9 Click the map type that best illustrates your data (Ranged, Dot Density, Graduated, Pie Chart, or Bar Chart). See "Map types" on page 267.

10 In the Options area, apply formatting options to your map.

11 Click the **Text** tab.



Map title:

Legend

● Full legend

○ Compact legend

○ No legend

Legend title

● Made by map ○ Specify

Title:

Subtitle:

12 In the **Map title** field, enter a title for your map.

13 In the Legend area, you can click one of the following options:

- **Full legend** to show a detailed legend on your map
- **Compact legend** to show a condensed legend on your map
- **No legend** to exclude the legend from your map.

14 If you click **Full legend**, click **Made by map** to have Crystal Reports create a legend title based on your map, or click **Specify** to enter your own legend title and subtitle.

15 Click **OK**.

Your map is placed in the Header or Footer section of the report, depending on your selection in Step 3.

Mapping on group fields (Group layout)

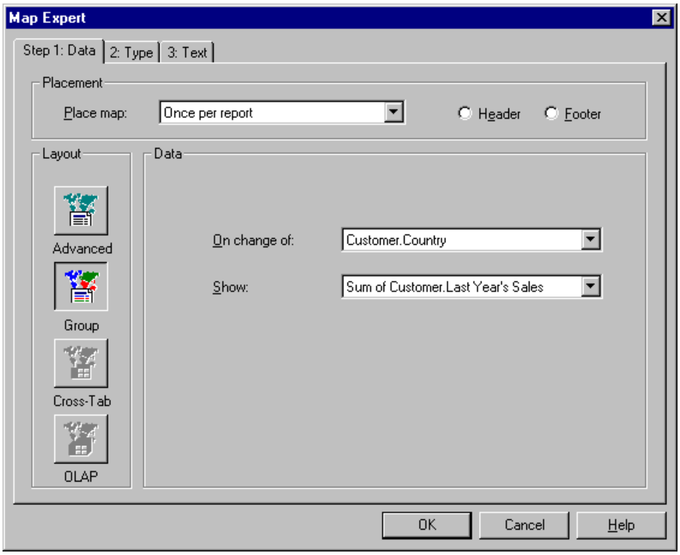
To map on a group, you can use the Group layout, in which you show a summary (such as Last Year's Sales) on change of a geographic field (such as Region). In order to create a map using the Group layout, you must have at least one group and at least one summary field for that group.

To map on a group

1 On the **Insert** menu, click **Map**.

Tip: Another way to do this is to click the Insert Map button on the Standard toolbar.

The Map Expert dialog box appears.



Place map: Once per report

○ Header ○ Footer

On change of: Customer.Country

Show: Sum of Customer.Last Year's Sales

2 On the **Data** tab, in the Layout area, click **Group**, if it is not already selected.

3 In the Placement area, in the **Place map** list, specify how often your map appears on the report, then click **Header** or **Footer** to specify where to place your map.

4 In the Data area, in the **On change of** list, click the group field you want to base your map on, then in the **Show** list, click the summary field you want to display on your map.

5. Click the **Type** tab.
6. Click the map type that best illustrates your data (Ranged, Dot Density, or Graduated). See “Map types” on page 267.
7. In the Options area, apply formatting options to your map.
8. Click the **Text** tab.
9. In the **Map title** field, enter a title for your map.
10. In the Legend area, you can click one of the following options:
    - **Full legend** to show a detailed legend on your map
    - **Compact legend** to show a condensed legend on your map
    - **No legend** to exclude the legend from your map.
11. If you click **Full legend**, click **Made by map** to have Crystal Reports create a legend title based on your map, or click **Specify** to enter your own legend title and subtitle.
12. Click **OK**.

    Your map is placed in the Header or Footer section of the report, depending on your selection in Step 3.

Mapping on Cross-Tab summaries (Cross-Tab layout)

With the Cross-Tab layout, you can map on a Cross-Tab summary field. For example, with a Cross-Tab that shows the total number of items of a product sold in each region of the United States, you may want to include a map that shows the percentage of the total sales of the product provided by each region.

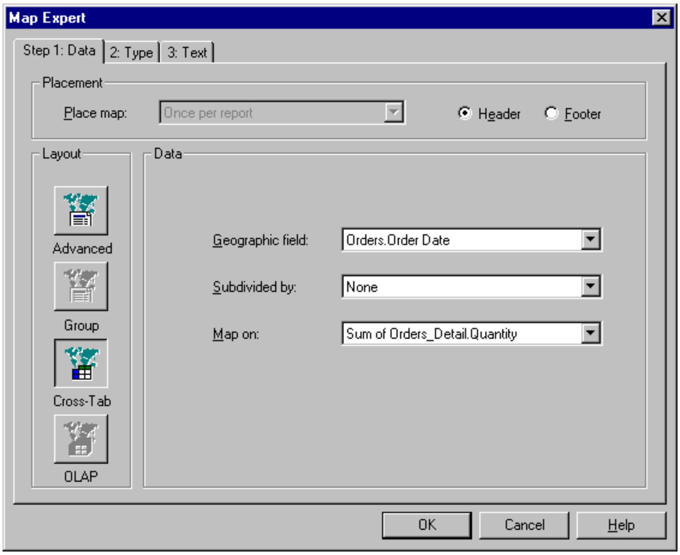
In order to create a Cross-Tab map you must first have a Cross-Tab in your report. For more information about Cross-Tabs, see “Cross-Tab Objects” on page 299.

To map on a Cross-Tab summary

1. Select the Cross-Tab summary field you want to base your map on.
2. On the **Insert** menu, click **Map**.

    Tip: Another way to do this is to click the Insert Map button on the Standard toolbar.

The Map Expert dialog box appears.



Place map: Once per report

● Header ○ Footer

Geographic field: Orders.Order Date

Subdivided by: None

Map on: Sum of Orders_Detail.Quantity

3 On the **Data** tab, in the Layout area, click **Cross-Tab**, if it is not already selected.

4 In the Placement area, click **Header** or **Footer** to specify where to place your map.

5 In the Data area, in the **Geographic field** list, click a geographic field to supply the geographic areas for your map.

6 If necessary, in the **Subdivided by** list, click another field to map on.

7 Crystal Reports uses this field to subdivide the pie or bar charts appearing on your map.

8 In the **Map on** list, click a summary field to supply the numeric data for your map.

9 Click the **Type** tab.

10 Click the map type that best illustrates your data. See "Map types" on page 267.

- If you plan to map on only one field, be sure to select **None** from the **Subdivided by** list, then choose either the Ranged, Dot Density, or Graduated map type.
- If you plan to map on two fields, be sure to select an additional field from the **Subdivided by** list, then choose either the Pie Chart or Bar Chart map type.

11 In the Options area, apply formatting options to your map.

12 Click the **Text** tab.

13 In the **Map title** field, enter a title for your map.

14 In the Legend area, you can click one of the following options:

- **Full legend** to show a detailed legend on your map
- **Compact legend** to show a condensed legend on your map
- **No legend** to exclude the legend from your map.

15 If you click **Full legend**, click **Made by map** to have Crystal Reports create a legend title based on your map, or click **Specify** to enter your own legend title and subtitle.

16 Click **OK**.

Your map is placed in the Header or Footer section of the report, depending on your selection in Step 4.

Mapping on an OLAP cube (OLAP layout)

Using the OLAP layout, you can map on an OLAP grid. In order to create an OLAP map, you must first have an OLAP grid in your report. For more information about OLAP grids, see "Creating an OLAP report" on page 318.

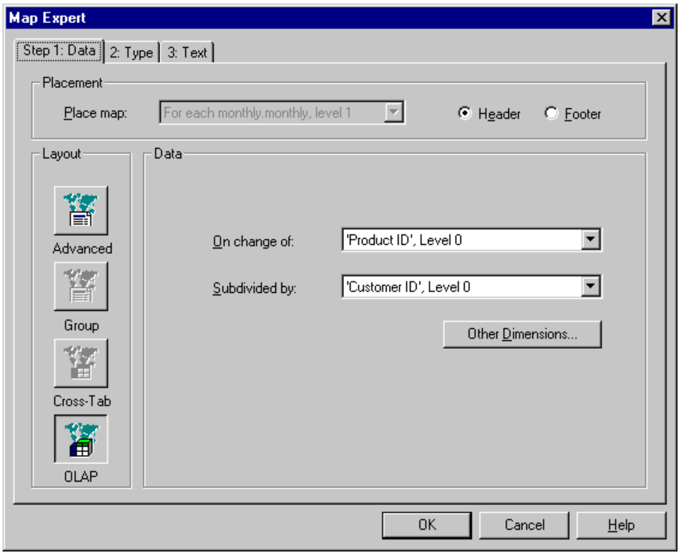
To map on an OLAP cube

1 Select the OLAP grid on which you want to map.

2 On the **Insert** menu, click **Map**.

Tip: Another way to do this is to click the Insert Map button on the Standard toolbar.

The Map Expert dialog box appears.



Map Expert

Placement

Place map: For each monthly.monthly, level 1

● Header ○ Footer

Data

On change of: 'Product ID', Level 0

Subdivided by: 'Customer ID', Level 0

3 On the **Data** tab, in the Layout area, click **OLAP**, if it is not already selected.

4 In the Placement area, click **Header** or **Footer** to specify where to place your map.

5 In the Data area, click the **On change of** list and select the field on which you want to base your map.

Note: This field should contain geographic information, such as Country.

6 If necessary, click the **Subdivided by** list and select a secondary row or column to incorporate into your map.

Note: Choosing to subdivide by a secondary mapping field, either row or column, will affect the map types available in Step 8.

7 Click the **Type** tab.

8 Click the map type that best illustrates your data. If there is no secondary mapping field, you can choose **Ranged**, **Dot Density**, or **Graduated**; if there *is* a secondary mapping field, you can choose **Pie Chart** or **Bar Chart**. See "Map types" on page 267.

9 In the Options area, apply formatting options to your map.

10 Click the **Text** tab.

11 In the **Map title** field, enter a title for your map.

12 In the Legend area, you can click one of the following options:

- **Full legend** to show a detailed legend on your map
- **Compact legend** to show a condensed legend on your map
- **No legend** to exclude the legend from your map.

Note: If you click Full legend, click "Made by map" to have Crystal Reports create a legend title based on your map, or click Specify to enter your own legend title and subtitle.

13 Click **OK**.

Your map is placed in the Header or Footer section of the report, depending on your selection in Step 4.

Mapping on subreports

Inserting a map into a subreport involves the same process used to insert a map into the main report. The only difference is that you must insert the map with the subreport Design tab active.

The process for creating a map varies depending on the type of data you are mapping on.

To map on a subreport

1. On the **Insert** menu, click **Map**.

   Tip: Another way to do this is to click the Insert Map button on the Standard toolbar.

   The Map Expert dialog box appears.
2. Make any necessary changes to the Map Expert, then click **OK**.

   Crystal Reports places the map in the section you specified in the Map Expert, and returns you to the report.
3. Resize or move your map as desired.

Working with maps

Editing maps using the Map Expert

Once you have created a map, you may want to add a new title, headings, and a legend, and you may want to change fonts, or even the map type. From either the Design or Preview tab, you can open the Map Expert and make your modifications.

To edit a map using the Map Expert

1. Right-click the map to bring up the shortcut menu.
2. On the shortcut menu, click **Map Expert**.
3. In the Map Expert dialog box, make the desired changes.
4. Click **OK**.

Editing maps on the Analyzer tab

Crystal Reports provides the Analyzer tab for advanced mapping. Use this tab to organize your maps for more efficient analysis. The Analyzer tab allows you to set the properties of an individual instance of a map independently from the rest of the report.

To launch the Analyzer tab

1. On the **Preview** tab, right-click the map to bring up the shortcut menu.
2. On the shortcut menu, click **Map Analyzer**.

   Crystal Reports opens the Analyzer tab.

When viewing a map on the Analyzer tab, you can perform the following operations:

- zoom in, zoom out, and pan the map

- center the map
- change the map title, map type, and set the overall style of the map
- reorganize the layers of report elements (for example, you can display the major cities layer above the highways layer)
- resolve data mismatches
- change the geographic map
- hide and show the Map Navigator.

Note: The Analyzer tab's zooming, panning, and centering capabilities are also available as shortcut commands on the Map Navigator and the Preview tab (right-click the map object on the Preview tab). For more information about the Map Navigator, see "Zooming, panning, and centering maps with the Map Navigator" on page 286.

With the Analyzer tab, up to two map objects can be viewed simultaneously. In this case, the Analyzer tab becomes a split-screen tab, with each map appearing in a separate section. To resize a section, move the cursor over the section boundary until the cursor becomes a Section Sizing cursor, and then drag the boundary line to expand or reduce the size of the section, as needed.

The Analyzer tab can also be used to work with map objects that have been placed within a subreport. The program opens a separate Analyzer tab for each subreport that contains a map object you are analyzing. You can view up to two map objects from the same subreport simultaneously on the Analyzer tab.

Note: You can also resize the map legend displayed to the right of each map.

Changing the map title

1. On the **Analyzer** tab, right-click the map to bring up the shortcut menu.
2. On the shortcut menu, point to **Format Map** and choose **Title**.
3. In the Change Map Title dialog box, enter a new title.
4. Click **OK** to save your changes.
   Crystal Reports returns you to the Analyzer tab and implements your changes.

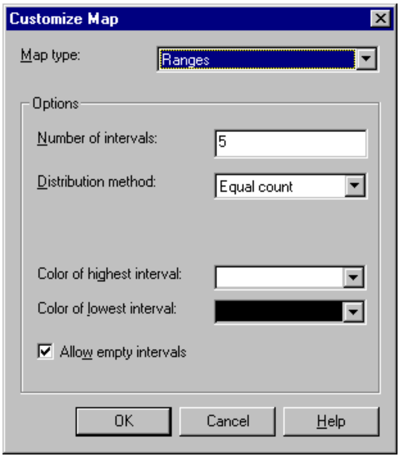
Changing the map type

From within the Analyzer tab, you can change the map type and set the properties for that map. For example, if you are analyzing a Ranged map, and you want to see how the data would look if it were presented in a Dot Density style, you can rearrange the map without having to return to the Map Expert and rework the format. The Analyzer tab also offers more advanced options for changing the appearance of the map you have specified.

To change the map type

1 On the **Analyzer** tab, right-click the map to bring up the shortcut menu.

2 On the shortcut menu, point to **Format Map** and choose **Type**.
The Customize Map dialog box appears.



Map type: Ranges

Number of intervals: 5

Distribution method: Equal count

Color of highest interval:

Color of lowest interval:

☑ Allow empty intervals

3 In the **Map type** list, click the new map type.

4 In the Options area, apply formatting options to your new map.
The properties made available to you vary depending on the map you have selected.

5 Click **OK** to save your changes.
Crystal Reports returns you to the Analyzer tab and implements your changes.

Changing map layers

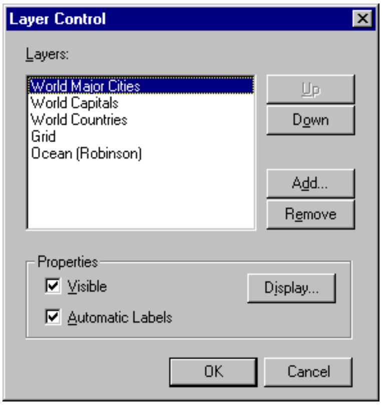
It takes only a moment to rearrange map layers. You can also set properties, such as visibility for those layers, and you can add and remove layers as needed.

To change map layers

1 On the **Analyzer** tab, right-click the map to bring up the shortcut menu.

2 On the shortcut menu, point to **Format Map** and choose **Layers**.

The Layer Control dialog box appears.



World Major Cities
World Capitals
World Countries
Grid
Ocean (Robinson)

☑ Visible

☑ Automatic Labels

3 In the **Layers** list, click **Up** or **Down** to arrange the map layers.

4 In the Properties area, set the properties for each map layer, specifying whether the layer is visible and whether it is automatically labeled.

5 If necessary, click **Display** to open the Display Properties dialog box.

6 Use this dialog box to set the default display mode and zoom range (the minimum and maximum possible magnification) for the layer in question, then click **OK** to return to the Layer Control dialog box.

7 Click **OK** to save your changes.

Crystal Reports returns you to the Analyzer tab and implements your changes.

Resolving data mismatches

Sometimes the map you are working with uses a different geographic name from the one used in the database. For example, a map of Europe may use “United Kingdom” while the database uses the abbreviation “U.K.” Until you resolve the data mismatch, the map will not be able to display any information for the selected geographic region.

To resolve data mismatches

1 On the **Analyzer** menu, click **Resolve Mismatch**.

The Map Region Resolver dialog box appears, listing the geographic names that do not correspond between the database and map.

2 On the **Keyword Matching** tab, in the **Assign this heading** list that is used by the database, click the heading you want to resolve.

The selected heading may be automatically assigned to a keyword from the **To this keyword** list that is used by the map.

3 To assign a keyword yourself, click another keyword from the **To this keyword** list.

4 If there is a match, click **Assign**.

The new assignment is displayed in the bottom part of the Map Region Resolver dialog box.

5 Once you are finished resolving the data mismatches, click **OK**.

The map can now display data for the selected geographic region.

Changing the geographic map

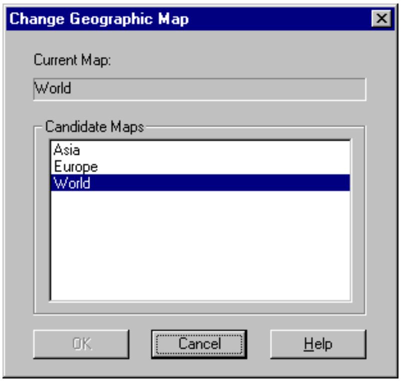
If you prefer to have your data values presented with a different geographic map, specify your changes using the Change Map dialog box.

To change the geographic map

1 On the **Analyzer** tab, right-click the map to bring up the shortcut menu.

2 On the shortcut menu, click **Change Geographic Map**.

The Change Geographic Map dialog box appears, displaying the name of the current map you are using, and a list of replacement maps that you can choose from.



Change Geographic Map

Current Map: World

Candidate Maps

Asia
Europe
World

3 In the **Candidate Maps** list, click the new map you want to use.

4 Click **OK** to save your changes.

Crystal Reports returns you to the Analyzer tab and implements your changes.

Changing map size and position

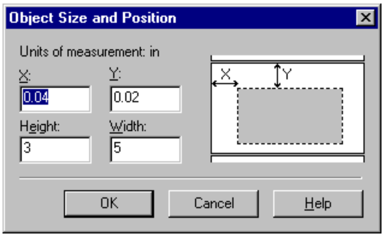
You can use the Object Size and Position dialog box to specify the height, width, and position of your map, in inches.

To change map size and position

1 On the **Design** or **Preview** tab, right-click the map to bring up the shortcut menu.

2 On the shortcut menu, click **Object Size and Position**.

The Object Size and Position dialog box appears.



Object Size and Position

Units of measurement: in

X: 0.04

Y: 0.02

Height: 3

Width: 5

3 In the **X** text box, enter the distance in inches from the left edge of the map object to the left margin of the report section. The default distance is 0.04 inches.

4 In the **Y** text box, enter the distance in inches from the upper edge of the map object to the upper margin of the report section.

The default distance is 0.02 inches.

5 In the **Height** text box, enter the vertical height of the map, in inches.

6 In the **Width** text box, enter the horizontal height of the map, in inches.

7 Click **OK** to save your changes.

Crystal Reports returns you to the report and implements your changes.

Note: You can also change size and position by moving and resizing the map. However, the Object Size and Position dialog box allows you to make exact adjustments.

Changing the border of a map

1 On the **Design** or **Preview** tab, right-click the map to bring up the shortcut menu.

2 On the shortcut menu, click **Borders and Colors**.

The Format Editor dialog box appears with the Border tab open.

3 Change the line style, color, background color, and add or remove a drop shadow from the map border.

4 Click **OK** to save your changes.

Crystal Reports returns you to the report and implements your changes.

Using the underlay feature with maps

Since maps can only print in certain areas and sections of your report, the underlay feature makes mapping even more useful. Instead of having a map print before the data it represents, you can have the map appear alongside the data, thus making the report easier to read and easier to understand.

To underlay a map

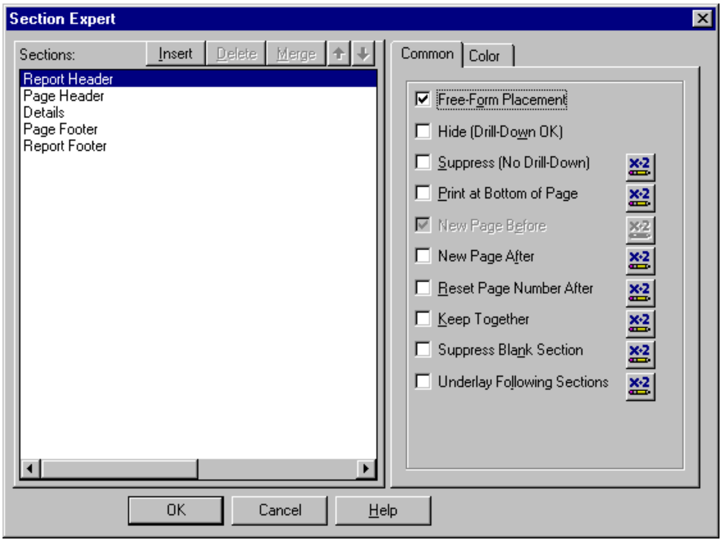
1 Create your map and place it in the Report Header section.

For more information, see "Creating maps" on page 270.

2 On the **Insert** menu, click **Section**.

Tip: Another way to do this is to click the Section Expert button on the Standard toolbar.

The Section Expert dialog box appears.



Report Header
Page Header
Details
Page Footer
Report Footer

☑ Free-Form Placement
☐ Hide (Drill-Down OK)
☐ Suppress (No Drill-Down)
☐ Print at Bottom of Page
☑ New Page Before
☐ New Page After
☐ Reset Page Number After
☐ Keep Together
☐ Suppress Blank Section
☐ Underlay Following Sections

3 In the Sections area, click **Report Header**, then select the **Underlay Following Sections** check box.

4 Click **OK**.

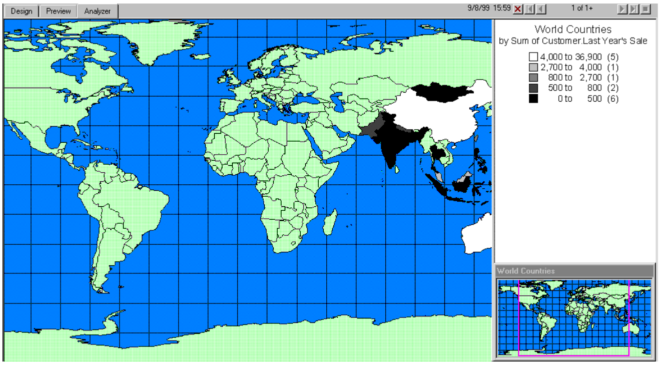
Crystal Reports returns you to the report. Your map will now underlay the sections below it.

5 If necessary, move or resize the map.

Zooming, panning, and centering maps with the Map Navigator

On the Analyzer tab, the Map Navigator provides a small scale version of the currently displayed map. Similar to the Preview and Analyzer tabs, the Map Navigator includes commands for zooming, panning, and centering geographic maps. You can zoom in and out on your maps whenever you need to view more specific or general information.

By panning, you can quickly drag your map anywhere. You can then return your map to the center of the tab by using the center command.



One of the benefits to using the Map Navigator is that it always offers a complete view of the geographic map you are working on. If you use the Map Navigator to perform a command on a particular part of the map, the changes are reflected on the full-sized map on the Analyzer tab. For example, you could choose the Zoom In command on the Map Navigator to examine the eastern United States in more detail. The full-sized map displayed on the Analyzer tab would then zoom in one level of magnification on the eastern United States, leaving the map on the Map Navigator intact.

Note: While the Map Navigator allows you to easily select a portion of the map, it does not display your changes in detail. The mapping tasks that require more attention to detail should be performed with the full-sized map displayed on the Analyzer tab.

To zoom in and out on a map

1 On the **Map Navigator**, right-click the map to bring up the shortcut menu.

2 On the shortcut menu, click **Zoom In**.

3 Drag the selection cursor around the map section you want to see, then decrease the size of the cursor to zoom in on the section.

The full-sized map on the Analyzer tab zooms in one level of magnification.

4 Decrease the size of the selection cursor again to zoom in further.

5 To zoom out on a map, right-click the **Map Navigator** to bring up the shortcut menu.

6 On the shortcut menu, click **Zoom Out**.

7 Increase the size of the selection cursor to zoom out on the map section.

The full-sized map on the Analyzer tab zooms out a level of magnification.

8 Increase the size of the selection cursor again to zoom out further.

To pan a map

1 On the **Map Navigator**, right-click the map to bring up the shortcut menu.

2 On the shortcut menu, click **Pan**.

3 Click the panning cursor, then drag it to the desired map section.

To center a map

1 On the **Map Navigator**, right-click the map to bring up the shortcut menu.

2 On the shortcut menu, click **Center Map**.

Note:

- You can also click None from the shortcut menu to disable all movement options. This will prevent accidental panning or zooming.
- If a map occurs once for each instance of a group, any panning or zooming settings you specify are instance-specific. In other words, if you zoom in on the map in one group header, this setting is only for that group header. None of the headers have been changed. This way you can set each map to different settings.
- If the Save Data with Report option is selected from the File menu, your panning and zooming settings will be saved with the report. If the option is not selected, then only the default map settings will be saved.

Hiding and showing the Map Navigator

You have the option of hiding or showing the Map Navigator.

To hide the Map Navigator

1. On the **Analyzer** tab, right-click the map to bring up the shortcut menu.
2. On the shortcut menu, point to **Map Navigator** and click **Hide**.
   The Map Navigator is removed from the Analyzer tab.

To show the Map Navigator

1. On the **Analyzer** tab, right-click the map to bring up the shortcut menu.
2. On the shortcut menu, point to **Map Navigator** and click **Show**.
   The Map Navigator is displayed on the Analyzer tab.

OLE

15

This chapter explains how Object Linking and Embedding (OLE) can be used to edit graphics or other objects from within your report instead of opening an additional application.

OLE overview

Object Linking and Embedding (OLE) enables you to insert objects (OLE objects) into a report from other applications (OLE server applications) and then use those applications from within Crystal Reports to edit the objects if necessary.

If you were not using OLE, you would have to exit Crystal Reports, open the original application, change the object, return to Crystal Reports, delete the object originally inserted, and then insert the newly revised object.

OLE terminology

Before you work with OLE objects in Crystal Reports, it might be helpful to review some terminology:

- **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 that can be edited using the original application.
- **OLE Object**
  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. A bitmap created in Paint, an Excel spreadsheet, or a graph from MS Graph may all be OLE objects if they are inserted in the receiving document as OLE objects. If they are not inserted as OLE objects, they retain no relationship with the original application.
- **OLE Container Application**
  An OLE container application is one that can contain and process OLE objects created in other applications (such as Paint or Excel). Crystal Reports is a container application.
- **Container Document**
  A container document is a document that was created using the container application and that contains an OLE object.
- **OLE Server Application**
  An OLE server application is an application that is able to create OLE objects which can then be placed into container documents. Microsoft Word and Excel are examples of applications that are both OLE servers and OLE containers. That is, they can both create new OLE objects and they can contain OLE objects created elsewhere.
- **Server Document**
  The server document is a file created in the server application that stores the original OLE object.

Types of OLE objects

- **Static object**
  A static OLE object is a picture of an object that can be displayed and printed, but not edited in place. It does not have a connection to the server document.
- **Embedded object**
  An embedded object contains a representation of the object, as well as information that defines the content. It does not have a connection to the original object in the server document. As a general rule, use embedded objects when you need to edit the object independently of the original server document.
- **Linked object**
  A linked object contains a representation of the data and a link to the file where it originated. As a general rule, use linked objects when the data in the server document is changing and you want the object in your report to be updated when you open the report.

General OLE considerations

There are several points to keep in mind when utilizing OLE functionality.

- When you double-click an embedded OLE object, the object is activated for editing and the Report Designer merges its menus and toolbars with those of the object's server application. When you are finished editing, click outside the object and the Report Designer toolbars reappear.
- When you double-click a linked OLE object, the program opens the object's server application with the object displayed and ready for editing. You cannot 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, displaying the original in the server application limits access to one editor at a time.
- Windows 3.x shipped Microsoft Paintbrush™ (PBRUSH.EXE) as the native bitmap editor. Windows 95 and Windows 98 ship Microsoft Paint™ (MSPAINT.EXE) as the new bitmap editor. While Windows 95 and Windows 98 know to register and convert PBRUSH.EXE OLE bitmaps as MSPAINT OLE bitmaps, the opposite is not true.
- You can open a report containing a bitmap object created in Windows 3.1, in Windows 95 and Windows 98; however, you cannot open a Windows 95 and Windows 98 report that contains a bitmap object in Windows 3.1. Thus, if you are going to be working back and forth between these two operating systems, it is best to create the reports in Windows 3.1 rather than in Windows 95 and Windows 98.

Inserting OLE objects into reports

There are several ways to insert OLE objects into an application.

- The OLE object choice on the Insert menu can be used to import an existing object or create a new one. You can place either embedded or linked objects using this method.
- Paste Special on the Edit menu can be used to cut the object from an OLE server application and paste it in a report. If the object can be pasted in multiple formats, you need to decide which format to use. For example, when inserting text from a Microsoft Word document, you can paste the text as Microsoft Word document text (which can be edited in Word), or as a metafile, which is simply a non-editable picture of the text. Use Paste Special to place either embedded or linked objects.
- 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 OLE server application in another and then dragging the object between the two. However, when an object is dragged into a report, the object is embedded, not linked.

Note: When inserting and selected information (portions of larger files), it is best to use either Copy and Paste Special, or drag and drop editing. When inserting entire files, click OLE Object from the Insert menu.

To drag and drop OLE objects

This procedure assumes that you have Microsoft Excel or another spreadsheet program on your computer that is an OLE server application.

1. Open both Crystal Reports and Microsoft Excel and size the program windows so that they are both visible. You will need to be able to see both the object's origin and its destination.
2. Open an existing report in Crystal Reports.
3. Create a spreadsheet in Microsoft Excel by typing the numbers one through ten (1-10) in cells A1 through A10.
4. Select cells A1 through A10.
5. Move the cursor to the left edge of the highlighted cells until an arrow pointer appears.
6. Press the left mouse button and hold it down. This activates the cells to be moved. Since you're only going to copy the cells, press the Ctrl key and hold it down too.
7. Drag the cells into the Details section of the report in Crystal Reports, and when you have the cells positioned, release the mouse button then the Ctrl key.
8. Right-click the spreadsheet object in the report and observe that it is identified as an OLE object and that the second command from the last identifies it as a Worksheet Object.

9 Double-click the spreadsheet object.

The menus and tools change to those from the OLE server application. You can now edit the object in-place. The object is an embedded object. Any changes you make to the object will not affect the original.

10 Click outside the object when you are finished.

The menus and tools change back to those of Crystal Reports.

How OLE objects are represented in a report

An OLE object is displayed in a report in any of several different formats depending on how you chose to insert the object.

- If you have chosen to display the object as an icon, the icon will appear in the report. Users can choose to edit the bitmap later by double-clicking the icon.
- If the object was created from an existing file, the data from that file (or an icon) is displayed in the report. This data can be edited by double-clicking the object or its icon.
- If you are creating a new object, the application for the object type that you chose will open, and you can begin designing the object. When you are finished, close or exit the application. The object (or its icon) will be displayed in the report.

Editing OLE objects in reports

In-place editing

In-place editing is the ability to change an OLE object's properties from within an OLE container application (such as Crystal Reports). The container application's menu items change to provide the editing tools from the server application so that you can make changes easily.

When an OLE object is placed in a report, the object becomes part of the report. To edit the object, double-click it and then modify it using the editing tools found in the object's original application or from a similar application that allows in-place editing.

Dynamic OLE menu commands

The Object on the Edit menu is dynamic; it changes to reflect the properties of the selected object.

The object may be described as a Document object, Bitmap Image object, Picture object, Worksheet object, or something similarly descriptive.

- if the object is embedded, the Edit menu displays those commands that are available to that type of embedded object

- if the object is linked, the Edit menu displays commands for that type of linked object.

Commands on the context menus change in a similar fashion.

These dynamic commands are provided to give you more control when working with OLE objects.

OLE and the Insert Picture command

When the Object command on the Insert menu is used to place pictures in a report, the picture can be treated as either an embedded or a linked object.

When the Picture command on the Insert menu is used to place pictures in a report, the Report Designer converts them into static objects.

Static objects do not support editing. To edit a static object, it has to be converted into an editable type of object using the Convert command on the Edit menu.

Working with static OLE objects

To insert a static OLE object

1 Create a report using the sample data from **Xtreme.mdb**. Place any field in the **Details** section of the report.

2 Click the **Design** tab.

3 On the **Insert** menu, click **Picture**.

Tip: Another way to do this is to click the picture button on the Supplementary toolbar.

The Open dialog box appears.

Note: Crystal Reports supports Bitmap, TIFF, JPEG, and PNG files.

4 Select **Xtreme.bmp** from the `\Program Files\Seagate Software\Crystal Reports\Samples\En\Databases` directory.

Note: The location of the bitmap may vary depending on the destination specified during installation.

5 Click **Open**.

6 When the object frame appears, place the picture on the far left side of the **Report Header** section.

7 Right-click the picture, and notice that the picture is identified as an OLE object on the shortcut menu.

8 When you double-click the object, nothing happens. A static OLE object cannot be edited.

To convert a static OLE object to a bitmap image object

1 Right-click the static OLE object you placed, and choose **Convert Picture Object** from the shortcut menu.

  The Convert dialog box appears.

2 Select **Paintbrush Picture** in the **Object Type** list, and click **OK**.

  You have just converted a static OLE object to a modifiable OLE object.

3 Right-click the object you have just converted.

  The second command from the bottom of the shortcut menu is now Bitmap Image Object, the same command that appeared for the first object you linked.

4 Double-click the object.

  Microsoft Paint, or another graphics application installed on your machine opens, just as it did for the linked bitmap image object.

5 Save the report as OLE.RPT.

Working with embedded vs. linked objects

Since embedded and linked objects each have different properties, it is important for you to consider the capabilities of each when deciding which OLE format to use.

Embedded objects

An embedded object can be created from within Crystal Reports or by using a file that already exists. If you use an existing file, the object is copied to the report.

When changes are made to an embedded object it does not affect the original file and vice versa. The changes are made only to the object within your report.

To insert an embedded OLE object

1 Click the **Design** tab.

2 On the **Insert** menu, click **OLE Object**.

  The Insert Object dialog box appears.

3 Select **Create From File**.

  The dialog box changes, allowing you to either type in an object name or browse.

4 Click **Browse** and choose **Xtreme.bmp** from the `\Program Files\Seagate Software\Crystal Reports\Samples\En\Databases` directory.

  Note: The location of the bitmap may vary depending on the destination specified during installation.

5 Click **Open** to return to the Insert Object dialog box.

6 Click **OK** to return to the report. An object frame appears.

7 Place the object in the Report Header section, just to the right of the first picture.

8 Right-click this picture and observe that it is also identified as an OLE Object on the shortcut menu. The second command from the bottom of the shortcut menu identifies it as a Bitmap Image Object. It is an embedded OLE object.

9 Double-click the object. Note that you have remained in Crystal Reports but the menus and tools have merged with Microsoft Paint, or those of another graphics package installed on your machine. These tools can be used to edit the bitmap in place. Remember that even though you are working with the tools of the OLE server application, you are not working on the original file; you are working with a copy and nothing you do here will affect the original.

Linked objects

When a linked object is inserted into a report, Windows copies a snapshot of the data from a file that already exists. The image of the object is added to your report along with a reference to the data used to create it. The actual data remains with the original file.

When the object is activated from within a report, the original file is opened inside the application that was used to create it. Any changes you make directly affect the original file also.

If you want the data in an object to remain available to other applications, and always reflect the most current changes to the data, link the object to the report.

When you have a linked object and you break the link using the Links dialog box, all connections to the original data in the server document are broken. A linked object in a container application is merely a representation of that object and a link between the object and the server document. When you break the link you're left with only the representation, with no relationship to the original data or to the editing capabilities of the server application. In this situation, Crystal Reports turns the object into a picture, a stand-alone object that can neither be edited using OLE capabilities nor converted into a modifiable OLE object.

To link a bitmap image object

1 On the **Insert** menu, click **OLE Object**.

The Insert Object dialog box appears.

2 Select **Create From File**.

The dialog box changes, enabling you to enter an object name or browse.

3 Click **Browse** and choose **Xtreme.bmp** from the `\Program Files\Seagate Software\Crystal Reports\Samples\En\Databases` directory.

Note: The location of the bitmap may vary depending on the destination specified during installation.

4 Click **Open** to return to the Insert Object dialog box.

5 Select the **Link** check box.

6 Click **OK** to return to the report.

An object frame appears.

7 Place this object just to the right of the embedded object.

8 Right-click this object and observe that this object is identified as an OLE object. The second command from the bottom of the shortcut menu identifies this as a Linked Bitmap Image Object.

9 Double-click the object. Microsoft Paint or the graphics application you are using opens, displaying the original file. Any changes you make to the original will be reflected in the object that appears in your report.

10 Close Microsoft Paint or the graphics application that you are using and return to your report.

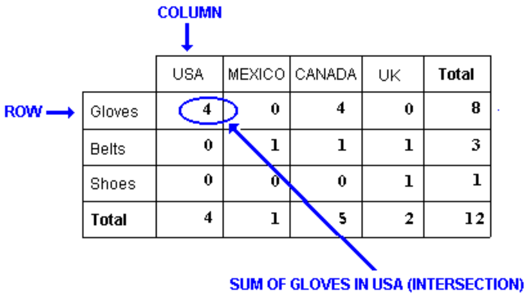
Cross-Tab Objects 16

A Cross-Tab object is a grid that displays values that are grouped and summarized in two directions. This chapter provides you with information about how you can use Cross-Tab objects in your report.

What is a Cross-Tab object?

A Cross-Tab object is a grid that returns values based on the criteria you specify. Data is presented in compact rows and columns. This format makes it easy to compare data and identify trends. It is made up of three elements:

- rows
- columns
- summary fields.

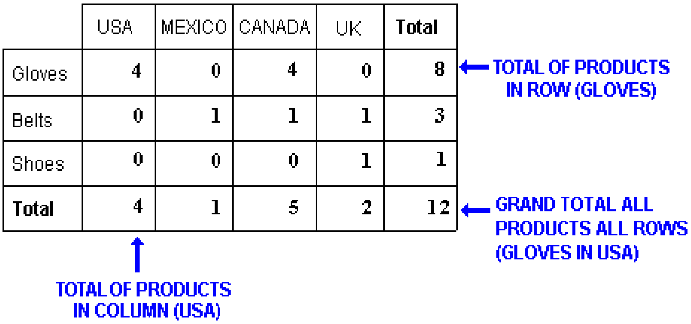


	USA	MEXICO	CANADA	UK	Total
Gloves	4	0	4	0	8
Belts	0	1	1	1	3
Shoes	0	0	0	1	1
Total	4	1	5	2	12

SUM OF GLOVES IN USA (INTERSECTION)

- The rows in a Cross-Tab run horizontally (from side to side). In the example above, “Gloves” is a row.
- The columns in a Cross-Tab run vertically (up and down). In the example above, “USA” is a column.
- The summary fields are found at the intersection of rows and columns. The value found at each intersection represents a summary (sum, count, etc.) of those records that meet the row and the column criteria. In the example above, the value at the intersection of “Gloves” and “USA” is four, the number of gloves sold in the USA.

A Cross-Tab also includes several totals:



	USA	MEXICO	CANADA	UK	Total
Gloves	4	0	4	0	8
Belts	0	1	1	1	3
Shoes	0	0	0	1	1
Total	4	1	5	2	12

TOTAL OF PRODUCTS IN ROW (GLOVES)

GRAND TOTAL ALL PRODUCTS ALL ROWS (GLOVES IN USA)

TOTAL OF PRODUCTS IN COLUMN (USA)

- At the end of each row is a total for that row. In the example above, this total represents a single product sold in all countries. At the end of the "Gloves" row is the value 8, the total number of gloves sold in all countries.
- At the bottom of each column is a total for that column. In the example above, this total represents all products sold in a single country. The value at the bottom of the "USA" column is four, the total number of products (gloves, belts and shoes) sold in the USA.
- At the intersection of the Totals column (totals for the products) and the Totals row (totals for the countries) is a grand total. In the example above, the value at the intersection of the Total Column and Total Row is 12, the total number of all products sold in all countries.

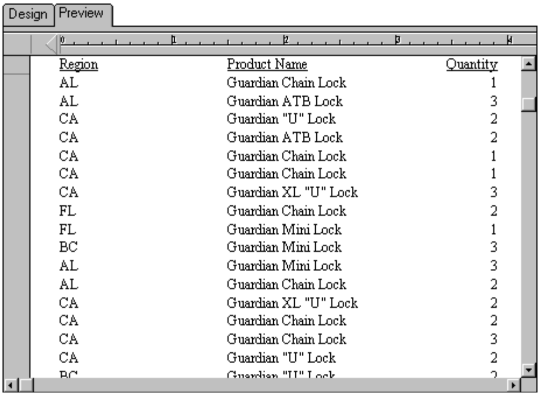
Cross-Tab example

This example demonstrates how you can use a Cross-Tab object to make it easier to understand complex data. The goal of this report is to analyze the unit sales of five different bicycle locks in four different regions (a unit sales of locks *by* region report). For greater clarity, only the most essential information in these reports has been included:

- the region from which the order came
- the name of the lock
- 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



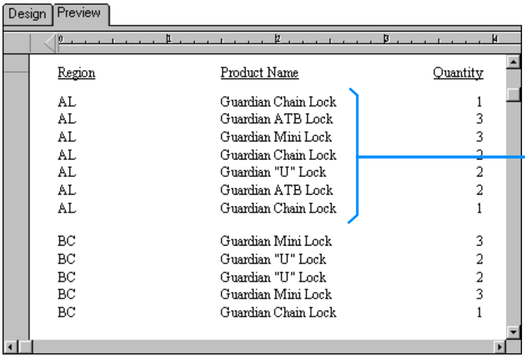
Region	Product Name	Quantity
AL	Guardian Chain Lock	1
AL	Guardian ATB Lock	3
CA	Guardian "U" Lock	2
CA	Guardian ATB Lock	2
CA	Guardian Chain Lock	1
CA	Guardian Chain Lock	1
CA	Guardian XL "U" Lock	3
FL	Guardian Chain Lock	2
FL	Guardian Mini Lock	1
BC	Guardian Mini Lock	3
AL	Guardian Mini Lock	3
AL	Guardian 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 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 next logical step is to group the data in some way. You can group it by region, or by product line. The following section shows a look at both of these options.

Report of order data - grouped by region

This report uses the data seen in the first report, but here the data is grouped by region. All the orders in each region are grouped together, but each regional group contains orders for different types of locks. Because the groups contain different kinds of data, summarizing the Quantity field will determine the total number of locks sold per Region, but not the total of each type.



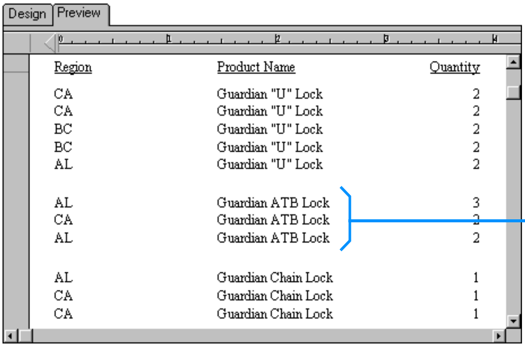
Region	Product Name	Quantity
AL	Guardian Chain Lock	1
AL	Guardian ATB Lock	3
AL	Guardian Mini Lock	3
AL	Guardian Chain Lock	2
AL	Guardian "U" Lock	2
AL	Guardian ATB Lock	2
AL	Guardian Chain Lock	1
BC	Guardian Mini Lock	3
BC	Guardian "U" Lock	2
BC	Guardian "U" Lock	2
BC	Guardian Mini Lock	3
BC	Guardian Chain Lock	1

Each region group contains orders for different kinds of locks.

Report of order data - grouped by product

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 you closer to your goal, but you are still a long way from having the information you need.

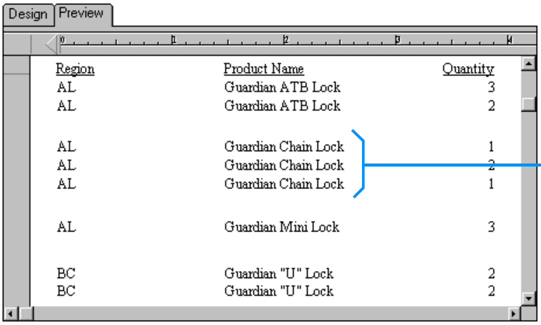


Region	Product Name	Quantity
CA	Guardian "U" Lock	2
CA	Guardian "U" Lock	2
BC	Guardian "U" Lock	2
BC	Guardian "U" Lock	2
AL	Guardian "U" Lock	2
AL	Guardian ATB Lock	3
CA	Guardian ATB Lock	2
AL	Guardian ATB Lock	2
AL	Guardian Chain Lock	1
CA	Guardian Chain Lock	1
CA	Guardian Chain Lock	1

Each product group contains orders for many regions.

Report of order data - grouped by region and product

This report is the logical next step. If the *By Region* report contains multiple products in each region group, and the *By Product* report contains multiple regions in each product group, then it seems to make sense to combine the two. Doing that, you group first by Region and then by Product.



Region	Product Name	Quantity
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

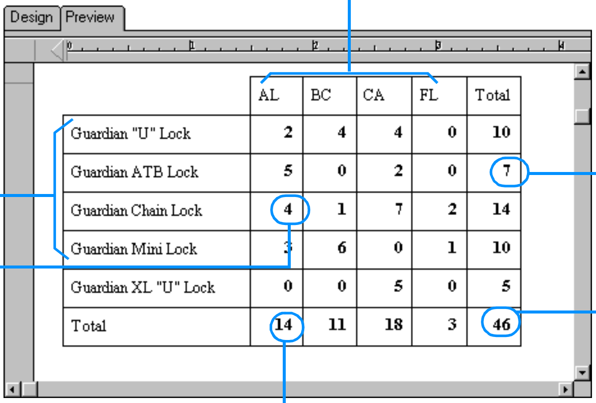
Each group contains orders for one product for one region.

But the data is all spread out and remains difficult to analyze. This information is useful, and with a little work you can use a report like this to get the comparison information you need. However, a Cross-Tab offers a better solution.

Order data in a Cross-Tab object

With Cross-Tabs, all the information you need is provided in a compact format. The report shows the products sold in each region 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.

Regions



	AL	BC	CA	FL	Total
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	14	11	18	3	46

Total - one product in all regions

Product names

Total - one product in one region

Total - all products in all regions

Total - all products in one region

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 for a particular region; for example, the total number of Guardian Mini Locks purchased in British Columbia
- 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 purchased 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 enables you to compare your customers' purchasing habits quickly.

Creating a Cross-Tab report

This section provides you with the steps to create a Cross-Tab object in a new report and how to add a Cross-Tab object to an existing report.

Things to keep in mind when using Cross-Tab objects:

- You can have multiple rows, columns, and summarized fields.
- You can insert as many Cross-Tab objects in a report as you need.
- You can insert the Cross-Tab into either the Report Header or Footer, or the Group Header or Footer.
- You can place Cross-Tab objects in subreports. This is useful when you want to refer to the results from another report.

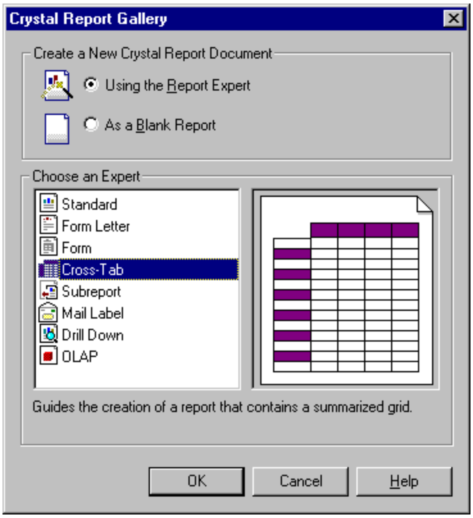
To create a new cross-tab report

1 On the **File** menu, click **New**.

Tip: Another way to do this is to click the New button on the Standard toolbar.

The Crystal Report Gallery appears.

2 In the **Choose an Expert** area click **Cross-Tab**.



Crystal Report Gallery

Create a New Crystal Report Document

● Using the Report Expert

○ As a Blank Report

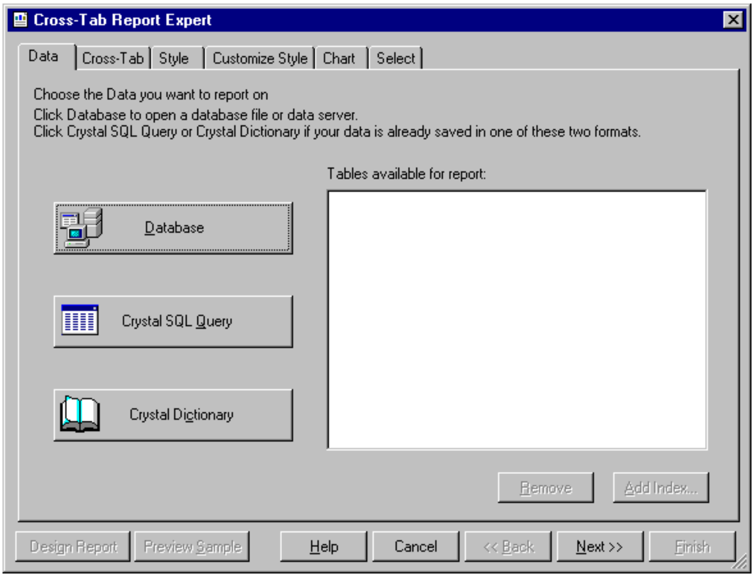
Choose an Expert

- Standard
- Form Letter
- Form
- Cross-Tab
- Subreport
- Mail Label
- Drill Down
- OLAP

Guides the creation of a report that contains a summarized grid.

3 Click **OK**.

The Cross-Tab Report Expert appears.



Cross-Tab Report Expert

Choose the Data you want to report on

Click Database to open a database file or data server.

Click Crystal SQL Query or Crystal Dictionary if your data is already saved in one of these two formats.

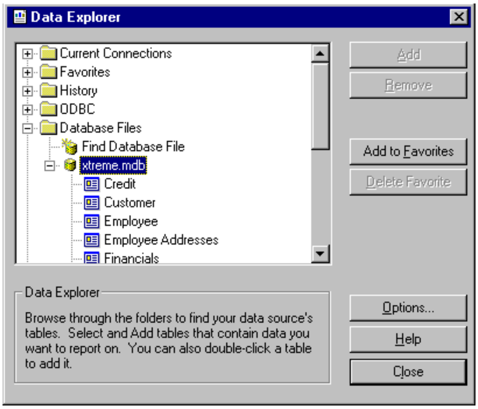
Tables available for report:

Specifying the data source

1 Click the data source type: Database, Crystal SQL Query, or Crystal Dictionary.
This example uses Database.
The Data Explorer dialog box appears.

2 Locate the data source.
This example uses the Xtreme Sample Database.

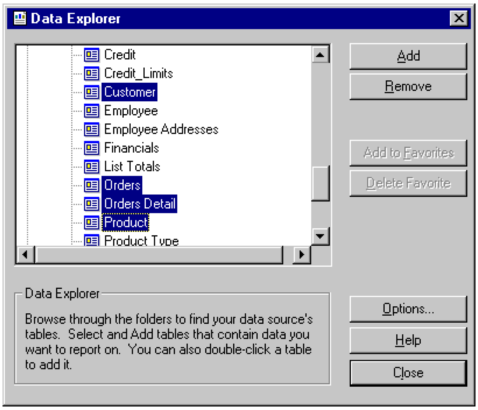
3 Click the plus sign to expand the folder.



Browse through the folders to find your data source's tables. Select and Add tables that contain data you want to report on. You can also double-click a table to add it.

4 Select the tables that contain the fields to include in the report.

Tip: Use the Ctrl-click combination to pick a non-continuous range of fields and the Shift-click combination to pick a continuous list of fields.
This example uses Customer, Orders, Orders Detail and Product.



Browse through the folders to find your data source's tables. Select and Add tables that contain data you want to report on. You can also double-click a table to add it.

5 Click **Add**.

6 Click **Close**.
The Links tab appears.

Modifying the links

1 In this example, check to make sure that the Customer table is linked to the Product table via the Orders and Orders Detail tables.

If you do not like the linking that Crystal Reports has automatically suggested, change it by clicking Clear Links, and then select the fields that you would like to link together.

2 Click **Next**.

The Cross-Tab tab appears.

Defining the structure of the Cross-Tab

Add fields from the Available Fields area to the Rows, Columns, and Summarized Field areas.

1 From the **Customer** table, click **Region**, then click **Add Row**.

Tip: You can also add fields to the different Cross-Tab sections by clicking the field and then dragging and dropping it into either the Columns, Rows, or Summarized Field area.

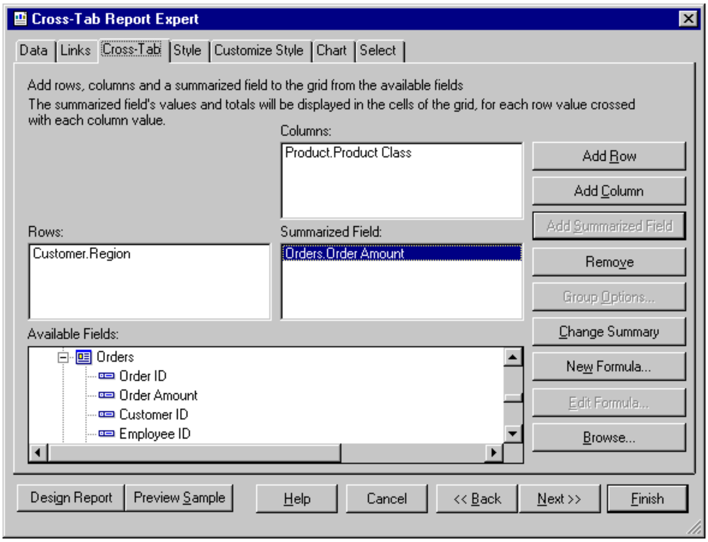
The Region field is added to the Rows area.

2 From the **Product** table, click **Product Class**, then click **Add Column**.

The Product Class field is added to the Columns area.

3 From the **Orders** table, click **Order Amount**, then click **Add Summarized Field**.

The Orders Amount is added to the Summarized Field area.



Cross-Tab Report Expert

Add rows, columns and a summarized field to the grid from the available fields
The summarized field's values and totals will be displayed in the cells of the grid, for each row value crossed with each column value.

Columns: Product.Product Class

Rows: Customer.Region

Summarized Field: Orders.Order Amount

Available Fields:
- Orders
  - Order ID
  - Order Amount
  - Customer ID
  - Employee ID

4 Apply additional structure by:
- Adding formulas to any of the fields—select the field then, click New Formula.

  For more information see "Working with the Formula Editor" on page 387.
- Defining the sort method—select the field then, click Group Options (ascending, descending or specified order).
- Changing the summary operation—select the field then, click Change Summary.

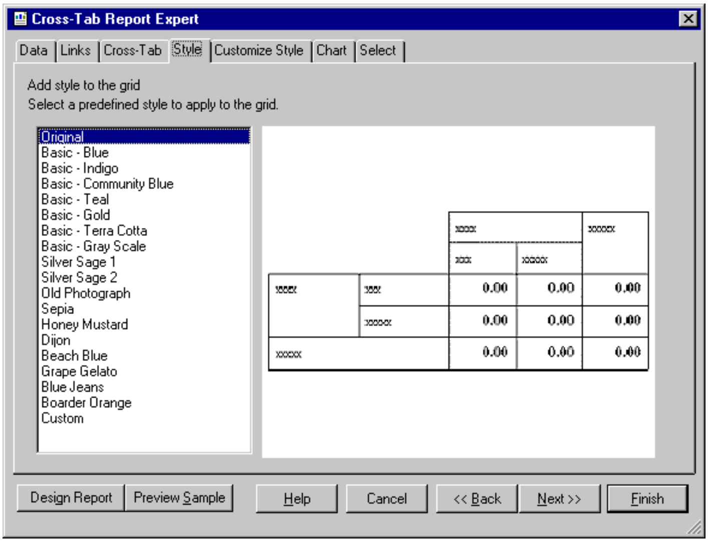
5 Click **Next**.

The Style tab appears.

Applying a predefined style

1 Select a style.

For this example choose Original.



Add style to the grid

Select a predefined style to apply to the grid.

Original
Basic - Blue
Basic - Indigo
Basic - Community Blue
Basic - Teal
Basic - Gold
Basic - Terra Cotta
Basic - Gray Scale
Silver Sage 1
Silver Sage 2
Old Photograph
Sepia
Honey Mustard
Dijon
Beach Blue
Grape Gelato
Blue Jeans
Boarder Orange
Custom

2 Click **Next**.

The Customize Style tab appears.

Formatting the Cross-Tab

Tip: If you do not want to customize the style of your Cross-Tab, click Next.

Change the background color of a row or column

1. Select the row or column.
2. Select the color from the **Background Color** drop-down list.
3. Click **Next** if you have finished customizing your Cross-Tab.

Create an Alias for a row or column

You can create aliases to shorten long row or column names. This is useful when you plan on referring to a row or column in conditional formatting formulas.

1. Select the row or column.
2. Enter the alias name.
3. Click **Next** if you have finished customizing your Cross-Tab.

Changing Grid Options

You can change the formatting of your grid using the check boxes associated with the following options:

- Show cell margins
- Indent row labels
- Repeat row labels
- Keep columns together
- Row totals on top
- Column totals on left
- Suppress empty rows
- Suppress empty columns
- Suppress row grand totals
- Suppress column grand totals.

Format grid lines

In the Format grid lines area you can define whether the lines show or not, as well as the color, style, and width.

1. Click **Format grid lines**.

   The Format Grid Lines dialog box appears.
2. From the list, choose the description of the area where you would like the lines to appear, or click the appropriate area in the Format Grid Line diagram.
3. Select the color, style, and width.

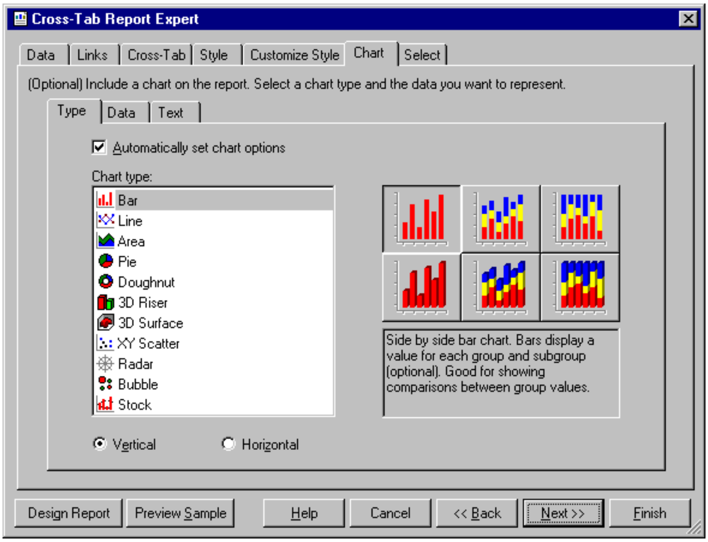
4 Select the **Draw** check box to have your lines appear in the report.

5 Click **OK**.

6 Click **Next**.
The Chart tab appears.

Adding a chart

1 Work through the various tabs to add a chart to your cross-tab.



(Optional) Include a chart on the report. Select a chart type and the data you want to represent.

☑ Automatically set chart options

Chart type: Bar

Side by side bar chart. Bars display a value for each group and subgroup (optional). Good for showing comparisons between group values.

● Vertical ○ Horizontal

For more information see "Charting concepts" on page 248.

2 Click **Next**.

3 The Select tab appears.

Selecting fields

Use this tab to apply selection conditions. This limits the results that are returned in your report to only those records that you are interested in.

1 Click the field to apply selection to.
For this example choose Region.

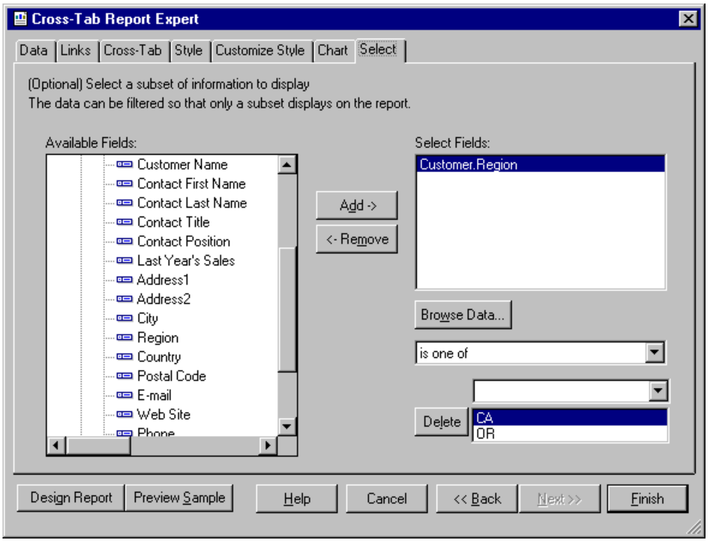
2 Click **Add**.
The field is added to the Select Fields area and the filter drop-down list appears below the Select Fields area.

3 From the filter drop-down list choose a filter method.

Choose "is one of" as the filter method.

Choose AZ, CA, and OR from the drop-down list.

The report will provide you with a breakdown and summary of accessory and bicycle sales for Arizona, California, and Oregon.



(Optional) Select a subset of information to display

The data can be filtered so that only a subset displays on the report.

Select Fields: Customer.Region

is one of

CA
OR

Completing the report

1 Click **Finish**.

The Cross-Tab report appears in the Preview tab.

2 If you do not want the Cross-Tab to appear in the **Report Header**, click the **Design** tab, and move the Cross-Tab to the **Report Footer**, or **Group Header** or **Footer**.

Note: If you place the Cross-Tab in the Report Footer, unsuppress the footer by right-clicking in the Report Footer's grey area and selecting Don't Suppress.

3 Click the **Preview** tab to view the report.

To add a Cross-Tab to an existing report

In this example, a Cross-Tab is added to an existing report to show the sales in North America compared to the rest of the world.

1 Open your report.

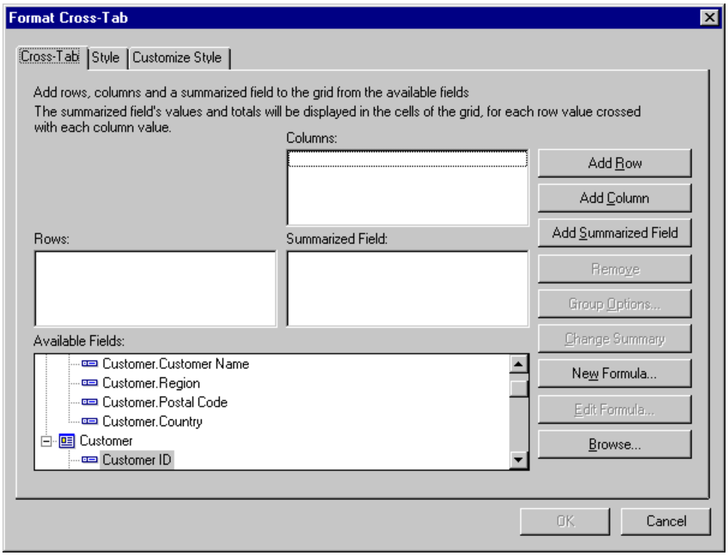
This example uses the Group by Intervals.rpt included in the Feature Examples directory.

2 Click the **Design** tab.

3 On the **Insert** menu, click **Cross-Tab**.

Tip: Another way to do this is to click the Insert Cross-Tab button on the Supplementary toolbar.

The Format Cross-Tab dialog box appears. This dialog box is made up of three tabs: Cross-Tab, Style, and Customize Style.



Add rows, columns and a summarized field to the grid from the available fields
The summarized field's values and totals will be displayed in the cells of the grid, for each row value crossed with each column value.

Defining the structure of the Cross-Tab

Add fields to the Rows, Columns, and Summarized Field areas.

1 From the **Available Fields** list, click **Country**, then click **Add Row**.

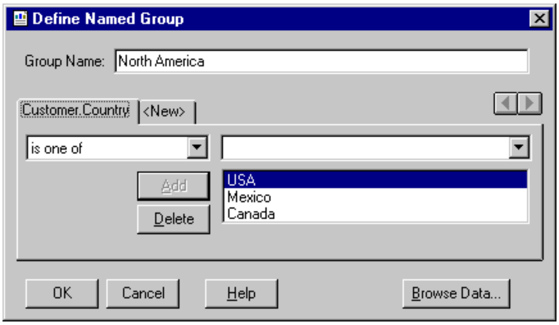
Tip: You can also add the fields to the different Cross-Tab sections by clicking the field and then dragging and dropping it into either the Columns, Rows, or Summarized Field area.

The Country field is added to the Rows area.

2 From the **Available Fields** list, click **Last Year's Sales**, then click **Add Summarized Field**.

The Last Year's Sales field is added to the Summarized Field area.

3. Apply additional structure:
   - click **Country** in the Rows area
   - click **Group Options**
   - select **in specified order** from the drop-down list
   - type **North America** for the Name of the Named Group
   - click **New**
   - choose **is one of** from the drop-down list
   - select **Canada**, **Mexico**, and **USA** from the drop-down list



Define Named Group

Group Name: North America

Customer.Country <New>

is one of

USA
Mexico
Canada

   - click **OK** to close the Define Named Group dialog box
   - click **OK** to close the Cross-Tab Group Options dialog box.

Choosing a predefined style

1. Click the **Style** tab.
2. Select a style.

Applying a custom style

1. Click the **Customize Style** tab.
2. Customize the style as desired.

Finishing the Cross-Tab

1. Click **OK**.
2. Place the Cross-Tab in either the Report Header or Report Footer.
3. On the **Report** menu, click **Refresh Report Data**.
   The updated report appears.

Formatting Cross-Tabs

Crystal Reports has powerful formatting capabilities that can be applied to Cross-Tabs. The following sections describe these key procedures:

- Changing width and height of Cross-Tab cells
- Formatting background color of entire rows/columns
- Formatting fields individually
- Formatting several fields at one time
- Suppressing Cross-Tab data
- Printing Cross-Tabs that span multiple pages.

By applying such formatting as background color, borders, and fonts, you can emphasize important data and create professional-looking and easy-to-understand Cross-Tabs.

For more information, see "Formatting properties" on page 226.

Changing width and height of Cross-Tab cells

1. In the **Design** tab, click a Row or Column heading within the Cross-Tab to activate the sizing handles.
2. Click and drag one of the sizing handles to expand the width or height of the Cross-Tab cell.
3. Click the **Preview** tab to view the changes.

Note: Changing one row or column heading affects all of the rows or columns.

Formatting background color of entire rows/columns

Use background colors to emphasize rows or columns within you Cross-Tab.

1. Right-click the blank top-left area of the Cross-Tab and select **Format Cross-Tab** from the shortcut menu.

   The Cross-Tab dialog box appears.
2. Click the **Customize Style** tab.
3. Click the row (in the **Rows** area) or column (in the **Columns** area), and select a color from the **Background Color** drop-down list.
4. Click **OK** to return to the Cross-Tab.

   The row/column is formatted as specified.

Formatting fields individually

1 Right-click the field you want to format and choose **Format Field** from the shortcut menu.

The Format Editor appears.

2 In the Format Editor, make font, background, borders, numbering, currency symbols, and printing characteristics choices.

3 Click **OK** to return to the Cross-Tab.

The field is formatted as specified.

Formatting several fields at one time

1 Use the Shift-click method to highlight the desired fields.

2 Right-click any one of them and choose **Format Objects** from the shortcut menu.

The Format Editor appears.

3 In the Format Editor, make font, background, borders, numbering, currency symbols, and printing characteristics choices.

4 Click **OK** to return to the Cross-Tab.

The fields are formatted as specified.

Suppressing Cross-Tab data

This section provides you with the steps required to suppress data in your report. You can suppress:

- empty rows and columns
- row and column grand totals
- subtotals and their labels.

To suppress empty rows and columns

1 Right-click the blank top-left area of the Cross-Tab and select **Format Cross-Tab** from the shortcut menu.

The Cross-Tab dialog box appears.

2 Click the **Customize Style** tab.

3 Select either the **Suppress Empty Rows** or **Suppress Empty Columns** check box.

4 Click **OK**.

Now, when you print the report, empty rows and/or columns will not appear.

To suppress row and column grand totals

1. Right-click the blank top-left area of the Cross-Tab and select **Format Cross-Tab** from the shortcut menu.

   The Cross-Tab dialog box appears.
2. Click the **Customize Style** tab.
3. Select either the **Suppress Row Grand Totals** or **Suppress Column Grand Totals** check boxes.
4. Click **OK**.

To suppress subtotals and labels

If you have more than two groups in your Cross-Tab you can suppress the subtotal and label for one of them.

1. Right-click the blank top-left area of the Cross-Tab and select **Format Cross-Tab** from the shortcut menu.

   The Format Cross-Tab dialog box appears.
2. Click the **Customize Style** tab.
3. Click the field whose subtotal you want to suppress.

   The Suppress Subtotal and the Suppress Label check boxes become active.
4. In the Group Options area, select the **Suppress Subtotal** check box.
5. Click the **Suppress Label** check box to suppress the label associated with subtotal.
6. Click **OK**.

Printing Cross-Tabs that span multiple pages

When you create a Cross-Tab that is wider or longer than the specified page size, the program automatically spans the printing across enough pages to accommodate the size of the Cross-Tab. In the Preview tab, a line will appear at each page break. For ease in reading, column headings will be repeated on subsequent pages. Row headings can also be repeated using the Keep Columns Together option.

To repeat row labels

1. Right-click the blank top-left area of the Cross-Tab and select **Format Cross-Tab** from the shortcut menu.

   The Cross-Tab dialog box appears.
2. Click the **Customize Style** tab.
3. Select the **Repeat Row Labels** check box, if desired.

Creating and Updating OLAP Reports 17

This chapter describes how to create OLAP reports with a variety of OLAP and Multi-Dimensional data sources and how to update them when the location of your data changes.

OLAP reporting with Crystal Reports

Although relational databases such as SQL servers and PC databases are the most common sources of data, Online Analytical Processing (OLAP) and Multi-Dimensional Data (MDD) are rapidly becoming the popular data-storage and analysis formats. Crystal Reports provides the same access and reporting features for OLAP data sources that it provides for relational data.

Note: The term OLAP is used in this topic to refer to all common forms of OLAP and Multi-Dimensional Data storage and access systems.

OLAP Grid Objects

When you design a report with OLAP data, Crystal Reports creates a primary report containing one or more OLAP grid objects. OLAP grid objects look and act much like Cross-Tab objects, but they are designed specifically for OLAP data.

OLAP grid objects provide true multi-dimensional reporting. Add dimensions to either axis to analyze three, four, or more dimensions in a single OLAP grid. Or instead of viewing multiple dimensions within one OLAP grid, create multiple OLAP grids within the same report.

Creating an OLAP report

OLAP reports are created using the OLAP Report Expert. The expert is made up of seven tabs. Before you start creating your report, ensure you have the following information:

- OLAP Type (Essbase Cube, IBM DB2 OLAP Server™, OLE DB for OLAP, Informix® MetaCube®, or Holos HDC Cube)
- Server/Database name
- User ID
- Password
- Parameters (applies only to OLE DB for OLAP).

To create an OLAP report

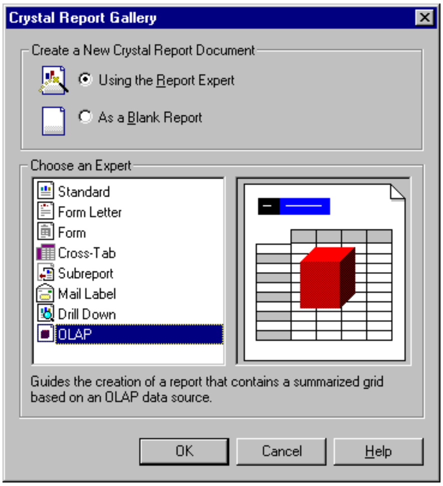
This procedure uses the sample xtreme.hdc cube included in the OLAP Data folder.

1. From the **File** menu, click **New**.

   The Crystal Report Gallery appears.

   Tip: Another way to do this is to click the New button on the Standard toolbar.

2 Select **OLAP**.



Guides the creation of a report that contains a summarized grid based on an OLAP data source.

3 Click **OK**.

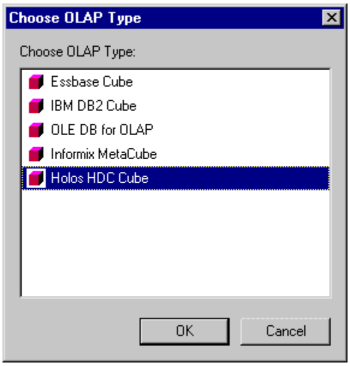
The OLAP Report Expert appears with the Data tab active.

Specifying the data source

Use the Data tab to define the OLAP data source.

1 Click **Server Type**. The Choose OLAP Type dialog box appears.

2 Select the OLAP Type you are using.



3 Click **OK**.

The Browse For Database dialog box appears.

4 Click **Browse**, and select the OLAP source.

This example uses the Xtreme.hdc cube found in the OLAP Data folder.

5 Click **Open**.

The Browse For Database dialog box appears.

6 Click **OK**.

The Data tab appears with the supplied data source information.

7 Click **Next**.

Defining the structure of the grid

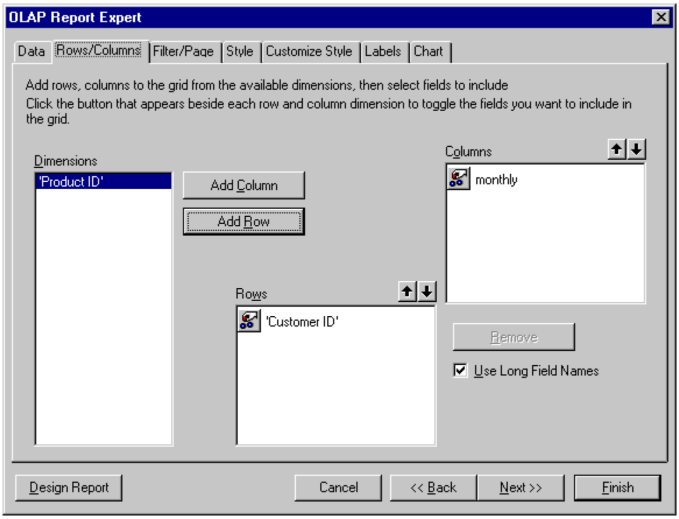
Use the Rows/Columns tab to define how your data is structured within the grid. You can place the dimensions into either the Columns or Rows areas.

Tip: It is also possible to drag and drop the dimensions into either the Rows or Columns area.

1 Select a dimension to appear in the report.

2 Click either **Add Row** or **Add Column**, depending on where you would like the dimension to be located.

3 Continue adding dimensions to the Rows and Columns areas.



OLAP Report Expert

Add rows, columns to the grid from the available dimensions, then select fields to include
Click the button that appears beside each row and column dimension to toggle the fields you want to include in the grid.

Dimensions: 'Product ID'

Columns: monthly

Rows: 'Customer ID'

☑ Use Long Field Names

4 Click the **Up** and **Down** arrow buttons associated with the Rows and Columns area to arrange the order of the dimensions.

Note:

- If you accidentally add a dimension to either the Rows or Columns area, click Remove to return it to the Dimensions list.
- The Use Long Field Names check box is selected by default. This means that, for example, the month field will be displayed as January, February, etc., as opposed to M01, M02, etc.

5 Click the field picker to specify the fields to be included in your report.
The Select Fields for Dimension dialog box appears.

6 Select the fields you would like to include.

7 Click **OK**.

8 Click **Next**.
The Filter/Page tab appears.

Specifying the number of grids and applying filters

Use this tab to limit the records that are returned, and specify the number of grids that are required.

The Filter area determines the boundaries that will be placed on your report. For example, if you have a time dimension that is made up of fiscal quarter fields, you could specify the report return the results for a specific quarter.

Tip: All records are unfiltered by default; if you do not need to filter records or specify the number of grids, click Next.

Note: If all of your fields were used in the Rows/Column tab, nothing will appear here.

The Page area enables you to determine the number of grids and the subject of each. For example, if you have a product dimension, you could put this in the page area and specify two different products. This would result in two grids with the same rows, columns, and formatting, but each one would be based on a different product.

Specify a filter

1 Click the field picker to determine the field that is going to be used as the filter.
The Select Fields for Dimension dialog box appears.

2 Select the appropriate field, expanding the structure if necessary.

3 Click **OK**.

4 Click **Next** if you do not want to add a page.
The Style tab appears.

Add a page

1 Drag and drop the dimension(s) into the page area.
The Select Fields for Dimension dialog box appears.

2 Expand the dimensions structure and select the appropriate fields.

3 Click **OK**.

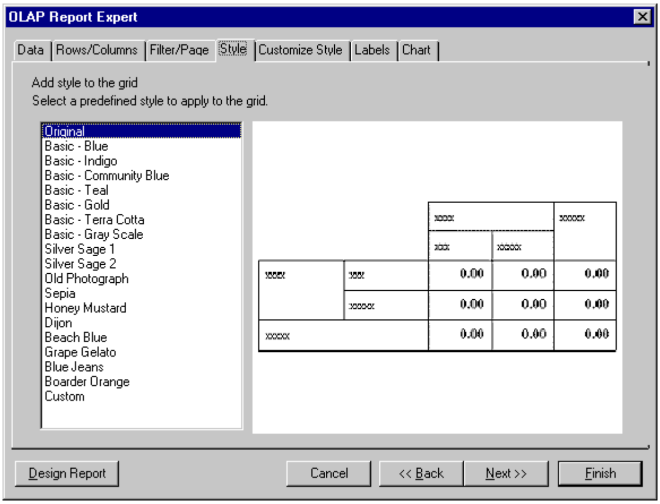
4 Click **Next**.
The Style tab appears.

Applying a predefined style

1 Select a predefined style for the grid from the list.

Tip: If you do not want to use a predefined style, click Next.

A preview of the color scheme you select appears on the right.



Add style to the grid

Select a predefined style to apply to the grid.

Note: The color of the style you select may not appear correctly if your screen resolution is set to 256 colors. Increase the resolution to correct this.

2 Click **Next**.

The Customize Style tab appears.

Formatting the grid

Tip: If you do not want to customize the style of your grid, click Next.

Change the background color of a dimension

1 Select the dimension.

2 Select the color from the Background Color drop-down list.

3 Click **Next** if you have finished customizing your grid.

Create an Alias for a Dimension

You can create aliases to shorten long dimension names. This is useful when you plan on referring to a dimension in a conditional formatting formula (GridRowColumnValue function).

1 Select the dimension.

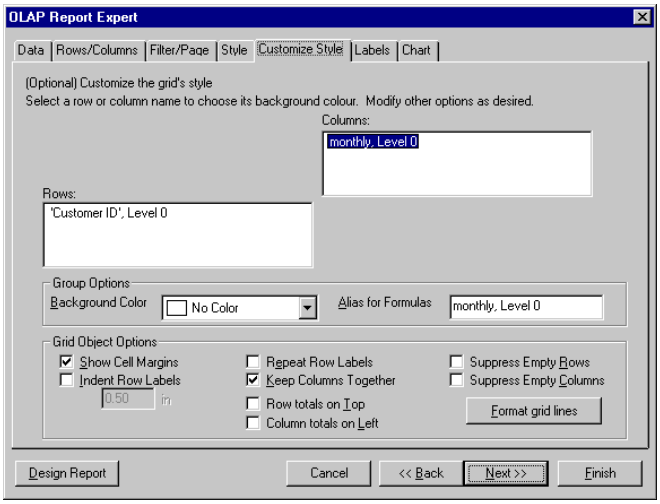
2 Enter the alias name.

3 Click **Next** if you have finished customizing your grid.

Changing Grid Object Options

You can change the formatting of your grid objects using the check boxes associated with the following options:

- Show cell margins
- Indent row labels
- Repeat row labels
- Keep columns together
- Row totals on top
- Column totals on left
- Suppress empty rows
- Suppress empty columns.



(Optional) Customize the grid's style
Select a row or column name to choose its background colour. Modify other options as desired.

Columns: monthly, Level 0

Rows: 'Customer ID', Level 0

Group Options
Background Color: No Color
Alias for Formulas: monthly, Level 0

Grid Object Options
☑ Show Cell Margins
☐ Indent Row Labels 0.50 in
☐ Repeat Row Labels
☑ Keep Columns Together
☐ Row totals on Top
☐ Column totals on Left
☐ Suppress Empty Rows
☐ Suppress Empty Columns

Format grid lines

In the Format Grid Lines dialog box, you can define whether the lines show or not, as well as the color, style, and width.

1 Click **Format grid lines**.
The Format Grid Lines dialog box appears.

2 From the list, choose the description of the area where you would like the lines to appear, or click the appropriate area in the Format Grid Line diagram.

3 Select the color, style, and width.

4 Select the **Draw** check box to have your lines appear in the report.

5 Click **OK**.

6 Click **Next**.

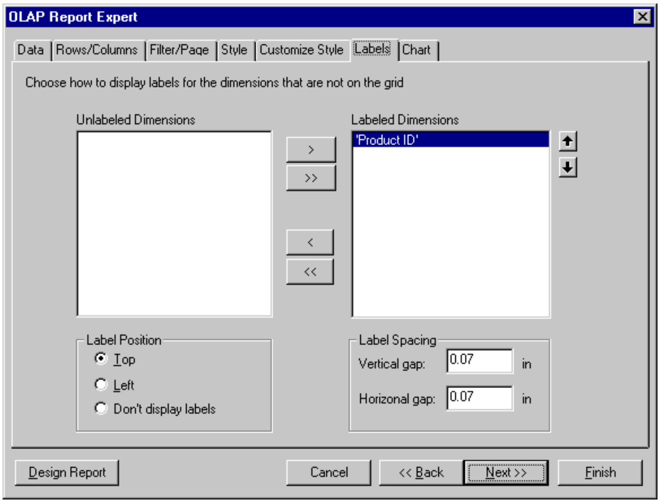
The Labels tab appears.

Labelling dimensions

In the Labels tab, you are able to define which dimensions will be labeled and how they will be displayed in the grid. The dimensions available to you are the ones that have been designated as a page or filter.

1 Using the arrow buttons, move the dimensions to the Unlabeled and Labeled areas as required.

2 Make changes to the appearance of the label by using the options available in the Label Position and Label Spacing areas.



Choose how to display labels for the dimensions that are not on the grid

Unlabeled Dimensions

Labeled Dimensions: 'Product ID'

Label Position

● Top
○ Left
○ Don't display labels

Label Spacing

Vertical gap: 0.07 in

Horizonal gap: 0.07 in

3 Click **Next**.

The Chart tab appears.

Inserting a chart

The charting process for OLAP reports is similar to Standard reports.

For more information refer to "Charting on an OLAP cube (OLAP layout)" on page 257 and "Creating charts" on page 252.

Completing the process

To finalize your report, click Finish.

Updating an OLAP report

The location of the OLAP data accessed by your report may change. Some possible scenarios include:

- the location of the OLAP server or database may change due to restructuring of Information System resources
- another cube instance may have been successfully processed and reflect more current information
- new reports may be developed on a smaller database that represents a subset or production version of the primary OLAP database, then ported over to the actual working database to report on real data.

Before updating your report, it is important to ensure that is compatible with the data. Some things to look for include:

- removing dimensions contained in the report that are not in the cube
- removing fields referred to explicitly in the design of the report, such as a formula.

Note: If a report contains subreports or grids, and the database used for them have changed name or location, you must update each subreport or grid.

To update a cube location in an OLAP grid object

1. Select the grid by clicking on the border.
2. From the **Database** menu, click **Set OLAP Cube Location**.

   Tip: Or use the right mouse button (right-click), and select Set OLAP Cube Location.
   The Confirm Command dialog box appears.
3. Click **Yes**.
   The Set OLAP Database Location dialog box appears.
4. Click **Select**.
   The Choose OLAP Type dialog box appears.
5. Select the OLAP Type.
6. Click **OK**.
   The Connect To Server dialog box appears.
7. Enter the **OLAP Server**, **User ID**, and **Password**.

   Note: The fields will vary depending on the OLAP type you are using.
8. Click **Connect**.
   The Chose Cube dialog box appears.
9. Select a cube.
10. Click **OK**.
    The Set OLAP Location dialog box appears.

11 Click **OK**.

The Propagate Set Location dialog box appears.

12 Click **Yes** to update the remaining OLAP grids and follow the updating the database location steps; otherwise click **No** to return to the report.

Updating the database location

1 Click **Yes** to update the database location.

The Set Location dialog box appears.

2 Select the database, then click **Set Location**.

The Data Explorer appears.

3 Expand the directory for the cube.

4 Select the table, then click **Set**.

The Connect to Server dialog box appears.

5 Enter the **OLAP Server**, **User ID**, and **Password**.

Note: The fields will vary depending on the OLAP type you are using.

6 Click **Connect**.

The Chose Cube Dialog box appears.

7 Select the cube, then click **OK**.

8 The Propagate Set Location dialog box appears with a message asking you to propagate the server and database changes across tables with the same original information.

9 Click **Yes**.

Using the Informix MetaCube SQL Method for reporting

There are two methods for creating OLAP reports based on Informix MetaCube information. The first is using the OLAP Report Expert, and the second is using the Informix MetaCube SQL Method.

This section provides you with instructions for creating OLAP reports for Informix MetaCube databases using the Informix MetaCube SQL Method.

Creating a report with Informix MetaCube is very similar to creating a report based on any relational database. When creating a report, you will see a list of Tables, Sample tables, and Predefined queries with dimensions, attributes, and measures. When working with OLAP data, it is often a good idea to create a Cross-Tab style report. In addition, it is also a good idea to add a filter to your report to limit the number of records that are returned when you run the report. This filtering can be done using the Record Selection Formula Editor.

The following procedures use the Metademo DataSource included with the Informix Metacube CD.

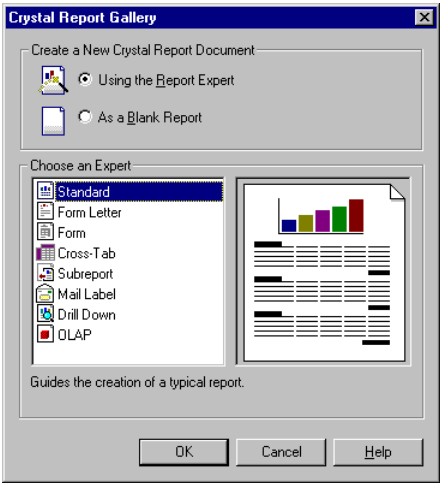
To create an OLAP report using the Informix MetaCube SQL Method

1 On the **File** menu, click **New**.

Tip: Another way to do this is to click the New button on the Standard toolbar.

The Crystal Report Gallery appears.

2 Confirm that **Using the Report Expert** and **Standard** are selected, then click **OK**.



Crystal Report Gallery

Create a New Crystal Report Document

● Using the Report Expert

○ As a Blank Report

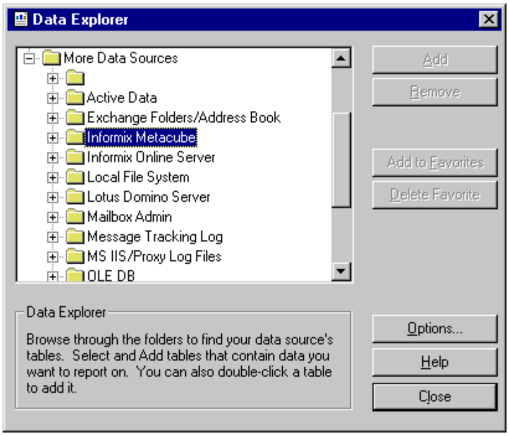
Choose an Expert

Guides the creation of a typical report.

3 Click **Database**.

The Data Explorer appears.

4 Expand the **More Data Sources** folder, then click the **Informix MetaCube** folder.

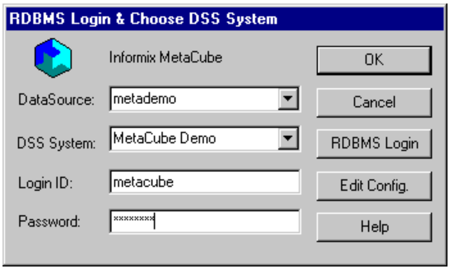


Data Explorer

Data Explorer

Browse through the folders to find your data source's tables. Select and Add tables that contain data you want to report on. You can also double-click a table to add it.

5 In the RDBMS Login & Choose DSS System dialog box, enter the Password, then click **OK**.


RDBMS Login & Choose DSS System

Informix MetaCube

DataSource: metademo

DSS System: MetaCube Demo

Login ID: metacube

Password: ********

6 Select the **Brand Channel Net Profit** table.

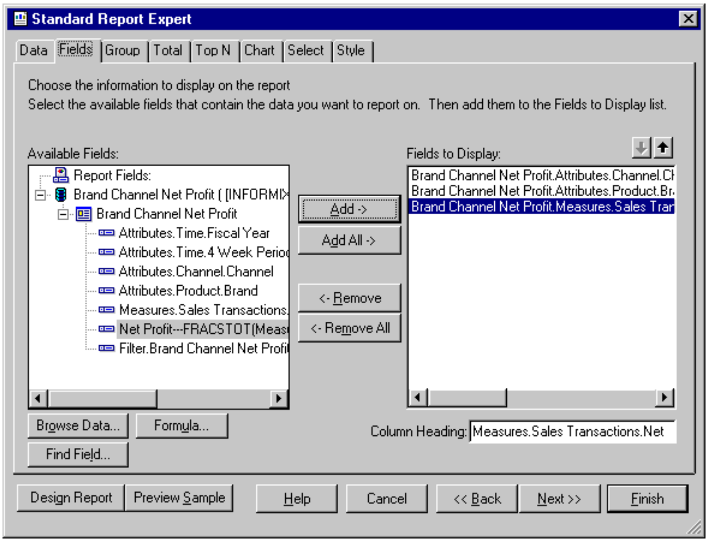
7 Click **Add**, then click **Close**.

The Standard Report Expert appears.

8 Click **Next**.

The Fields tab appears.

9 For this example select the **Channel.Channel**, **Product Brand**, and **Sales Transactions** fields then click **Add**.


Standard Report Expert

Choose the information to display on the report

Select the available fields that contain the data you want to report on. Then add them to the Fields to Display list.

Available Fields:

- Report Fields:
- Brand Channel Net Profit ([INFORMIX
  - Brand Channel Net Profit
    - Attributes.Time.Fiscal Year
    - Attributes.Time.4 Week Perioc
    - Attributes.Channel.Channel
    - Attributes.Product.Brand
    - Measures.Sales Transactions.
    - Net Profit--FRACSTOT(Meas
    - Filter.Brand Channel Net Profi

Fields to Display:

- Brand Channel Net Profit.Attributes.Channel.Ch
- Brand Channel Net Profit.Attributes.Product.Br
- Brand Channel Net Profit.Measures.Sales Trar

Column Heading: Measures.Sales Transactions.Net

10 Click **Preview Sample** to view the report.

Note: You may have to adjust the column widths of the fields and field headers to see all the data. If you wish to hide the Channel Field column simply select the top element in the field, click the right mouse button and select Delete from the shortcut menu.

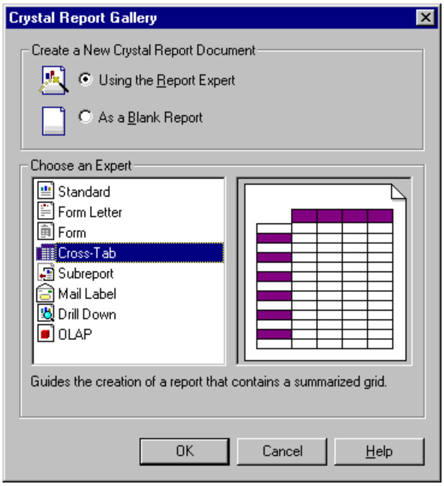
To create an Informix MetaCube Cross-Tab report

1 On the **File** menu, click **New**.

Tip: Another way to do this is to click the New button on the Standard toolbar.

The Crystal Report Gallery appears.

2 Select the **Cross-Tab Expert**.



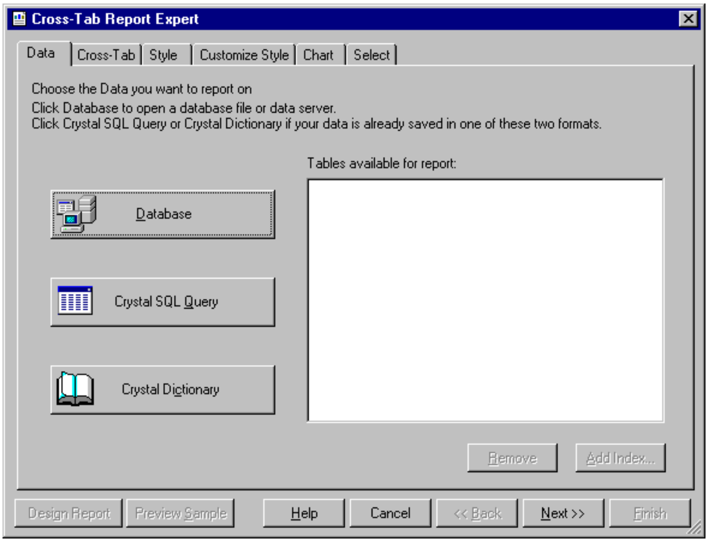
● Using the Report Expert

○ As a Blank Report

Guides the creation of a report that contains a summarized grid.

3 Click **OK**.

The Cross-Tab Report Expert appears.

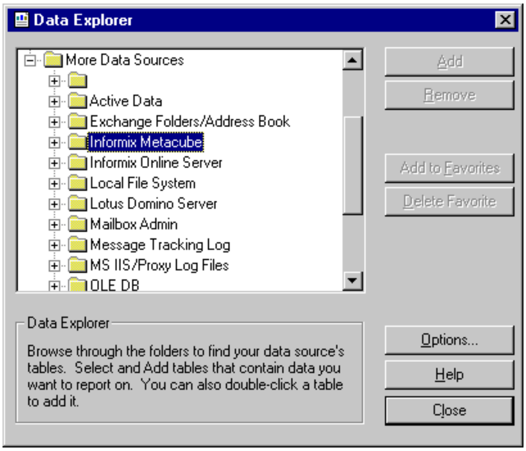


Choose the Data you want to report on
Click Database to open a database file or data server.
Click Crystal SQL Query or Crystal Dictionary if your data is already saved in one of these two formats.

4 Click **Database**.

The Data Explorer appears.

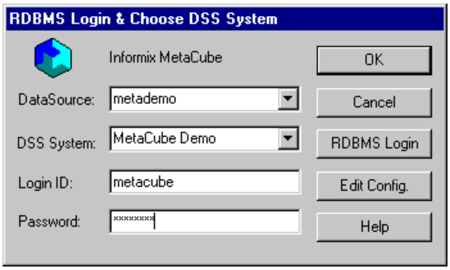
5 Expand the **More Data Sources** folder, then expand the **Informix Metacube** folder.



Data Explorer

Browse through the folders to find your data source's tables. Select and Add tables that contain data you want to report on. You can also double-click a table to add it.

6 In the RDBMS Login & Choose DSS System dialog box, enter the password, then click **OK**.



DataSource: metademo

DSS System: MetaCube Demo

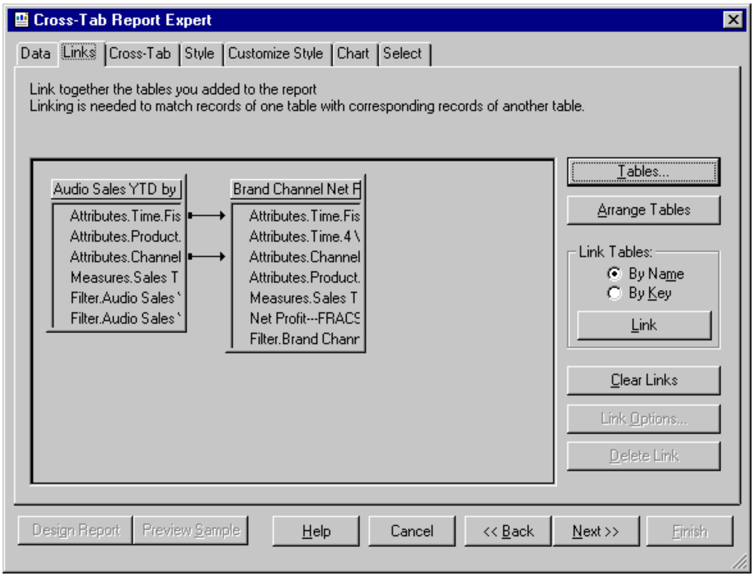
Login ID: metacube

Password: ********

Note: If you have already been working with Informix MetaCube reports during this session, you may not be required to login.

7 Select the **Audio Sales YTD by Channel** and **Brand Channel Net Profit** tables, then click **Add**, then click **Close**.

The Cross-Tab Report Expert appears with the Links tab active.



Cross-Tab Report Expert

Link together the tables you added to the report
Linking is needed to match records of one table with corresponding records of another table.

Audio Sales YTD by: Attributes.Time.Fis, Attributes.Product., Attributes.Channel, Measures.Sales T, Filter.Audio Sales`, Filter.Audio Sales`

Brand Channel Net F: Attributes.Time.Fis, Attributes.Time.4 \, Attributes.Channel, Attributes.Product., Measures.Sales T, Net Profit--FRACS, Filter.Brand Chanr

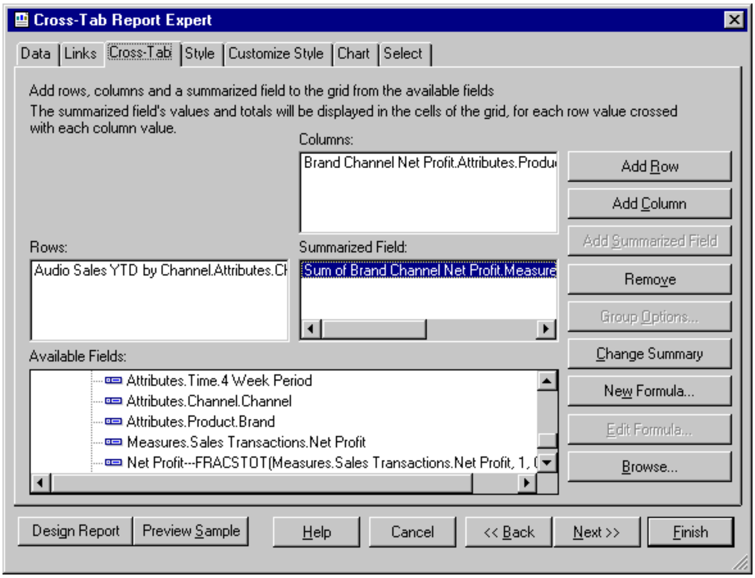
Link Tables: ● By Name ○ By Key

8 Click **Next**.

The Cross-Tab tab appears.

9 Drag and drop the following fields from the Report Fields area to the appropriate Cross-Tab areas:

- **Attributes.Channel.Channel** field from the Audio Sales YTD Channel table to the Rows area
- **Attributes.Product.Brand** field from the Brand Channel Net Profit table to the Columns area
- **Measures.Sales Transactions.Net Profit** field from the Brand Channel Net Profit table to the Summarized Fields area.



Cross-Tab Report Expert

Add rows, columns and a summarized field to the grid from the available fields
The summarized field's values and totals will be displayed in the cells of the grid, for each row value crossed with each column value.

Columns: Brand Channel Net Profit.Attributes.Produ

Rows: Audio Sales YTD by Channel.Attributes.Ch

Summarized Field: Sum of Brand Channel Net Profit.Measure

Available Fields:
- Attributes.Time.4 Week Period
- Attributes.Channel.Channel
- Attributes.Product.Brand
- Measures.Sales Transactions.Net Profit
- Net Profit---FRACSTOT(Measures.Sales Transactions.Net Profit, 1, (

10 Click **Finish** to see a Cross-Tab report based on the fields you selected.

11 Save the report.

This report is used in the adding a field to a Cross-Tab report tutorial.

To add a field to an Informix MetaCube Cross-Tab report

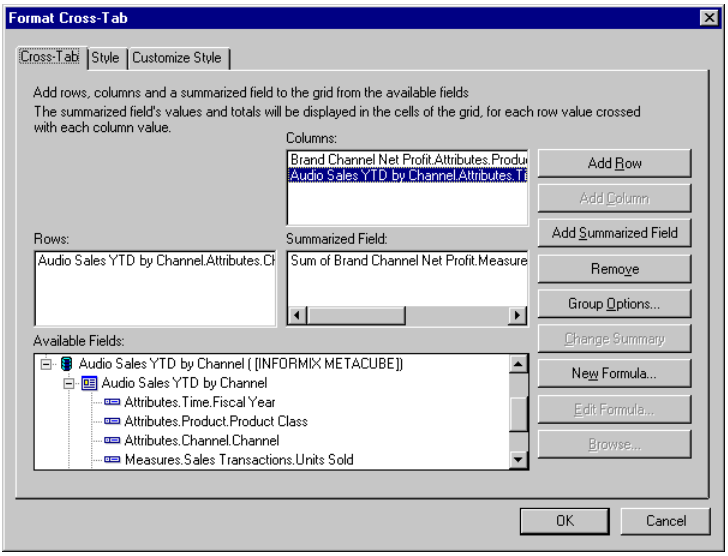
This tutorial is based on the report created in the tutorial for creating Informix MetaCube Cross-Tab reports.

1 Select the entire Cross-Tab object by clicking in the top, left-hand corner of the object above the top row header, to the left of the far left column header.

2 On the **Format** menu, click **Format Cross-Tab**.

The Format Cross-Tab dialog box appears.

3 Drag and drop the **Attributes.Time.Fiscal Year** field from the Audio Sales YTD by Channel table into the Columns area.



Add rows, columns and a summarized field to the grid from the available fields
The summarized field's values and totals will be displayed in the cells of the grid, for each row value crossed with each column value.

Columns:
Brand Channel Net Profit.Attributes.Produ
Audio Sales YTD by Channel.Attributes.Ti

Rows:
Audio Sales YTD by Channel.Attributes.Ch

Summarized Field:
Sum of Brand Channel Net Profit.Measure

Available Fields:
Audio Sales YTD by Channel ([INFORMIX METACUBE])
Audio Sales YTD by Channel
Attributes.Time.Fiscal Year
Attributes.Product.Product Class
Attributes.Channel.Channel
Measures.Sales Transactions.Units Sold

4 Click **OK**.

Fiscal Years is now added to the Cross-Tab report.

Note: You may have to expand the columns to see all of the information.

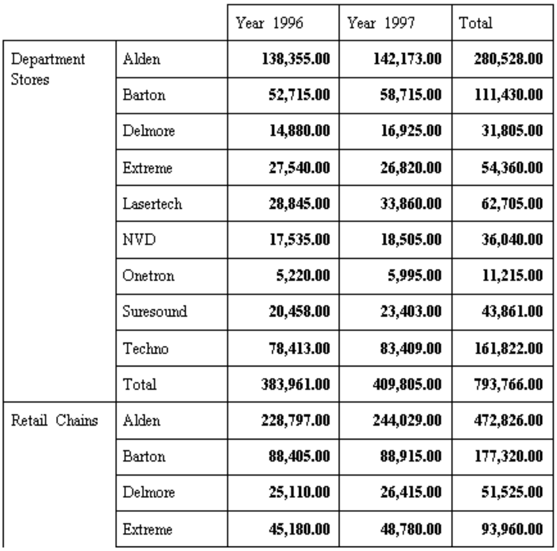
5 Save the report.

This report is used in the pivot fields in an Informix MetaCube Cross-Tab report tutorial.

To pivot fields in an Informix MetaCube Cross-Tab report

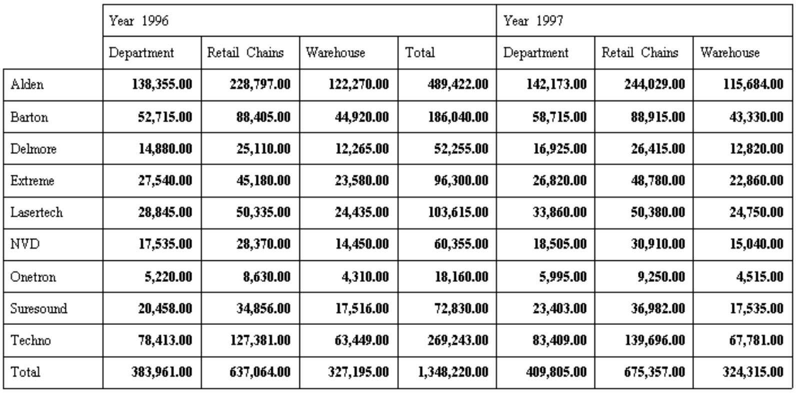
This tutorial is based on the report created in the tutorial on how to add a field to an Informix MetaCube Cross-Tab report. Because of the numerous brands, the report is difficult to work with as far as viewing data or printing. To make it more user friendly, it would be beneficial to switch the fields so that the columns became rows and rows became columns.

1 Click on the top column header “Alden” and drag it so that it is positioned over the row header “Department Stores” and drop it.



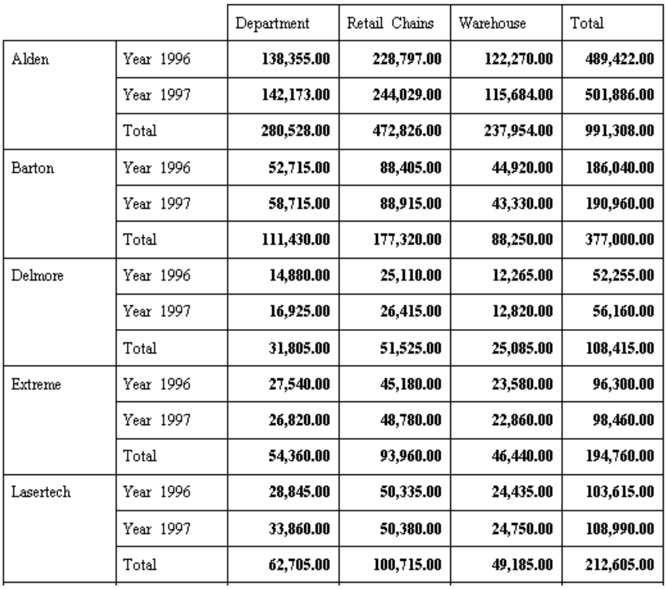
		Year 1996	Year 1997	Total
Department Stores	Alden	138,355.00	142,173.00	280,528.00
	Barton	52,715.00	58,715.00	111,430.00
	Delmore	14,880.00	16,925.00	31,805.00
	Extreme	27,540.00	26,820.00	54,360.00
	Lasertech	28,845.00	33,860.00	62,705.00
	NVD	17,535.00	18,505.00	36,040.00
	Onetron	5,220.00	5,995.00	11,215.00
	Suresound	20,458.00	23,403.00	43,861.00
	Techno	78,413.00	83,409.00	161,822.00
	Total	383,961.00	409,805.00	793,766.00
Retail Chains	Alden	228,797.00	244,029.00	472,826.00
	Barton	88,405.00	88,915.00	177,320.00
	Delmore	25,110.00	26,415.00	51,525.00
	Extreme	45,180.00	48,780.00	93,960.00

2 Now take the row header “Department Stores” and drag it so it is positioned over the column header “Year 1996" and drop it.



	Year 1996				Year 1997		
	Department	Retail Chains	Warehouse	Total	Department	Retail Chains	Warehouse
Alden	138,355.00	228,797.00	122,270.00	489,422.00	142,173.00	244,029.00	115,684.00
Barton	52,715.00	88,405.00	44,920.00	186,040.00	58,715.00	88,915.00	43,330.00
Delmore	14,880.00	25,110.00	12,265.00	52,255.00	16,925.00	26,415.00	12,820.00
Extreme	27,540.00	45,180.00	23,580.00	96,300.00	26,820.00	48,780.00	22,860.00
Lasertech	28,845.00	50,335.00	24,435.00	103,615.00	33,860.00	50,380.00	24,750.00
NVD	17,535.00	28,370.00	14,450.00	60,355.00	18,505.00	30,910.00	15,040.00
Onetron	5,220.00	8,630.00	4,310.00	18,160.00	5,995.00	9,250.00	4,515.00
Suresound	20,458.00	34,856.00	17,516.00	72,830.00	23,403.00	36,982.00	17,535.00
Techno	78,413.00	127,381.00	63,449.00	269,243.00	83,409.00	139,696.00	67,781.00
Total	383,961.00	637,064.00	327,195.00	1,348,220.00	409,805.00	675,357.00	324,315.00

3 To finish, take the “Year 1996" column header and position it over the Alden row header and drop it.



		Department	Retail Chains	Warehouse	Total
Alden	Year 1996	138,355.00	228,797.00	122,270.00	489,422.00
	Year 1997	142,173.00	244,029.00	115,684.00	501,886.00
	Total	280,528.00	472,826.00	237,954.00	991,308.00
Barton	Year 1996	52,715.00	88,405.00	44,920.00	186,040.00
	Year 1997	58,715.00	88,915.00	43,330.00	190,960.00
	Total	111,430.00	177,320.00	88,250.00	377,000.00
Delmore	Year 1996	14,880.00	25,110.00	12,265.00	52,255.00
	Year 1997	16,925.00	26,415.00	12,820.00	56,160.00
	Total	31,805.00	51,525.00	25,085.00	108,415.00
Extreme	Year 1996	27,540.00	45,180.00	23,580.00	96,300.00
	Year 1997	26,820.00	48,780.00	22,860.00	98,460.00
	Total	54,360.00	93,960.00	46,440.00	194,760.00
Lasertech	Year 1996	28,845.00	50,335.00	24,435.00	103,615.00
	Year 1997	33,860.00	50,380.00	24,750.00	108,990.00
	Total	62,705.00	100,715.00	49,185.00	212,605.00

To update the location for an Informix MetaCube using the MetaCube SQL method

1 On the Database menu, click **Set Location**.
The Set Location dialog box appears.

2 Select the Database, then click **Set Location**.
The Data Explorer appears.

3 Select the table.

4 Click **Set**.
The Propagate Set Location dialog box appears.

5 Click **Yes**.

6 Click **Done**.

Distributing and Viewing Reports 18

This chapter provides you with information about how to distribute finished reports using a variety of methods (printing, faxing, exporting). It also includes information about opening reports in Enterprise folders.

Distributing Reports

Crystal Reports enables you to distribute your report using a variety of methods. This chapter covers:

- Printing a report
- Faxing a report
- Exporting a report
- Saving your report in Version 7 format
- Working with Enterprise folders.

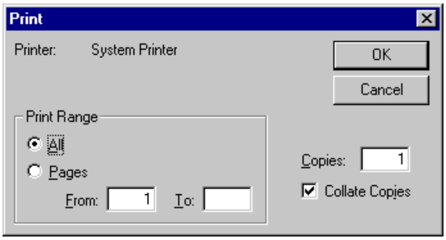
Printing a report

To print a report

1 On the **File** menu, select **Print**, then click **Printer**.

Tip: Another way to do this is to click the Print button on the Standard toolbar.

The Print dialog box appears.



Printer: System Printer

Print Range

● All

○ Pages

From: 1 To:

Copies: 1

☑ Collate Copies

2 Choose the following:

- Print range: all pages or a specific range.
- Number of copies.
- Collate copies: if this option is selected, the report will print each page in order. For example, if you are printing two copies of a report with four pages your report will print page one, two, three, and four of the first copy, then print the second copy.

3 Click **OK**.

The Printing Records dialog box appears showing the progress of your print job.

Faxing 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.

To fax a report

1. On the **File** menu, click **Printer Setup**.

   The Print Setup dialog box appears.
2. Select the fax driver from the **Name** drop-down list.
3. Click **OK**.
4. On the **File** menu, select **Print**, then click **Printer**.

   The Print dialog box appears.
5. Click **OK**.

   Your fax application appears, prompting you to select a cover page and to fill in the appropriate fax information.

Exporting a report

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

The exporting process requires you to specify a format and a destination. The format determines the file type, and the destination determines where the file is located.

Format Types

Crystal Reports provides you with over twenty different export format types. They include:

- Adobe Acrobat (PDF)
- Character-separated values
- Comma-separated values (CSV)
- Crystal Reports (RPT)
- Crystal Reports 7 (RPT)
- Data Interchange Format (DIF)
- Microsoft® Excel
- HTML 3.2
- HTML 4.0 (DHTML)
- Lotus® 1-2-3
- ODBC
- Paginated Text
- Record style (columns of values)
- Report Definition

- Rich Text Format
- Tab-separated text
- Tab-separated values
- Text
- Word Format
- Word for Windows document
- XML.

In addition to the standard export format types installed on your machine you may find additional export format types are available to you. These are determined by the DLL files on your local machine.

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

Destination

The destination determines the export location of your report. Crystal Report enables you to choose one of six destinations. They include:

- Application
- Disk file
- Microsoft® Exchange® folder
- Lotus® Domino™
- Lotus® cc:Mail (VIM)
- Microsoft Mail™ (MAPI).

Exporting to an application

If you export to an application, the program exports the report to a "temp" file in the specified format and then opens the file in the appropriate application.

Note:

- when exporting in ODBC format, "Application" and "Disk file" are the same
- you will need to specify a file path if you are exporting in HTML format
- the file name of the report and the file name of the "temp" file cannot be the same.

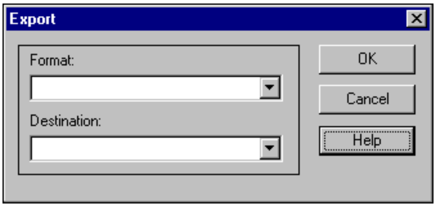
The sections below provide instructions on how to export a report to the Microsoft Excel 7.0 Extended format for each of the different destination types.

To export to an application

1. Open the report you would like to export.
2. On the **File** menu, select **Print**, then click **Export**.

Tip: Another way to do this is to click the Export button on the Standard toolbar.

The Export dialog box appears.



3. Select the export format type from the **Format** drop-down list. In this case, choose **Excel 7.0 (XLS) (Extended)**.
4. Select **Application** from the **Destination** drop-down list.
5. Click **OK**.

   The Format Options dialog box appears.
6. Change the formatting options as needed.
7. Click **OK**.

   The Exporting Record dialog box appears.

Note: Click Cancel Exporting to cancel the export process.

The program exports the report and opens it in the appropriate application. In this example, Microsoft Excel opens with the exported data.

Exporting to a disk file

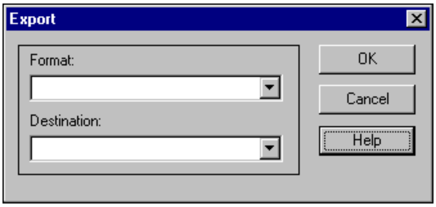
If you export to a disk file, the program saves the report to the disk or diskette you have specified.

To export to a disk file

1. Open the report you would like to export.

   Tip: Another way to do this is to click the Export button on the Standard toolbar.
2. On the **File** menu, select **Print**, then click **Export**.

   The Export dialog box appears.



3 Select the export format type from the **Format** drop-down list. In this case, choose **Excel 7.0 (XLS) (Extended)**.

4 Select **Disk file** from the **Destination** drop-down list.

5 Click **OK**.

The Format Options dialog box appears.

6 Change the formatting options as needed.

7 Click **OK**.

The Choose Export File dialog box appears.

8 Select the appropriate directory.

9 Enter the File name.

10 Click **Save**.

Note: Click Cancel Exporting to cancel the export process.

The program exports the report and saves it in the designated directory. In this example the file is saved in the “temp” directory in an .xls format.

Exporting to a Microsoft Exchange folder

Crystal Reports enables you to export a report file to a Microsoft Exchange folder. You select the folder, and the report is stored there in the format that you specify. A Microsoft Exchange folder can contain standard notes (mail), files, and instances of Microsoft Exchange forms.

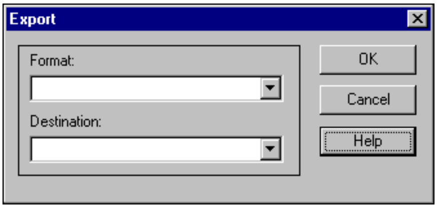
To export to a Microsoft Exchange folder

1 Open the report you would like to export.

2 On the **File** menu, select **Print**, then click **Export**.

Tip: Another way to do this is to click the Export button on the Standard toolbar.

The Export dialog box appears.



3 Select the export format type from the **Format** drop-down list. In this case, choose **Excel 7.0 (XLS) (Extended)**.

4 Select **Exchange Folder** from the **Destination** drop-down list.

5 Click **OK**.

The Format Options dialog box appears.

6 Change the formatting options as needed.

7 Click **OK**.

The Choose Profile dialog box appears.

Note: You will be prompted by the Microsoft Outlook™ Setup Expert if Microsoft Exchange, Microsoft Mail, or Internet E-mail is not set up on your machine.

8 Select the desired profile from the **Profile Name** drop-down list. If the profile is not listed, click **New** to create it.

9 Click **OK** when finished.

10 When the Select a Folder dialog box appears, select the folder in the profile in which you want the report to appear, and click **OK**.

The report is exported to the Microsoft Exchange folder you selected. The exported report can now be accessed through the Microsoft Exchange client.

Exporting to Lotus Domino

You must have Version 3.0 or later of the Lotus Domino client. You will also require, at a minimum, depositor access. Crystal Reports will not export to a Lotus Domino OS/2 client.

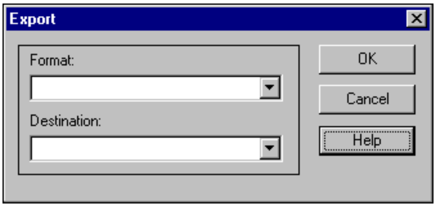
To export to Lotus Domino

1 Open the report you would like to export.

2 On the **File** menu, select **Print**, then click **Export**.

Tip: Another way to do this is to click the Export button on the Standard toolbar.

The Export dialog box appears.



3 Select the export format type from the **Format** drop-down list. In this case, choose **Excel 7.0 (XLS) (Extended)**.

4 Select **Lotus Domino Database** from the **Destination** drop-down list.

5. Click **OK**.

   The Format Options dialog box appears.
6. Change the format settings as required.
7. Click **OK**.

   The Select Database dialog box appears.
8. Double-click the Lotus Domino server you would like to export your report to.

   The file name defaults.
9. Select the database you would like to export the report to.
10. Click **OK**.

    The Comments dialog box appears.
11. Type in any comments that are to appear when another user selects your report from the Lotus Domino Desktop.
12. Click **OK**.

    The export process begins.

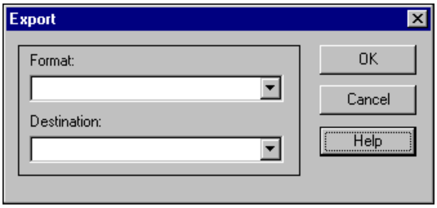
The next time a user logs onto the Lotus Domino database you specified, they will see the report in their desktop. The user can double-click the report file name to display the comments you wrote and then double-click the report icon to view the report.

Exporting to Microsoft Mail (MAPI)

1. Open the report you would like to export.
2. On the **File** menu, select **Print**, then click **Export**.

   Tip: Another way to do this is to click the Export button on the Standard toolbar.

   The Export dialog box appears.



3. Select the export format type from the **Format** drop-down list. In this case, choose **Excel 7.0 (XLS) (Extended)**.
4. Select **Microsoft Mail (MAPI)** from the **Destination** drop-down list.
5. Click **OK**.

   The Format Options dialog box appears.

6 Change the formatting options as needed.

7 Click **OK**.
The Send Mail dialog box appears.

8 Enter the address details, then click **Send**.
The Exporting Record dialog box appears.

Note: Click Cancel Exporting to cancel the export process.

Saving your report in Version 7 format

To enable people who are using Version 7 of Crystal Reports to make changes in the report you have created in Version 8, save your report as Crystal Reports (Version 7).

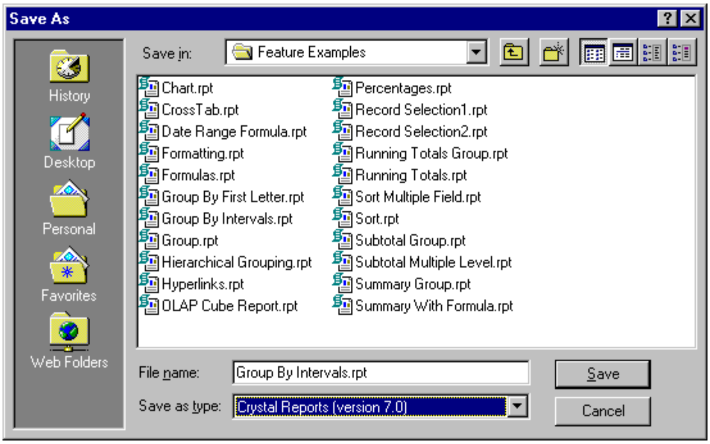
To save your report in Version 7 format

1 On the **File** menu, click **Save As**.
The Save As dialog box appears.

2 Choose a folder.

3 Enter the File name.
Check to make sure it has an .rpt extension.

4 From the **Save as type** drop-down list, select **Crystal Reports (Version 7)**.



Save in: Feature Examples

Chart.rpt
CrossTab.rpt
Date Range Formula.rpt
Formatting.rpt
Formulas.rpt
Group By First Letter.rpt
Group By Intervals.rpt
Group.rpt
Hierarchical Grouping.rpt
Hyperlinks.rpt
OLAP Cube Report.rpt
Percentages.rpt
Record Selection1.rpt
Record Selection2.rpt
Running Totals Group.rpt
Running Totals.rpt
Sort Multiple Field.rpt
Sort.rpt
Subtotal Group.rpt
Subtotal Multiple Level.rpt
Summary Group.rpt
Summary With Formula.rpt

File name: Group By Intervals.rpt

Save as type: Crystal Reports (version 7.0)

5 Click **Save**.

Note: Crystal Reports Version 8 features are not supported in Version 7. You may not want to save your report in Version 7 format if you are using these features. For example, if your report uses formulas created with Basic syntax, the report will not compile correctly if you save in 7.0 format.

Working with Web Folders

You have the ability to open and save reports using Web Folders if you:

- are running Windows 2000 or have Office 2000 installed
- have access to a web server that is configured to support Web Folders
- add a Web Folder from this server into your Web Folders folder.

To open your report

1. On the **File** menu, click **Open**.
   The Open dialog box appears.
2. Click **Web Folders**.
3. Open the folder that contains the report.
4. Double-click to open the report.

To save your report

1. On the **File** menu, click **Save As**.
   The Save As dialog box appears.
2. Click **Web Folders**.
3. Locate the folder you would like to save the report to.
4. Enter the file name.
5. Click **Save**.

Note: After making changes to a report in a Web Folder, you must save the changes to the same file in the same Web Folder.

Working with Enterprise folders

You can open reports from Crystal Enterprise folders in Crystal Reports. If Crystal Reports detects Crystal Enterprise on your machine, it displays the appropriate folders in the Open/Save As dialog box.

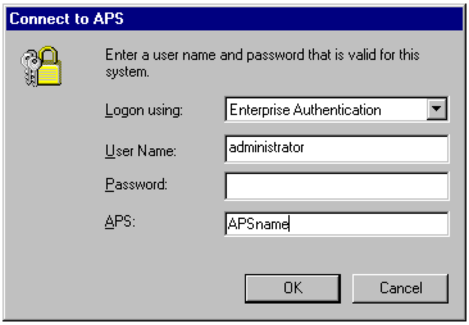
To open a report in an Enterprise folder

1 On the **File** menu, click **Open**.

2 In the Open dialog box, click **Enterprise Folders**.

Note: The Enterprise Folders button is not visible if Crystal Reports doesn't detect Crystal Enterprise on your machine.

If you have not already logged on to an Automated Process Scheduler (APS), the Connect to APS dialog box appears.



Connect to APS

Enter a user name and password that is valid for this system.

Logon using: Enterprise Authentication

User Name: administrator

Password:

APS: APSname

Tip: You can also log on to an APS from the File menu before you open a report.

3 Click the **Logon using** list to select Enterprise Authentication or Windows NT Authentication.

Enterprise Authentication requires a user name and password that is recognized by Crystal Enterprise.

Windows NT authentication requires a user name and password that is recognized by Windows NT.

4 Enter your user name and password.

5 In the **APS Name** field, enter the name of the APS you want to connect to.

6 Click **OK**.

The Enterprise folders appear and you can select a report to open in Crystal Reports.

Note: After making changes to a report in an Enterprise folder, you must save the changes to the same file in the same Enterprise folder.

Report Alerts 19

This chapter provides information about creating and using alerts in your Crystal reports.

About Report Alerts

Report Alerts are custom messages created in Crystal Reports that appear when certain conditions are met by data in a report. Report Alerts may indicate action to be taken by the user or information about report data.

Report Alerts are created from formulas that evaluate conditions you specify. If the condition is true, the alert is triggered and its message is displayed. Messages can be text strings or formulas that combine text and report fields.

Once a Report Alert is triggered, it's not evaluated again until you refresh your report's data.

Because Report Alerts are specific to each report, you decide when to use them and when not to. They can be useful to point out important information, such as sales that fall above or below a limit. And the message is created by you, so it can be specific to your data.

Working with Report Alerts

This section of the guide focuses on the tasks you'll need to undertake in order to use Report Alerts:

- "Creating Report Alerts" on page 350.
- "Editing Report Alerts" on page 353.
- "Deleting Report Alerts" on page 353.
- "Viewing Report Alerts" on page 353.
- "Referring to Report Alerts in formulas" on page 354.

Creating Report Alerts

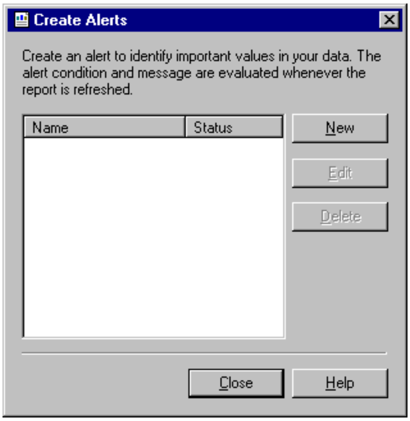
You must complete three steps when creating a Report Alert:

- name the alert
- define the condition that triggers the alert
- create the message you want to appear when the alert is triggered (this step is optional).

To create a Report Alert

1 On the Report menu, click **Create Alerts**.

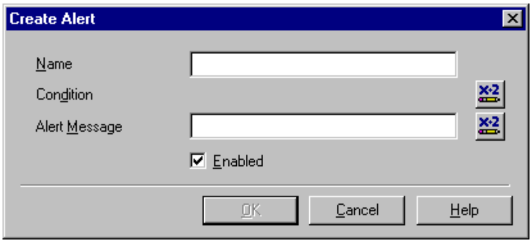
The Create Alerts dialog box appears.



Create an alert to identify important values in your data. The alert condition and message are evaluated whenever the report is refreshed.

2 Click **New**.

The Create Alert dialog box appears.



☑ Enabled

3 Enter a name for your new alert in the **Name** box.

4 Enter your alert message in the **Message** box.

The Message box lets you enter a message to be used as a default. If you want the same message to appear every time your alert is triggered, enter it in the Message box.

If, however, you want to use a formula so the message is customized with data elements, see the next step.

5 If you want to use a formula to create an alert message, click the formula button to the right of **Message**.

The Alert Message Formula Editor appears. This editor is the same as the Formula Editor; for information on how to use it, see "Working with the Formula Editor" on page 387.

6 Enter your alert message formula.

For example, if you want to see the message “Country is a star performer” (where Country is the name of a specific country), you might create the following formula:

```
GroupName ({Customer.Country}) + “ is a star performer”
```

Note:

- The result of an alert message formula must be a string.
- The DefaultAttribute function can be used to refer to a message added in the Message box. For example, if the message in your Message box is “ is a star performer,” your alert message formula might be:

```
GroupName ({Customer.Country]) + DefaultAttribute
```

- Message condition formulas can be created using either Crystal Syntax or Basic Syntax.

7 Click **Condition**.

The Alert Condition Formula Editor appears. This editor is like the Alert Message Formula Editor—including its ability to base formulas on recurring fields.

8 Enter your alert condition formula.

Alert formulas can be based on recurring records or on summary fields, but cannot be based on print-time fields, such as running totals or print time formulas. Alert formulas cannot have shared variables.

If an alert formula is based on a summary field, any recurring fields used must be constant over the summary field. For example, if you are grouping on Country, Region, and City, you might create an alert such as:

```
Sum ({Customer.Last Year’s Sales}, {Customer.Region})
```

In this case, your formula can refer to either Country or Region, but not City or Customer Name since these are not constant.

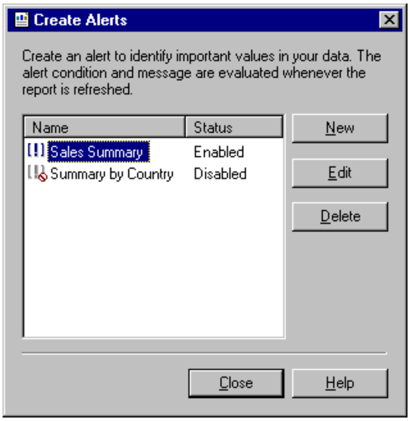
Note: Alert condition formulas can be created using either Crystal Syntax or Basic Syntax.

9 Clear the **Enable** check box if you do not want the alert to be evaluated.

Otherwise, leave it selected.

10 Click **OK** to save your alert.

You are returned to the Create Alerts dialog box and your new alert is listed. You can see its name and status (Enabled or Disabled).



Create an alert to identify important values in your data. The alert condition and message are evaluated whenever the report is refreshed.

Name	Status
Sales Summary	Enabled
Summary by Country	Disabled

Only enabled and disabled alerts appear in the Create Alerts dialog box. If an alert is triggered, it is seen in the View Alerts dialog box.

Editing Report Alerts

1. On the Report menu, click **Create Alerts**.
2. In the Create Alerts dialog box, select the alert you want to edit and click **Edit**.

   Tip: Double-clicking an alert also lets you edit it.
3. Make the changes you want in the Edit Alert dialog box.
4. Click **OK** to save your changes.

Deleting Report Alerts

1. On the Report menu, click **Create Alerts**.
2. In the Create Alerts dialog box, select the alert you want to delete and click **Delete**.

   The selected alert is removed from the Create Alerts dialog box.

Viewing Report Alerts

You can view triggered Report Alerts by:

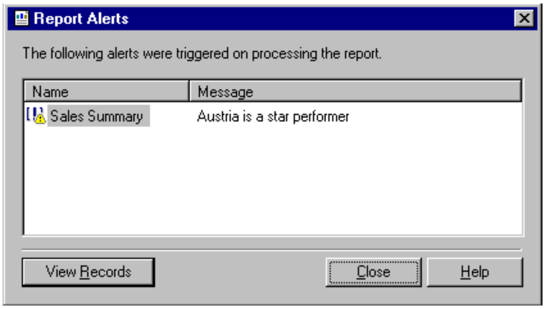
- refreshing your report's data
- selecting View Alerts from the Report menu.

Note: If you want to view alerts when report data is refreshed, you must select "Display Alerts on Refresh" on the Reporting tab of the Options dialog box (this option is also available on the Report Options dialog box).

To view Report Alerts

1. On the **Report** menu, click **View Alerts**.

   The Report Alerts dialog box appears.



Report Alerts

The following alerts were triggered on processing the report.

Name	Message
Sales Summary	Austria is a star performer

2. Select the alert whose records you want to see.
3. Click **View Records**.

   The report record that triggered the alert is shown. If the record is hidden, the record's group is shown but drill down is not performed.
4. Click **Reset** to return to the complete report.

Referring to Report Alerts in formulas

Alerts can be referred to in formulas. Any formula that references an alert becomes a print time formula.

The following functions are available:

- IsAlertEnabled( )
- IsAlertTriggered( )
- AlertMessage( )

These functions behave the same as alerts created in the Create Alerts dialog box:

- IsAlertTriggered("AlertName") is true only for the records on which the alert is triggered
- AlertMessage("AlertName") displays the message for a record when the alert is true.

Because alerts are not field objects (you cannot drop them into the report), alerts are represented differently from report fields in the formula editor. In the Functions tree, you'll see an Alerts heading where the functions themselves appear. Available alerts are listed by name below that.

XML 20

This chapter describes the XML data access and export capabilities of Crystal Reports. It begins with a brief overview of XML technology and then provides procedures for setting up XML data access and using the XML exporting features.

XML overview

XML (Extensible Markup Language) is a format for exchanging data. The format is not owned by any one organization, but is accepted by a host of companies as a standard for data interchange. XML facilitates moving structured data across the web by being independent of system or platform restrictions.

XML is the transmission format of B2B (Business to Business) and B2C (Business to Consumer), because it enables information exchange and makes it easy to publish report information on numerous platforms and devices.

Unlike HTML, which stores a limited amount of formatting with the content, XML content is stored without formatting—formatting information can be saved to a style sheet. XML content is structured to make it accessible to a wide range of users, applications, and platforms.

There are many advantages to using XML:

- XML files are reliable to use and easy to maintain because they contain only content.
- Multiple output formats can be supported simultaneously. From a single XML document, you can generate HTML for a browser, a PDF for printing, and so on.
- You can convert XML to new formats without affecting a file's content.
- Because XML uses style sheets to store formatting information, the look and feel of a web site can be changed easily.
- XML is easy to validate, so you're less likely to have incorrect information or corrupt files on your web site.
- XML makes searching through large numbers of files easier and more accurate due to its highly structured design.
- XML files can store content in multiple languages. An XML translator displays content for specific language preferences.

There are some drawbacks to consider as well:

- XML files can be very large and don't naturally support binary data, such as images.
- Existing content must be reworked to take advantage of XML's benefits and to reliably support the wide range of new devices.
- XML standards are still developing, so there's risk in selecting a particular methodology.
- Currently, there aren't many graphical content management tools available.

XML elements, tags, and structure

XML content is made up of elements that describe data. Each element usually contains two tags: one to open the element and one to close it (empty elements can be opened and closed in the same tag). You define the tags to suit your company's needs. For example, you might create a tag `<LastYearsSales>` to describe last year's sales totals. Any other tag is equally appropriate as long as it describes the data to your satisfaction and is used consistently throughout the XML file. XML tags are case sensitive and can contain additional information called attributes.

XML facilitates emulation of the hierarchical structure of documents by allowing elements to contain other elements. For example, you might show the structure of report data by grouping elements as follows:

```
<FormattedReportObject>
   <ObjectName>Field2</ObjectName>
   <FormattedValue>Australia</FormattedValue>
   <Value>Australia</Value>
</FormattedReportObject>
```

In this example, the `<FormattedReportObject>` element contains three other elements that describe the data within. This grouping of elements becomes, in essence, the structure of the XML content. You can add as many layers of grouping as needed to create an appropriate structure.

XML Schema

The XML Schema is a specification from the World Wide Web Consortium (W3C) that provides a way of defining the structure, content, and interpretation of XML documents. The XML Schema defines the data types that can be associated with XML elements and attributes. It also describes the structure of the contents of XML documents, and defines rules for governing schema validation. Based on these features, the XML Schema is able to enforce data integrity and compatibility among XML documents.

Because XML schemas are XML documents themselves, they can be managed by XML authoring tools.

Document Type Definition

The Document Type Definition (DTD), like the XML Schema, is a set of rules used to describe elements and attributes in an XML document. As a means of validation, the DTD originated in SGML. Because the DTD uses a syntax different from XML and has limitations in functionality, the XML Schema has begun to replace it.

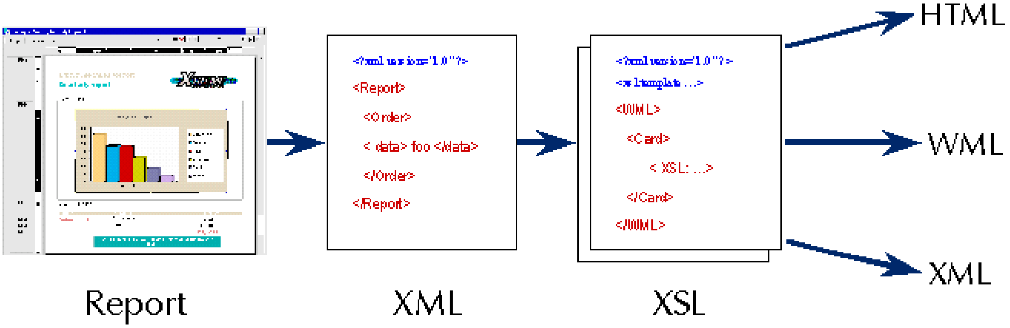
A DTD can be included in an XML file, or can be a separate file referenced by an XML file.

Style sheets

Style sheets provide formatting information for the presentation and layout of XML content. Like the DTD, style sheets can be included in an XML file, or can be a separate file referenced by an XML file.

XSL (Extensible Stylesheet Language) is the preferred type of style sheet for XML. XSL provides both a method for formatting XML documents and a method for transforming XML documents into formats recognizable to a browser.

For example, you might use XSL to transform a report's XML output so it can be read as Wireless Markup Language (WML) for use with a hand-held device.

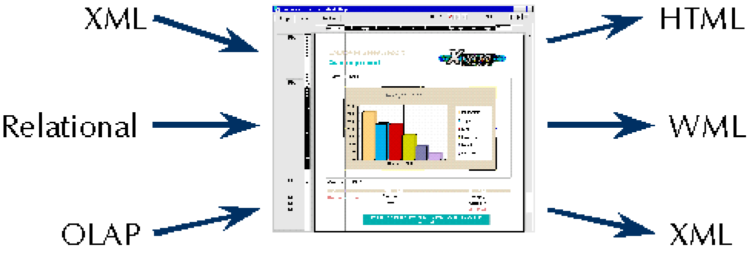


```
<?xml version="1.0"?>
<Report>
  <Order>
  < data> foo </data>
  </Order>
</Report>
```

```
<?xml version="1.0"?>
<xsl:template ...>
<WML>
  <Card>
      < XSL: ...>
  </Card>
</WML>
```

XML in Crystal Reports

In Crystal Reports, XML is both a data source and an export format. You can create a Crystal report from an XML data source, thus turning raw XML data into report information. You can also take a Crystal report and export it to XML, allowing you to transform the data to HTML, WML, and so on.



Each of the XML-related tasks you can carry out in Crystal Reports is described in the following sections:

- "Accessing XML data" on page 359.
- "Exporting to XML format" on page 363.
- "Customizing XML report definitions" on page 365.

Accessing XML data

Crystal Reports provides XML data access through a Merant DataDirect ODBC driver named Crxml15.dll. This driver is installed in the Windows\System or Winnt\System32 directory. Using the Merant driver, you can connect to an XML data source of your choice and design reports as you would from any other data source.

Note: Using the Merant driver to connect to an XML data source has some limitations. See "Merant XML driver limitations" in the online help for more information.

An ODBC data source, named CRXMLV36, is created when you install Crystal Reports. You can also set up an XML data source in the ODBC Data Source Administrator.

Setting up an XML data source

To set up an XML data source, you must use the Merant ODBC driver, Crxml15.dll. The driver supports access to the following XML file types:

- IE5 XML Data Islands
- ADO 2.5 persisted XML
- XML files with or without XML schemes.

Note: This section provides a generic description of how to set up an XML data source. For examples of how to set up data sources for IE5 XML Data Islands and ADO 2.5 persisted XML, visit the Crystal Reports site at:

`http://www.seagatesoftware.com/products/crystalreports`

The Merant driver requires an XML file structure that can be read like a database. Within an XML file, the first element encountered is considered the "table." Child elements of the node are considered the "rows." Child elements of the row nodes are considered "columns." Child elements of columns are considered part of the columns themselves. In the driver setup, you can use a Table Hint and/or a Row Hint to identify XSL patterns to be used as table and row nodes.

Table Hint

A Table Hint specifies an XSL pattern that indicates a table or rowset. Table Hints should point to a single node; otherwise, the first node in a set of nodes is used as the table node.

Row Hint

A Row Hint specifies an XSL pattern that indicates nodes to be used as rows in the table. The Row Hint also specifies whether the table content is based on an XML element or attribute type.

Specifying Table and Row Hints

The Merant driver assumes that row nodes are the immediate children of the table node:

- If you specify only a Table Hint, the row nodes are the children of the node the hint resolves to. It is assumed that all of the child nodes have the same name.
- If you specify only a Row Hint, the table node is the parent of the node the hint resolves to. If the Row Hint resolves to a set of nodes, the nodes in that set must all have the same parent.
- If you specify both a table hint and a row, the row hint is taken to be relative to the node the table hint resolves to.

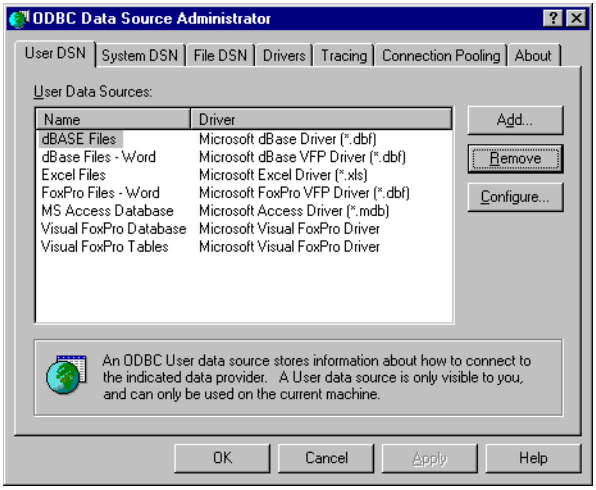
For examples of how to use hints, see “Table and Row Hint examples” in the online help.

To set up an XML data source

1 From the Control Panel, double-click **ODBC Data Sources**.

The ODBC Data Source Administrator dialog box appears.

Note: Crystal Reports installs version 3.510 of the ODBC Data Source Administrator. If you have a different version, steps in this procedure will vary slightly.

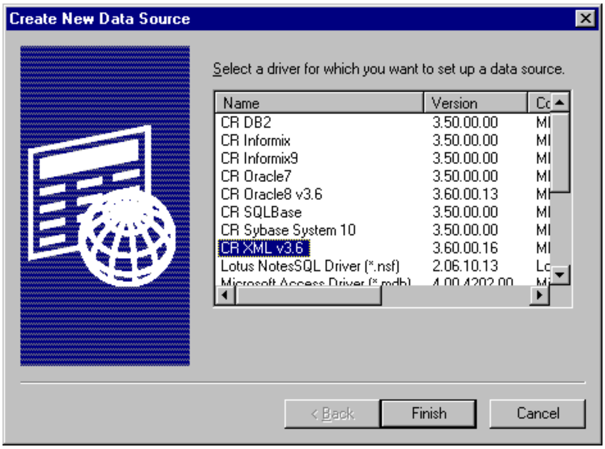


User Data Sources:

Name	Driver
dBASE Files	Microsoft dBase Driver (*.dbf)
dBase Files - Word	Microsoft dBase VFP Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
FoxPro Files - Word	Microsoft FoxPro VFP Driver (*.dbf)
MS Access Database	Microsoft Access Driver (*.mdb)
Visual FoxPro Database	Microsoft Visual FoxPro Driver
Visual FoxPro Tables	Microsoft Visual FoxPro Driver

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

2 Click **Add** to add a new ODBC data source.
The Create New Data Source dialog box appears.

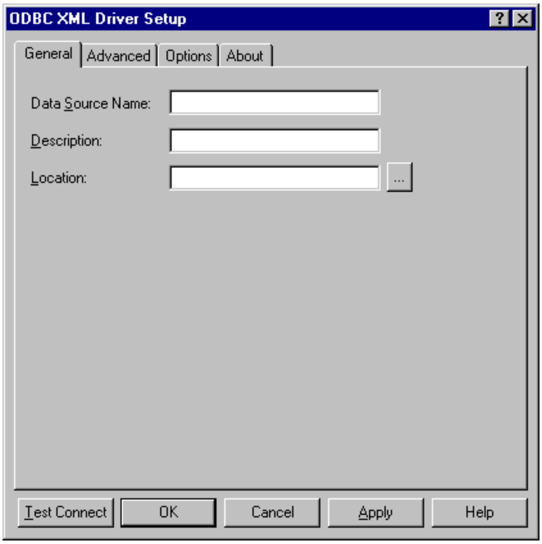


Select a driver for which you want to set up a data source.

Name	Version	Co
CR DB2	3.50.00.00	MI
CR Informix	3.50.00.00	MI
CR Informix9	3.50.00.00	MI
CR Oracle7	3.50.00.00	MI
CR Oracle8 v3.6	3.60.00.13	MI
CR SQLBase	3.50.00.00	MI
CR Sybase System 10	3.50.00.00	MI
CR XML v3.6	3.60.00.16	MI
Lotus NotesSQL Driver (*.nsf)	2.06.10.13	Lo
Microsoft Access Driver (*.mdb)	4.00.4202.00	Mi

3 Choose **CR XML v3.6** from the list.

4 Click **Finish**.
The ODBC XML Driver Setup dialog box appears.



Data Source Name:

Description:

Location:

5 On the **General** tab, enter a name and description for the XML data source.

6 Search for your XML data source, or enter its URL or directory path directly.

7 On the **Advanced** tab, enter a Table Hint and/or a Row Hint.

For examples of how to use hints, see “Table and Row Hint examples” in the online help.

For more information about other options on the Advanced and Options tabs, click the Help button on the ODBC XML Driver Setup dialog box.

8 Click **OK**.

You are returned to the ODBC Data Source Administrator. Your new XML data source is listed as a User Data Source.

9 Click **OK** when finished.

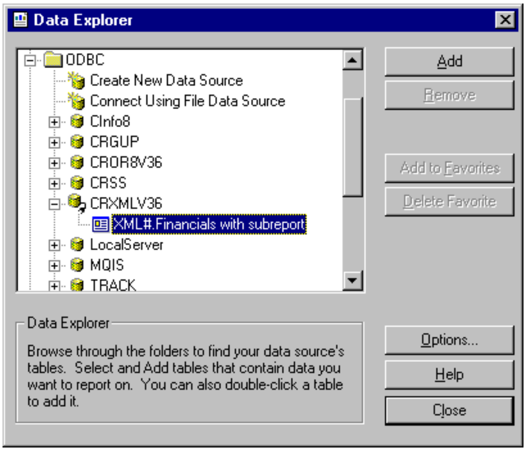
Selecting an XML data source

Once you’ve created an XML data source, it’s available in the Data Explorer in the ODBC folder.

XML data sources can be selected when creating new reports, or when working with the options on the Database menu.

To select an XML data source

1 Open the Data Explorer dialog box by creating a new report, or adding a database to an existing report.



Browse through the folders to find your data source's tables. Select and Add tables that contain data you want to report on. You can also double-click a table to add it.

2 Choose the XML data source you want to open from the ODBC folder and click **Add**.

If additional information is required to log on, another dialog box appears.

3 Type in the log on information you usually use to access this database and click **OK**.

If you did not specify an XML data source within the ODBC setup, the ODBC XML Driver Connection dialog box appears. Type the URL or directory of the XML file, enter an ID and password, and then click OK.

Exporting to XML format

The Crystal Reports XML export option lets you create raw XML data from your Crystal reports. With this data, you can apply a style sheet or format for use in another application. For example, you might apply a format that renders your XML data as an HTML file for distribution over the web.

Note: The XML export option does not render your Crystal report as an XML report. Rather, it creates raw XML data for you to manipulate as you need.

You can customize the structure of your XML data when exporting a report. For information on how to do this, see "Customizing XML report definitions" on page 365.

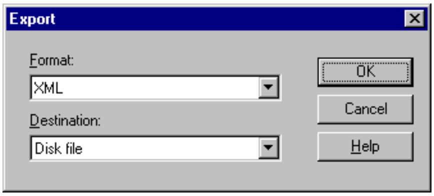
To export a report to XML

1 Open the report you want to export.

2 On the **File** menu, select **Print**, and then click **Export**.

Tip: Another way to do this is to click the Export button on the Standard toolbar.

The Export dialog box appears.



Export

Format: XML

Destination: Disk file

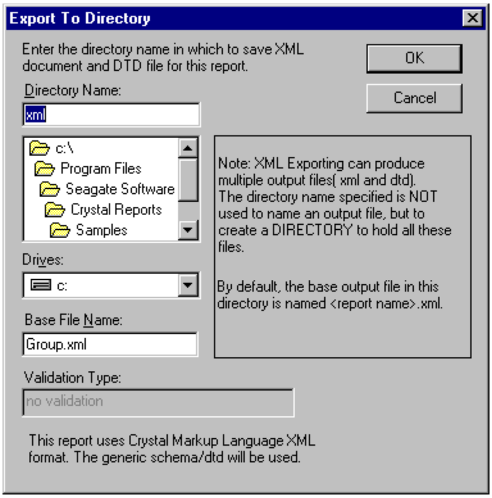
3 Select the export format type from the **Format** list. In this case, choose **XML**.

4 Select **Disk File** from the **Destination** list.

Note: Since XML is an ODBC format, "Application" and "Disk file" provide the same result.

5 Click **OK**.

The Export To Directory dialog box appears.



Export To Directory

Enter the directory name in which to save XML document and DTD file for this report.

Directory Name: xml

Drives: c:

Base File Name: Group.xml

Validation Type: no validation

Note: XML Exporting can produce multiple output files(xml and dtd). The directory name specified is NOT used to name an output file, but to create a DIRECTORY to hold all these files.

By default, the base output file in this directory is named <report name>.xml.

This report uses Crystal Markup Language XML format. The generic schema/dtd will be used.

6 Enter the name of the directory you want to store the XML file(s) in.

You can specify an existing directory on any drive by choosing from the boxes below the directory name.

7 Enter a base file name for your XML file(s).

Both the output XML file and its XSD or DTD—if you've specified an external schema or DTD on the Options tab of the XML Format dialog box—use this base file name.

Note: The Validation Type box indicates the validation choice you made on the options tab of the XML Format dialog box. You cannot change the validation type on the Export to Directory dialog box.

8 Click **OK** when finished.

The report is exported to the directory you selected. The exported report can now be accessed as raw XML data.

Note: Crystal Reports can also export reports to XML format from a browser using the URL command line. For information about how to do this, visit the Crystal Reports site at:

`http://www.seagatesoftware.com/products/crystalreports`

Customizing XML report definitions

Crystal Reports automatically provides an XML report definition for each report you create or edit. The nature of the changes you can make to this default definition depends on the output format you select on the Options tab of the XML Format dialog box:

- **Crystal ML Schema**

  The Crystal ML Schema is the default format. With it, you can create new attributes for existing XML elements, but you cannot make other kinds of changes.

  The advantage of using this format is that you don't have to create your own schema or DTD. The Crystal ML Schema is Crystal Reports' standard schema and is published at the following address:

  `http://www.seagatesoftware.com/xml/schema.xsd`

- **Custom Format**

  The Custom Format option lets you change element names, create attributes, and suppress XML tags. XML output created with this format is not controlled by the Crystal ML Schema; a custom DTD or schema is created on the fly.

  The advantage of using this format is that you're free to create your own XML format. You can rename XML elements to suit an existing DTD or schema, and you can suppress XML tags to change the hierarchy of your output—which, in turn, helps minimize data output.

You are able to select between these output format options as you like. If, however, you change from one option to the other after you have made modifications to element names and/or attributes, you will lose these changes—element names and attributes will be reset to the default values.

Note: The element tags created for reports using the Crystal ML Schema and the Custom Format are not the same. For more information about the differences, see "XML output specifications" on page 370.

Why create your own XML format?

Although the Crystal ML Schema provides a solid default format for exporting to XML, you might want to create your own XML format for various reasons:

- You need to incorporate data from a Crystal report into an existing XML document that has its own schema/DTD and tags. In this case, you would customize the Crystal report's XML data with tags that are consistent with your existing XML document.
- You have an application that needs to read XML data from Crystal Reports, but the application can identify only a specific set of tags. In this case, you would customize the Crystal report's XML tags to match the tags that the application can identify.
- You already use an industry standard XML schema/DTD that is required by your company. In this case, you would customize your Crystal Reports XML data so that it conforms to the standard you are required to use.

Customizing XML tags

Use the customization options in the XML Format dialog box when the report definition created by Crystal Reports doesn't provide the XML information you need. Customization can range from defining a few new XML element attributes to renaming elements and suppressing tags for fields you don't want to export data for.

When you choose Crystal ML Schema, you can carry out these tasks in the XML Format dialog box:

- view the report definition tree
- view XML element names and attributes
- add, edit, and delete XML tag attributes.

When you choose Custom Format, you can carry out these additional tasks:

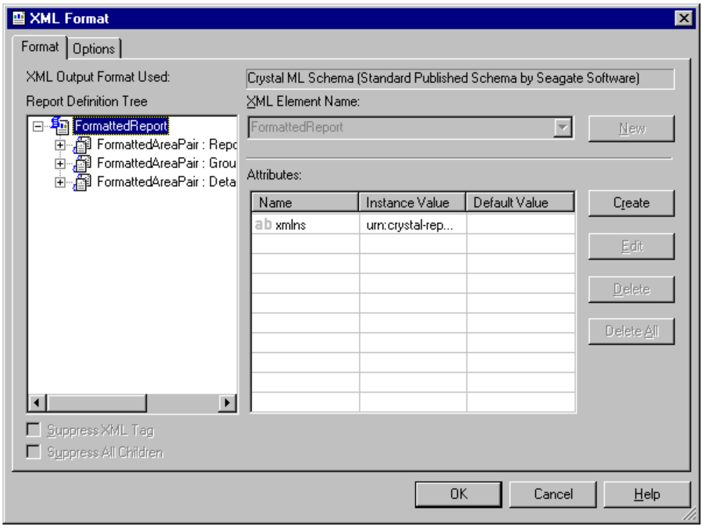
- rename XML tags
- suppress XML tags.

Note: You cannot add new report objects (tree nodes) while using the XML Format dialog box. If you want to create a new node, add a report object to your Crystal report, and it will be added to the Report Definition Tree.

To customize XML tags

1. Open a report in Crystal Reports.
2. On the **Format** menu, click **XML Format**.

   The XML Format dialog box appears with the report's definition tree displayed.



XML Output Format Used: Crystal ML Schema (Standard Published Schema by Seagate Software)

XML Element Name: FormattedReport

Attributes:

Name	Instance Value	Default Value
ab xmlns	urn:crystal-rep...	

Tip: The XML Output Format Used box displays the format your report definition is using. Click the Options tab to select a different option.

3 Expand the Report Definition Tree to find the report object whose XML tag you want to customize.

Note:

- Parent nodes can represent areas or sections of the report. For example, each report group has its own parent node.
- First-level child nodes can represent area pairs, such as a Report Header and a Report Footer.

Changing XML tag names

This option applies only to report definitions that use Custom Format.

1 Select a report object in the Report Definition Tree.

2 In the **XML Element Name** box, type your changes to the XML element name.

3 Click elsewhere in the Report Definition Tree to see the new element name.

When you export the report in XML format, the new tag name you assigned will be used.

Note: Two or more report objects in the Report Definition Tree can be associated with the same XML element. Use the New button (next to the XML Element Name list) to create a new XML element for the selected report object.

To create a new XML element

1 In the Report Definition Tree, select a report object that shares an XML element with another report object.

The New button is activated.

2 Click **New** and enter a new XML element name.

3 Click elsewhere in the Report Definition Tree to see the new element name.

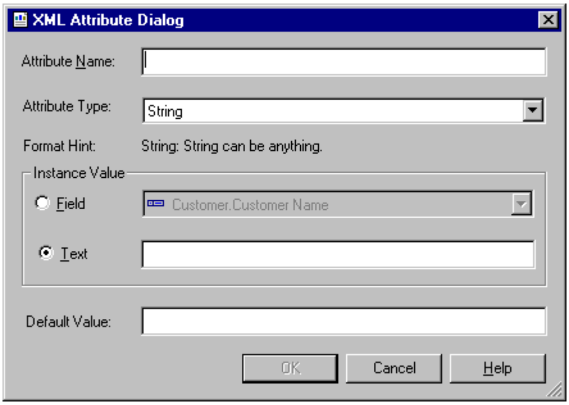
Adding XML attributes

This option applies to report definitions that use Crystal ML Schema or Custom Format.

1 Select a report object in the Report Definition Tree.

2 Click **Create**.

The XML Attribute dialog box appears.



Attribute Name:

Attribute Type: String

Format Hint: String: String can be anything.

Instance Value

○ Field Customer.Customer Name

● Text

Default Value:

3 In the **Attribute Name** box, enter a name for the attribute you are creating.

Note: Each attribute in an element must have a unique attribute name. Attribute names, however, don't have to be unique across elements.

4 In the Attribute Type list, click the appropriate option for the attribute you're creating:

- String
- Decimal
- Boolean
- Date
- Time
- Date Time.

5 In the Instance Value area, select **Field** or **Text**.

Field lets you select a report field of the specified type as your instance value. The Field option is available only if the corresponding report object is not Report (root node) or an Area Pair type.

Text lets you enter text in the associated box as an instance value for an element.

6 In the Default Value box, enter a value if you want to assure the attribute is not left empty when there is no instance value.

7 Click **OK**.

The OK button is not available until an attribute name and instance or default value is entered.

You are returned to the XML Format dialog box. Your new attribute appears in the Attributes list.

Editing XML attributes

This option applies to report definitions that use Crystal ML Schema or Custom Format.

Note: You can edit only those attributes you added using the XML Attribute dialog box.

1. Select an element in the Report Definition Tree that has attributes.
2. Select the attribute you want to edit in the XML Tag Attribute List and click **Edit**.

   Tip: Attributes prefaced with the letters "ab" in blue are attributes you can edit. Those prefaced with "ab" in grey cannot be edited.
3. Make your changes in the XML Attribute dialog box.
4. Click **OK** to save changes.

Deleting XML attributes

This option applies to report definitions that use Crystal ML Schema or Custom Format.

Note: You can delete only those attributes you added using the XML Attribute dialog box.

1. Select an element in the Report Definition Tree that has attributes.
2. Select the attribute you want to delete in the XML Tag Attribute List and click **Delete**.

   Tip: Attributes prefaced with the letters "ab" in blue are attributes you can delete. Those prefaced with "ab" in grey cannot be deleted.

If you want to delete all attributes added using the XML Attribute dialog box for the selected element, click Delete All.

Suppressing XML tags

This option applies only to report definitions that use Custom Format.

1. Select a report object in the Report Definition Tree.
2. Select the **Suppress XML Tag** check box.

The selected report object won't be exported, and the associated XML element is not available for editing.

Note: You can suppress all children of the selected report object by selecting the Suppress All Children check box.

To reverse this action, clear the Suppress XML Tag check box and/or the Suppress All Children check box.

XML output specifications

This section describes the output you can expect when exporting a Crystal report in XML format. As stated in the "Exporting to XML format" section, Crystal Reports creates raw XML data you can manipulate as you need. This data corresponds to the information in your Crystal report and includes elements for sections, areas, field objects, text objects, subreports, cross-tabs, and so on.

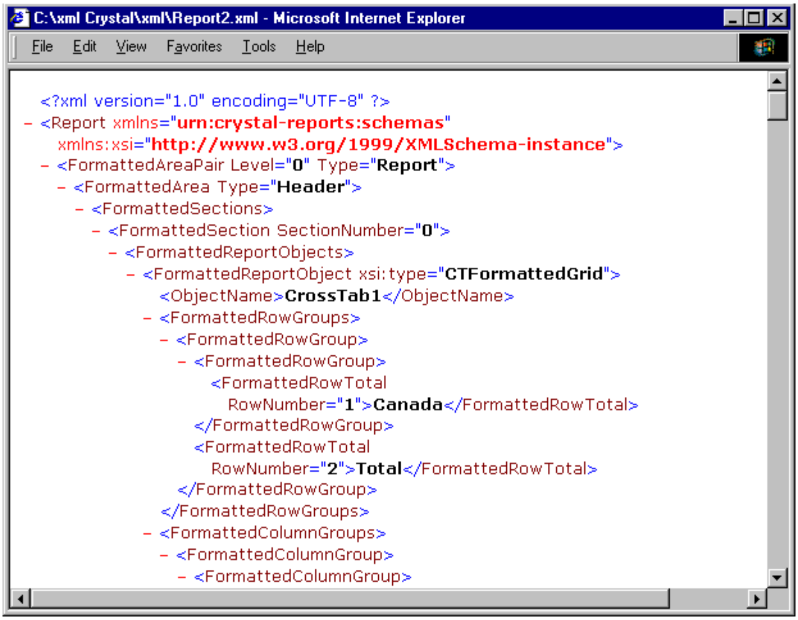
Note:

- Sections hidden or suppressed in a Crystal report are not exported as XML output even though they are displayed in the Report Definition Tree of the XML Format dialog box.
- Binary Large Object (BLOB) fields, such as charts, maps, and other types of graphics, are not exported.

The names and attributes of XML element tags depend on the output format you choose on the Others tab of the XML Format dialog box. Output for Crystal ML Schema and Custom Format is described below.

Crystal ML Schema

If you use a browser to look at an XML file exported using the Crystal ML Schema, you'll see something like this:


C:\xml Crystal\xml\Report2.xml - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Report xmlns="urn:crystal-reports:schemas"
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
  - <FormattedAreaPair Level="0" Type="Report">
    - <FormattedArea Type="Header">
      - <FormattedSections>
        - <FormattedSection SectionNumber="0">
          - <FormattedReportObjects>
            - <FormattedReportObject xsi:type="CTFormattedGrid">
                <ObjectName>CrossTab1</ObjectName>
              - <FormattedRowGroups>
                - <FormattedRowGroup>
                  - <FormattedRowGroup>
                      <FormattedRowTotal
                        RowNumber="1">Canada</FormattedRowTotal>
                    </FormattedRowGroup>
                    <FormattedRowTotal
                      RowNumber="2">Total</FormattedRowTotal>
                  </FormattedRowGroup>
                </FormattedRowGroups>
              - <FormattedColumnGroups>
                - <FormattedColumnGroup>
                  - <FormattedColumnGroup>
```

After the XML and document type declarations, there is a list of tags. These tags represent the contents of your report arranged sequentially, item by item. For example, the first few lines of the XML data shown above represent a report that begins with a header area containing a cross-tab object:

```
<Report xmlns="urn:crystal-reports:schemas" xmlns:xsi="http://www.w3.org/
1999/XMLSchema-instance">
  <FormattedAreaPair Level="0" Type="Report">
    <FormattedArea Type="Header">
      <FormattedSections>
        <FormattedSection SectionNumber="0">
          <FormattedReportObjects>
            <FormattedReportObject xsi:type="CTFormattedGrid">
              <ObjectName>CrossTab1</ObjectName>
```

- `<report>` represents the report itself
- `<FormattedAreaPair>` represents the generic report area pair (such as a header and footer)
- `<FormattedArea Type=”Header”>` represents the specific report area: the header
- `<FormattedSections>` represents a subdivision of the report header area
- `<FormattedSection SectionNumber=”0”>` represents an iteration of the subdivision
- `<FormattedReportObject>` represents a report object (in this case, a formatted grid, or cross-tab)
- `<ObjectName>` represents the report object’s (cross-tab’s) name.

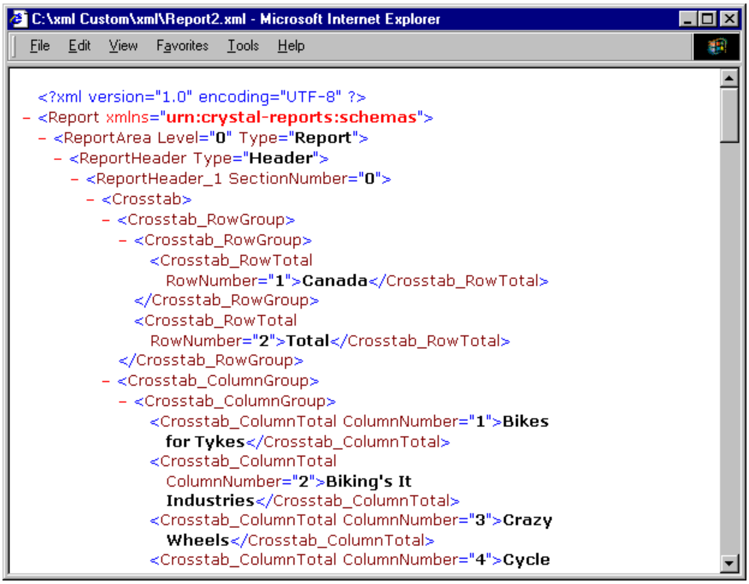
The entire XML file can be read in such a way. Each line in the XML data has a direct relation to your Crystal report and its Report Definition as defined in Crystal XML Schema and the XML Format dialog box.

Note: For more detailed information about the XML elements and tags in this example, see the Crystal ML Schema published at the following address:

`http://www.seagatesoftware.com/xml/schema.xsd`

Custom Format

If you use a browser to look at an XML file exported using the Custom Format, you'll see something like this:



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Report xmlns="urn:crystal-reports:schemas">
  - <ReportArea Level="0" Type="Report">
    - <ReportHeader Type="Header">
      - <ReportHeader_1 SectionNumber="0">
        - <Crosstab>
          - <Crosstab_RowGroup>
            - <Crosstab_RowGroup>
                <Crosstab_RowTotal
                  RowNumber="1">Canada</Crosstab_RowTotal>
              </Crosstab_RowGroup>
              <Crosstab_RowTotal
                RowNumber="2">Total</Crosstab_RowTotal>
            </Crosstab_RowGroup>
          - <Crosstab_ColumnGroup>
            - <Crosstab_ColumnGroup>
                <Crosstab_ColumnTotal ColumnNumber="1">Bikes
                  for Tykes</Crosstab_ColumnTotal>
                <Crosstab_ColumnTotal
                  ColumnNumber="2">Biking's It
                  Industries</Crosstab_ColumnTotal>
                <Crosstab_ColumnTotal ColumnNumber="3">Crazy
                  Wheels</Crosstab_ColumnTotal>
                <Crosstab_ColumnTotal ColumnNumber="4">Cycle
```

After the XML and document type declarations, there is a list of tags. These tags represent the contents of your report arranged sequentially, item by item. For example, the first few lines of the XML data shown above represent a report that begins with a header area containing a cross-tab object:

```
<Report xmlns="urn:crystal-reports:schemas">
  <ReportArea Level="0" Type="Report">
    <ReportHeader Type="Header">
      <ReportHeader_1 SectionNumber="0">
        <Crosstab>
```

- `<report>` represents the report itself
- `<ReportArea>` represents the generic report area
- `<ReportHeader Type=”Header”>` represents the specific report area: the header
- `<ReportHeader_1 SectionNumber=”0”>` represents a subdivision of the report header area
- `<Crosstab>` represents a cross-tab object.

The entire XML file can be read in such a way. Each line in the XML data has a direct relation to your Crystal report and its Report Definition as defined in the Customize XML dialog box.

Note: The element tag names in this example are the default names; they will be different if you changed them in the XML Format dialog box.

Creating Reports from Excel and Access

21

This chapter describes the Add-Ins for Microsoft Excel and Microsoft Access. It explains how you create reports from an Access table or query and an Excel spreadsheet with the Crystal Report Wizard.

Overview of the Crystal Reports Add-Ins

Crystal Reports takes advantage of Microsoft's Add-In technology by offering a report creation Wizard you can use with Microsoft Excel and Microsoft Access. The Crystal Report Wizard offers familiar report design capabilities directly within your Microsoft Office applications—you can create a Crystal report from your data without leaving Excel or Access.

The method for working with the Add-Ins is similar in both Excel and Access:

- open Microsoft Excel or Microsoft Access (the Add-Ins are automatically installed when you install Crystal Reports)
- select a spreadsheet (or selected cells), table, or query and launch the Wizard
- use the creation Wizard's familiar screen layout to create a report to your specifications.

Note: The Add-Ins work with only the Office 97 and Office 2000 versions of Excel and Access.

About the Microsoft Excel Add-In

Choose an entire data range in a spreadsheet, or select certain cells to create a Crystal report. The report you create is linked to your Excel spreadsheet and can be refreshed to show changes made to the spreadsheet data.

Note: Excel Add-In reports are updated when you refresh their data in Crystal Reports if you have:

- added a row to the selected spreadsheet range (but not pre-appended or appended a row)
- deleted a row from the selected spreadsheet range
- modified a cell within the selected spreadsheet range.

After you create a report using the Wizard, you can preview it using the default ActiveX viewer or you can launch Crystal Reports or Seagate Analysis to modify it.

About the Microsoft Access Add-In

Choose a table or query to create a Crystal report. The report you create is linked to your Access table or query and can be refreshed to show changes made to the database.

Note:

- Access Add-In reports will not reflect database changes until you close the modified table or query in Access and refresh the report data in Crystal Reports
- Access Add-In reports must be based on one table or query (but not a query with input parameters).

After you create a report using the Wizard, you can preview it using the default ActiveX viewer or you can launch Crystal Reports or Seagate Analysis to modify it.

Working with the Microsoft Excel Add-In

When you install Crystal Reports 8.5, the Excel Add-In is added automatically. A Crystal Report Wizard menu option is added to the Tools menu in Excel and a button is added to the Standard toolbar.

You can use the Crystal Report Wizard to create a report anytime you use Excel. The option remains available until you uncheck Crystal Report Wizard in the Add-Ins dialog box (under the Tools menu in Excel).

Note: The Excel Add-In file is placed by default in the `\\Program Files\Seagate Software\Report Designer Component` directory.

Using the Crystal Report Wizard to create a report in Excel

The Crystal Report Wizard in Excel offers familiar Crystal layout and selection screens to help you create a report from your spreadsheet. You can select a range of cells, or an entire data range in a spreadsheet to use as your report content.

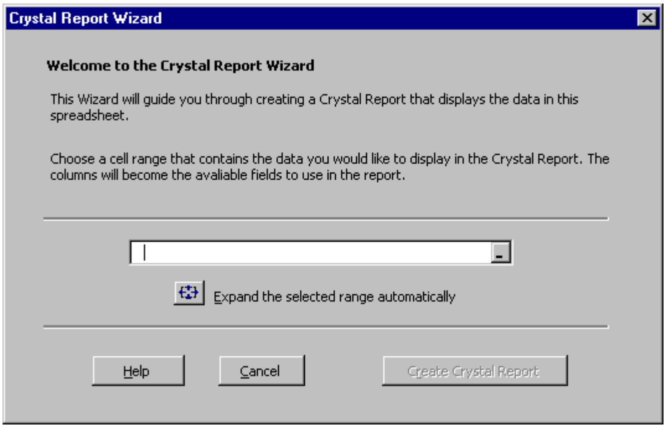
Note: When you create a report from an Excel spreadsheet, Crystal Reports creates a database file (.mdb). This file, the Excel spreadsheet file (.xls), and the report file itself (.rpt) cannot be moved or you'll lose the links between them and won't be able to refresh your report data later.

To create a report in Excel using the Crystal Report Wizard

1 On the **Tools** menu in Excel, choose **Crystal Report Wizard**.

Tip: Another way to open the Crystal Report Wizard is to click the Crystal Report Wizard button.

If your spreadsheet has unsaved data, Excel warns you to save before running the Wizard.


Crystal Report Wizard

Welcome to the Crystal Report Wizard

This Wizard will guide you through creating a Crystal Report that displays the data in this spreadsheet.

Choose a cell range that contains the data you would like to display in the Crystal Report. The columns will become the avaliable fields to use in the report.

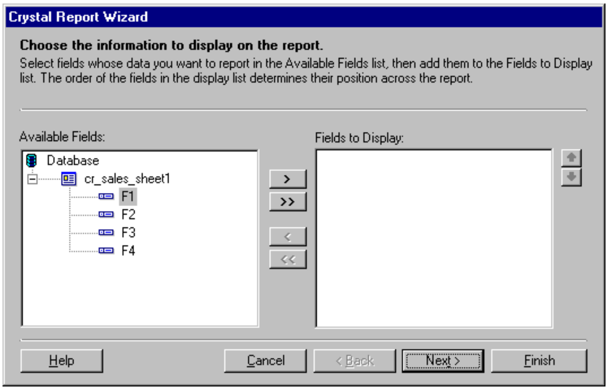
2 Type a data range in the cell range box, or click its button to choose a range directly from the spreadsheet.

Clicking the button minimizes the Wizard so you can click-drag to select a data range. To return to the Wizard, click the button on the minimized Wizard.

Note: To select an entire data range in the spreadsheet, click “Expand the selected range automatically.”

3 With a data range selected, click **Create Crystal Report**.

The Crystal Report Wizard appears.



Crystal Report Wizard

Choose the information to display on the report.

Select fields whose data you want to report in the Available Fields list, then add them to the Fields to Display list. The order of the fields in the display list determines their position across the report.

Note: As you design your report, the program creates a database file (.mdb). The database file, the spreadsheet file (.xls), and the resulting report file (.rpt) cannot be moved or you won’t be able to refresh your report data later.

4 Add the database fields you want to appear in your report to the **Fields to Display** list.

The arrow buttons on this dialog box enable you to move fields from one list to the other. Single arrows move only the selected field; double arrows move all fields at the same time.

5 Click **Next** to choose fields to group on.

6 Add the database fields you want to group on to the **Group By** box.

When a group field is selected, you can choose a sorting order from the Sort Order box.

7 Click **Next** to choose fields to summarize on.

8 Add the database fields you want to total on to the **Summarized Fields** list.

Summarized fields apply to the group specified in the For the Group box. You can use the same fields in each of several different groups.

When a summary field is selected, you can choose a summary type for it and add a grand total.

9 Click **Next** to sort groups on their summarized totals.

You can select Sort All Groups, or you can specify a Top or Bottom N.

10 Click **Next** to create a record selection for your report.

11 Add the database fields you'll use to filter your report to the **Filtered Fields** list.

Define your record selection by selecting from the box of limiting operators and entering the appropriate field value.

12 Click **Next** to select a report style.

13 Click **Next** to enter a title and saving location for your report.

14 Choose a method for viewing your report.

There are four options:

- Preview (read only)
- Edit with the Crystal Report Designer
- Analyze with Crystal Analysis
- View Later.

The Preview option opens the report in the ActiveX viewer. You can view, print, and refresh the report in the viewer.

The Edit and Analyze options open the report in the appropriate program. You can view, print, refresh, and modify the report while using all the capabilities of Crystal Reports or Seagate Analysis.

The View Later option saves the report with the name you gave it to the directory you specified.

15 Click **Finish**.

Now that your report has been created, you can use it and modify it as you would any other report in Crystal Reports. It remains linked to your Excel spreadsheet so if you refresh the report after altering the spreadsheet, your changed data is displayed (see "About the Microsoft Excel Add-In" on page 374).

Working with the Microsoft Access Add-In

When you install Crystal Reports 8.5, the Access Add-In is added automatically. A menu option called Crystal Report Wizard is added to the Add-Ins submenu of the Tools menu in Access.

Note: Installation of the Access Add-In is not automatic for Office 97. For instructions on how to install the Crystal Report Wizard in Access 97, see the Crystal Reports Online Help.

You can use the Crystal Report Wizard to create a report anytime you use Access. The option remains available until you uninstall Crystal Report Wizard in the Add-In Manager dialog box (under the Tools menu in Access).

Note: The Access Add-In files are placed by default in the `\\Program Files\Seagate Software\Report Designer Component` directory.

Using the Crystal Report Wizard to create a report in Access

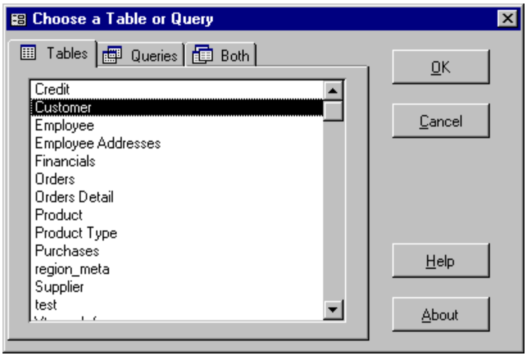
The Crystal Report Wizard offers familiar Crystal layout and selection screens to help you create a report from your table or query.

To create a report in Access using the Crystal Report Wizard

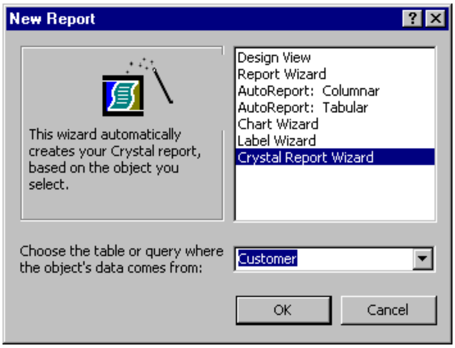
1 On the **Tools** menu in Access, point to **Add-Ins** and then click **Crystal Report Wizard**.

If your database table or query is open, Access warns you to save before running the wizard.

The Choose a Table or Query dialog box appears if you haven't already highlighted a table or query in the Database dialog box or the New Report dialog box.



Another way to open the Crystal Report Wizard is to select the Reports option in the Database dialog box, click New, select the Crystal Report Wizard, and then click OK. If you use this method, you can select your table or query on the New Report dialog box, and go to step 3.

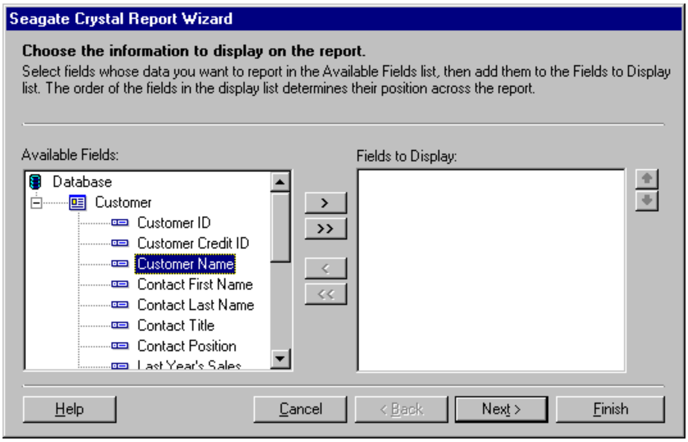


This wizard automatically creates your Crystal report, based on the object you select.

Choose the table or query where the object's data comes from: Customer

2 From the Choose a Table or Query dialog box, select the table or query you want to use and click **OK**.

The Crystal Report Wizard appears.



Seagate Crystal Report Wizard

Choose the information to display on the report.

Select fields whose data you want to report in the Available Fields list, then add them to the Fields to Display list. The order of the fields in the display list determines their position across the report.

3 Add the database fields you want to appear in your report to the **Fields to Display** list.

The arrow buttons on this dialog box enable you to move fields from one list to the other. Single arrows move only the selected field; double arrows move all fields at the same time.

4 Click **Next** to choose fields to group on.

5 Add the database fields you want to group on to the **Group By** box.

When a group field is selected, you can choose a sorting order from the Sort Order box.

6 Click **Next** to choose fields to summarize on.

7 Add the database fields you want to total on to the **Summarized Fields** list.

Summarized fields apply to the group specified in the For the Group box. You can use the same fields in each of several different groups.

When a summary field is selected, you can choose a summary type for it and add a grand total.

8 Click **Next** to sort groups on their summarized totals.

You can select Sort All Groups, or you can specify a Top or Bottom N.

9 Click **Next** to create a record selection for your report.

10 Add the database fields you'll use to filter your report to the **Filtered Fields** list.

Define your record selection by selecting from the box of limiting operators and entering the appropriate field value.

11 Click **Next** to select a report style.

12 Click **Next** to enter a title and saving location for your report.

13 Choose a method for viewing your report.

There are four options:

- Preview (read only)
- Edit with the Report Designer
- Analyze with Seagate Analysis
- View Later.

The Preview option opens the report in the ActiveX viewer. You can view, print, and refresh the report in the viewer.

The Edit and Analyze options open the report in the appropriate program. You can view, print, refresh, and modify the report while using all the capabilities of Crystal Reports or Seagate Analysis.

The View Later option saves the report with the name you gave it to the directory you specified.

Now that your report has been created, you can use it and modify it as you would any other report in Crystal Reports.

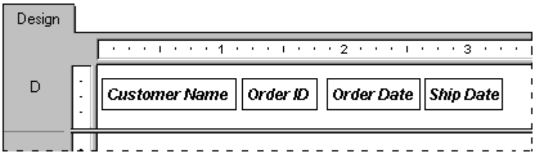
Using Formulas 22

This chapter explains the basics of formulas and introduces you to the Formula Editor in order for you to begin to create formulas.

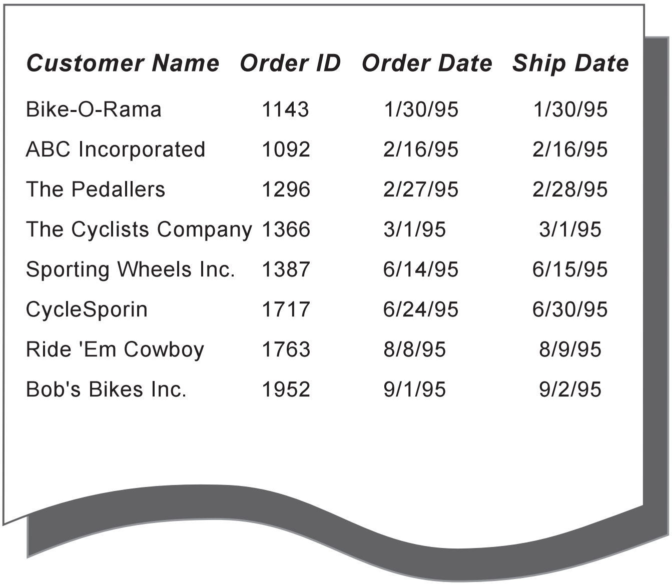
Formulas overview

In many cases, the data needed for a report already exists in database table fields. For example, to prepare an order list you would place the appropriate fields on the report.

By placing these fields...



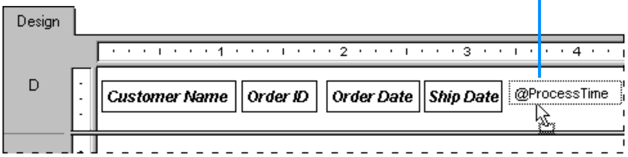
...you get this kind of report.



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

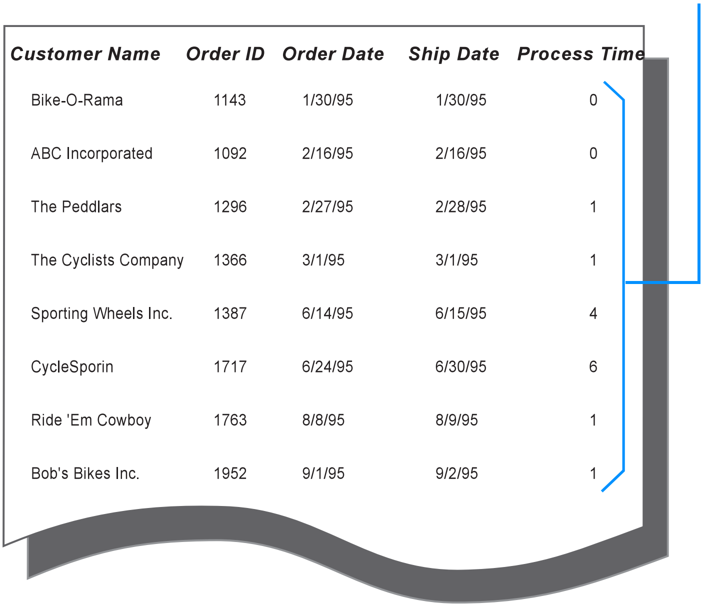
Sometimes, however, you need to put data on the report that does not exist in any of the data fields. In such cases, you need to create a formula. For example, to calculate the number of days it takes to process each order, you need a formula that determines the number of days between the order date and the ship date. Crystal Reports makes it easy for you to create such a formula.

Place the formula where you want it to appear in the report.



You get this kind of report:

The formula subtracts the Order Date from the Ship Date and then prints the result here.



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

Typical uses for formulas

There are many uses for formulas. If you have a need for specialized data manipulation, you can do it with a formula.

Creating calculated fields to add to your report

To calculate a price discounted 15%:

Crystal syntax example:

```
{Orders_Detail.Unit Price}*.85
```

Basic syntax example:

```
formula = {Orders_Detail.Unit Price}*.85
```

Formatting text on a report

To change all the values in the Customer Name field to uppercase:

Crystal syntax example:

```
UpperCase ({Customer.Customer Name})
```

Basic syntax example:

```
formula = UCase ({Customer.Customer Name})
```

Pulling out a portion, or portions, of a text string

To extract the first letter of the customer name:

Crystal syntax example:

```
{Customer.Customer Name} [1]
```

Basic syntax example:

```
formula = {Customer.Customer Name} (1)
```

Extracting parts of a date

To determine what month an order was placed:

Crystal syntax example:

```
Month ({Orders.Order Date})
```

Basic syntax example:

```
formula = Month ({Orders.Order Date})
```

Formula components and syntax

Formulas contain two critical parts: the components and the syntax. The components are the pieces that you add to create a formula while the syntax is the rules that you follow to organize the components.

Formula components

Creating a formula in Crystal Reports is like creating one in any spreadsheet application. You can use any of the following components in your formula:

Fields

Example: {customer.CUSTOMER LAST NAME}, {customer.LAST YEAR'S SALES}

Numbers

Example: 1, 2, 3.1416

Text

Example: “Quantity”, “:”, “your text”

Operators

Example: + (add), / (divide), -x (negate)

Operators are actions you can use in your formulas.

Functions

Example: Round (x), Trim (x)

Functions perform calculations such as average, sum, and count. All functions available are listed with their arguments and are arranged by their use.

Control Structures

Example: “If” and “Select”, “For” loops

Group field values

Example: Average (fld, condFld), Sum (fld, condFld, “condition”)

Group field values summarize a group. For example, you could use group field values to find the percentage of the grand total contributed by each group.

Other formulas

Example: {@GrossProfit}, {@QUOTA}

Formula syntax

Syntax rules are used to create correct formula. Some basic rules are:

- enclose text strings in quotation marks
- enclose arguments in parentheses (where applicable)
- referenced formulas are identified with a leading @ sign.

Crystal and Basic Syntax

When creating formulas, you have the option of using either Crystal or Basic syntax. Almost any formula written with one syntax can be written with the other. Reports can contain formulas that use Basic syntax as well as formulas that use Crystal syntax.

Crystal syntax is the formula language included in all prior versions of Crystal Reports. Basic Syntax is new to Crystal Reports 8.

If you are familiar with Microsoft Visual Basic or other versions of Basic, then Basic syntax may be more familiar to you. In general, Basic syntax is modeled on Visual Basic except that it has specific extensions to handle reporting.

If you are already comfortable with Crystal syntax, you can continue to use it, and benefit from the new functions, operators and control structures inspired by Visual Basic.

Note: Report processing is not slowed down by using Basic syntax. Reports using Basic syntax formulas can run on any machine that Crystal Reports runs on. Also, using Basic syntax formulas does not require distributing any additional files with your reports.

Related topics

To learn about Basic syntax, see "Creating Formulas with Basic Syntax" on page 399.

To learn about Crystal syntax, see "Creating Formulas with Crystal Syntax" on page 437.

Specifying formulas

There are four different groups of formulas in Crystal Reports: report, formatting, selection, and search formulas. The majority of formulas in a report are report formulas and conditional formatting formulas.

Report formulas

Report formulas are formulas that you create to stand alone in a report. For example, a formula that calculates the days between the order date and the shipping date is a report formula.

Conditional formatting formulas

Formatting formulas change the layout and design of a report, as well as the appearance of text, database fields, objects, or entire report sections. You format text through the Format Editor. If you need to create a formatting formula, you access the Formula Editor from the Format Editor. See “Working with conditional formatting” on page 237.

Selection formulas

Selection formulas specify and limit the records and groups that appear in a report. You normally do not enter these formulas directly, but instead specify the selection using the Select Expert. Crystal Reports then generates the record selection and group selection formula. You have the option to manually edit these formulas, but you must use Crystal syntax. See “Selecting records” on page 126.

Search formulas

Search formulas help you locate data in your report. Like selection formulas, you normally do not enter these formulas directly, but instead specify the search criteria using the Search Expert. Crystal Reports generates the formula. You have the option to manually edit these formulas, but you must use Crystal syntax.

Note: If you already know Basic syntax, you need to know only a small amount of Crystal syntax to modify most selection and search formulas.

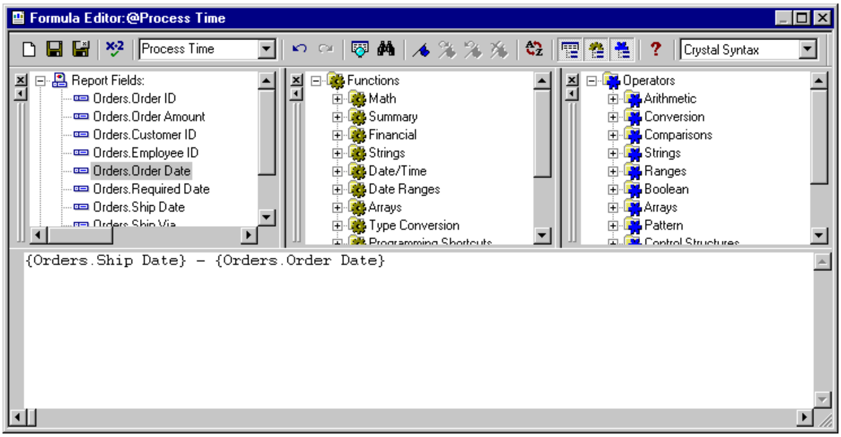
Working with the Formula Editor

In cases other than selection and search formulas—which you can define using the Select Expert and Search Expert—you create all formulas in the Formula Editor of Crystal Reports.

Accessing the Formula Editor

1. On the **Insert** menu, click **Field Object**.
2. In the Field Explorer dialog box, highlight **Formula Fields**.
3. Click **New**.
4. In the Formula Name dialog box, enter the name of the formula you will create.
5. Click **OK**.

   The Formula Editor appears.



```
{Orders.Ship Date} - {Orders.Order Date}
```

Understanding the sections of the Formula Editor

The Formula Editor contains four main windows.

Window	Description of contents
Report Fields	Report fields contain all database fields accessible for your report. They also contain any formulas or groups already created for the report.
Functions	Functions are prebuilt procedures that return values. They perform calculations such as average, sum, count, sin, trim, and uppercase.
Operators	Operators are the “action verbs” you use in formulas. They describe an operation or an action to take place between two or more values. Examples of operators: add, subtract, less than, and greater than.
Formula text window	Area where you create a formula.

Choosing the syntax

The top right corner of the Formula Editor contains the drop-down list where you choose either Crystal or Basic syntax for the formula you are creating.

Note: Changing the syntax from Crystal syntax to Basic syntax or vice versa will change the list of functions in the Functions window as well as the list of operators in the Operators window. The functions and operators differ from syntax to syntax.

The available report fields remain the same since the report fields are available to each syntax.

Setting the default syntax

When you open the Formula Editor, Crystal syntax appears as the syntax default. If you want to change the syntax default, select Options from the File menu, then click the Reporting tab. Choose the preferred syntax from the Formula Language drop-down list and click OK. When you access the Formula Editor, the syntax you selected appears as the default.

Entering formula components

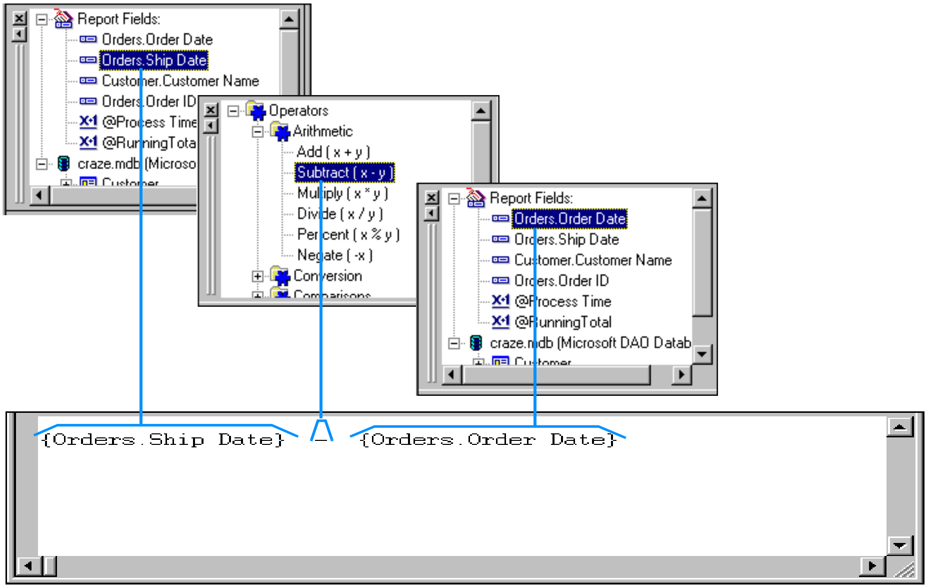
The Report Fields, Functions, and Operators tree at the top of the Formula Editor contain the primary formula components. Double-click any component from these trees to add this component to your formula.

For example, if you set the syntax to Basic Syntax and double-click the Operators > Control Structures > Multi-Line If in the Operators tree, the following text is transferred to the Formula text window with the cursor between the If and Then:

```
If | Then
ElseIf Then
Else
End If
```

The above text helps you organize the parts needed to write your formula.

The following example in Crystal syntax shows how double-clicking on the report fields, functions, and operators insert the required syntax for a formula.



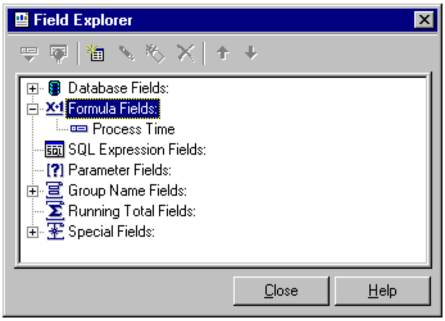
```
{Orders.Ship Date} - {Orders.Order Date}
```

Creating and modifying formulas

Creating a formula and inserting it into a report

1 On the **Insert** menu, click **Field Object**.

The Field Explorer dialog box appears.



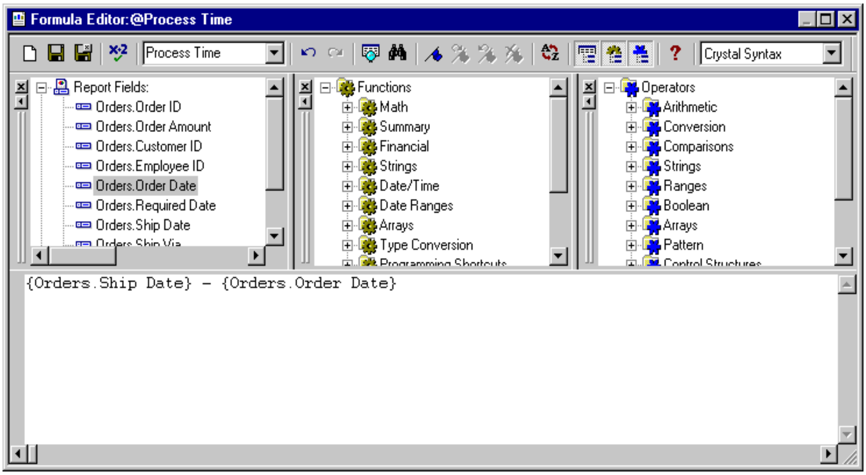
2 Select **Formula Fields**.

3 Click **New**.

4 In the Formula Name dialog box, enter the name you want to identify the formula.

5 Click **OK**.

The Formula Editor appears.



```
{Orders.Ship Date} - {Orders.Order Date}
```

6 Choose either Crystal or Basic syntax.

If you are unsure which syntax to choose see “Formula syntax” on page 386.

7. Enter the formula by typing in the components or selecting them from the component trees.
8. Click **Check** to identify any errors in the formula.
9. Fix any syntax errors the Formula Checker identifies.
10. When the formula has the correct syntax, click **Save and Close**.
    You return to the Field Explorer dialog box.
11. Double-click the formula and position the cursor where you want the formula to appear on your report.
12. Click once to set the field in the desired position.

Note: A formula that is placed on a report is indicated by @ (for example, @ProcessTime) on the Design tab.

Related topics

"Creating Formulas with Basic Syntax" on page 399.

"Creating Formulas with Crystal Syntax" on page 437.

Editing formulas

1. Choose **Insert** from the main menu and select **Field Object**
   The Field Explorer dialog box appears.
2. Right-click the formula you want to edit and choose **Edit**.
   The Formula Editor appears.
3. Edit the formula.
4. Click **Check** to identify any errors in the formula.
5. Fix any syntax errors the Formula Checker identifies.
6. When the formula has the correct syntax, click **Save and Close**.
   You return to the Field Explorer dialog box.

Searching and replacing text

1. On the **Insert** menu, click **Field Object**.
   The Field Explorer dialog box appears.
2. Right-click the formula you want to edit and choose **Edit**.
   The Formula Editor appears.
3. In the Formula Editor dialog box, click **Find/Replace** to open a Find dialog box.
   From this dialog box, you can search and replace text within the Formula text box.
4. Click the **Mark All** button to mark all occurrences of the search text.

5 Click the **Replace All** button to replace all occurrences of the search text with the contents of the Replace with text box.

You can also search (but not replace) within any of the Formula Editor trees (use the Search options to specify which list boxes you are searching). The Mark All, Replace, and Replace All buttons become inactive when you specify a search within a list box.

Copying formulas from online help

The Crystal Reports online help contains a section called "Formulas in Action" that covers 17 different formulas and report-related scenarios. Since the formulas you develop using the Formula Editor are text, you can copy useful online formulas directly into the Formula Editor and then modify them to fit your needs.

To copy formulas from online help

1. On the **Help** menu, click **Crystal Reports Help**.

   The Crystal Reports online help appears.
2. Click the **Index** tab.
3. Enter "Formulas in Action" in the keyword field and click **Display**.

   The Formulas in Action Help topics appear.
4. Scroll through the topic until you locate the formula you want to copy.
5. Highlight the formula, right-click, and choose **Copy** from the menu.

   Windows places a copy of the selected text on the Clipboard.
6. Return to Crystal Reports, choose **Insert** from the main menu and select **Field Object**.

   The Field Explorer dialog box appears.
7. Select **Formula Fields** and click the **New** button.

   The Formula Name dialog box appears.
8. Enter the name you want to identify the formula and click **OK**.

   The Formula Editor appears.
9. Place the insertion point where you want the text to appear in the **Formula** text box of the Formula Editor and press Ctrl-V to paste the text from the Clipboard.
10. Modify the formula by changing the fields, formulas, group fields, conditional statements, and text strings as necessary for use with the data in the new report.
11. When the formula has the correct syntax, click **Save and Close**.

    You return to the Field Explorer dialog box.

Copying formulas from one report to another

You may want to copy a formula created in one report for use in another report. Copy the text formula from one report to another via the Clipboard.

To copy a formula from one report to another

1. Select the formula field you want to copy in the report.
2. On the **Edit** menu, click **Copy**.
3. Open the report you want to copy the formula to.
4. Choose **Paste** from the Edit menu.
5. When the program displays the object frame, drag the formula to the new location.
6. To make changes to the formula, right-click the formula and choose **Edit Field Object** from the shortcut menu.

   The Formula Editor appears.
7. Delete the old values and type in the new values, or select them from the Fields, Functions, and/or Operators tree.
8. Click **Save and Close** when finished.

Key points for editing a copy of a formula

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 the 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 sure that the text strings used in the formula match values that actually exist in the new data.

Deleting formulas

When a formula is created and added to a report, the Report Designer:

- Stores the specification for creating the formula, using the name you assigned to it.
- 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 completely 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.

Removing the working formula from your report

1 Right-click the formula you want to delete from the report.

2 Select **Delete**.

Note: Even after the working copies of a formula have been deleted from the report, the formula specification remains unchanged. The specification is listed in the Field Explorer dialog box. It is available if you wish to enter the formula in the report again.

Deleting the formula specification

1 Choose **Insert** from the main menu and select **Field Object**.
The Field Explorer dialog box appears.

2 Right-click the formula you want to delete and choose **Delete**.

Note: A dialog box appears if this formula is currently in use in a report. If you delete this formula, you will delete all references of it in reports. Click Yes to delete.

Debugging formulas

Debugging tutorial

Follow the example below to learn the necessary steps for debugging a formula. After completing this exercise, use the same principles to debug your own formulas.

About this tutorial

- this tutorial uses the Xtreme.mdb sample database
- this tutorial uses Crystal syntax.

- The following formula is the formula you will test for errors:

```
If ({customer.CUSTOMER NAME} [1 to 2 = “Bi” and
ToText({customer,CUSTOMER ID}) [1] = “6”) Or
({customer.CUSTOMER NAME} [1] = 'Ro” and
ToText({customer.CUSTOMER ID}) [1] = “5”)
     “PREFERRED CUSTOMER”
Else
     “DOES NOT FIT CRITERIA”
```

If correct, this formula should pick out all customers whose names begin with “Bi” and whose customer IDs begin with “6” as well as those companies whose names begin with “Ro” and whose customer IDs begin with “5”. When printing the field, those selections will read “PREFERRED CUSTOMER”, while the rest will read “DOES NOT FIT CRITERIA”.

You will now break down the formula to check and see that each condition of the formula is working individually.

Formula1

1 To get started, create a report using the Customer table in **Xtreme.mdb** and place the following fields from left to right in the Details section:

```
{customer.CUSTOMER ID}
{customer.CUSTOMER NAME}
```

To test each portion of the formula, you will place a new formula field next to these two fields in the report.

2 Create a new formula called Formula1.

3 Type the following in the **Formula text** box of the Formula Editor:

```
If {customer.CUSTOMER NAME} [1 to 2 = “Bi” Then
     “TRUE”
Else
     “FALSE”
```

4 Click **Check** to test for errors. You will receive the following error message:

```
The ] is missing.
```

5 Correct the formula by inserting the missing “] “ after the 2.

6 Click **Check** again. You will receive the following message:

```
No errors found.
```

7 Click **Save and Close** to return to the Insert Fields dialog box.

8 Insert the corrected formula field to the right of the two data fields in the Details section of your report.

9 Click **Print Preview** on the Standard toolbar to check the values in the report and compare the fields to see if the field values returned by @Formula1 are correct.

You will find “TRUE” listed next to the customer names that begin with “Bi” and “FALSE” next to all the others. Now you will check the other portions of the formula. Create Formula2, Formula3, and Formula4, by following Steps 1 - 9, using the formulas specified below for each.

Insert each formula field on the same line of the Details section for easy comparison. Check each one for errors, fix as needed, and make sure the values returned are correct before moving on to Formula2.

Formula2

1 Create a new formula called Formula2.

2 Type the following in the *Formula text* box of the Formula Editor:

```
If ToText({customer,CUSTOMER ID}) [1] = “6” Then
      “TRUE”
Else
      “FALSE”
```

3 Click **Check** to test for errors. You will receive the following error message:

```
This field name is not known.
```

4 Correct the formula by replacing the comma (,) in the field name with a period (.).

5 Click **Check** again. The formula should now be error-free.

6 Place the formula to the right of the @Formula1 field.

7 Click **Print Preview** on the Standard toolbar to check the values in the report and compare the fields to see if the field values returned by @Formula2 are correct.

You should see “TRUE” next to all customer numbers that begin with 6 and “FALSE” next to all customer numbers that do not begin with 6.

Formula3

1 Create a new formula called Formula3.

2 Type the following in the **Formula text** box of the Formula Editor:

```
If {customer.CUSTOMER NAME} [1 to 2] = 'Ro” Then
      “TRUE”
Else
      “FALSE”
```

3 Click **Check** to test for errors. You will receive the following error message:

```
The matching ' for this string is missing.
```

4 Correct the formula by changing the single quote (‘) before Ro to a double quote (“).

5 Click **Check** again. The formula should now be error-free.

6 Place the formula to the right of the @Formula2 field.

7 Click **Print Preview** on the Standard toolbar to see the values in the report and compare the fields to check if the field values returned by @Formula3 are correct.

You should see “TRUE” next to all Customer names that begin with “Ro” and “FALSE” next to all Customer names that do not begin with “Ro”.

Formula4

1 Create a new formula called Formula4.

2 Type the following in the **Formula text** box of the Formula Editor:

```
If ToText({customer.CUSTOMER ID}) [1] = “5”
      “TRUE”
Else
      “FALSE”
```

3 Click **Check** to test for errors. You will receive the following error message:

```
The word 'then' is missing.
```

4 Correct the formula by typing in the word “Then” at the end of the first line after “5”.

5 Click **Check** again. The formula should now be error-free.

6 Place the formula to the right of the @Formula3 field.

7 Click **Print Preview** on the Standard toolbar to see the values in the report and compare the fields to check if the field values returned by @Formula4 are correct.

You should see “TRUE” next to all Customer IDs that begin with 5 and “FALSE” next to all Customer IDs that do not begin with 5.

Now that the formulas are error-free and the field values returned are correct, you will create a formula that links the separate components together. You will begin by linking the first two formulas (@Formula1 and @Formula2) and then you will add @Formula3 and @Formula4 to create the final formula @FinalFormula.

Formula1+2

1 Create a new formula called Formula1+2.

2 Type the following in the *Formula text* box of the Formula Editor:

```
If {customer.CUSTOMER NAME} [1 to 2] = “Bi” and
ToText({customer.CUSTOMER ID}) [1] = “6” Then
      “TRUE”
Else
      “FALSE”
```

3 Place the formula to the right of the @Formula4 field.

You should see “TRUE” next to each customer whose name begins with Bi and Id begins with 6, and “FALSE” next to all Customer IDs that do not meet this criteria.

If this formula is working correctly, you can create one last formula adding the code from @Formula3 and @Formula4.

FinalFormula

1 Create a new formula called FinalFormula.

2 Type the following in the Formula text box of the Formula Editor:

```
If ({customer.CUSTOMER NAME} [1 to 2] = “Bi” and
ToText({customer.CUSTOMER ID}) [1] = “6”) or
({customer.CUSTOMER NAME} [1 to 2] = “Ro” and
ToText({customer.CUSTOMER ID}) [1] = “5”) Then
     “PREFERRED CUSTOMER”
Else
     “DOESN'T FIT CRITERIA”;
```

3 Place the formula where you want it to appear in the Details section of the report. You can now delete all other formula fields from the report. See “Deleting formulas” on page 394.

You can use this same process of condition-by-condition testing for any formulas as a means of systematically checking them.

Creating Formulas with Basic Syntax 23

This chapter provides you with an overview of various parts of a formula. You learn about Basic syntax and the techniques you can use when creating a formula.

Basic syntax overview

When creating formulas, you have the option of using either Crystal or Basic syntax. Syntax rules are used to create correct formulas. Almost any formula written with one syntax can be written with the other. Reports can contain formulas that use Basic syntax as well as formulas that use Crystal syntax.

If you are familiar with Microsoft Visual Basic or other versions of Basic, then Basic syntax may be more familiar to you. In general, Basic syntax is similar to Visual Basic except that it has specific extensions to handle reporting.

Note: Report processing is not slowed down by using Basic syntax. Reports using Basic syntax formulas can run on any machine that Crystal Reports runs on. Also, using Basic syntax formulas does not require distributing any additional files with your reports.

Tips on learning Basic syntax

If you have no programming experience

In many cases you may not need to use the formula language. Crystal Reports includes several Experts that automatically handle situations where formulas could be used. These include the Select, Search, Running Totals and Highlighting Experts, and the Insert Summary and Insert Grand Total dialog boxes. Before creating formulas, check to see if you can use one of these tools.

However, you may need to create formulas without the help of an Expert. Read this chapter to learn about Basic syntax and the rules you need to follow to create a formula.

If you know Microsoft Visual Basic, VBScript, or another version of Basic

In order to create formulas using Basic syntax, you need to understand the following:

- How a Basic syntax formula refers to other fields in the report such as database fields, parameter fields, summary fields, running total fields and other formula fields.
- How to return a value from a formula by setting the special variable named *formula*.
- How to use functions specific to report processing such as ReportTitle and OnFirstRecord.
- How to use the type system. Basic syntax is strongly typed, similar to using Option Explicit in Visual Basic and there is no Variant type. Basic syntax supports separate Date, Time and DateTime types unlike just the Date type in Visual Basic. It also supports range types such as *10 To 20* to allow for ranges of values which are common in many reporting applications.

Read this chapter and access the accompanying online help to familiarize yourself with the above details.

Familiar features

- Many Basic syntax functions work in the same way as their counterparts in Visual Basic. This includes string functions such as Len, Mid and Filter, math functions such as Abs, Rnd and Sin, financial functions such as PV, programming shortcut functions such as IIF and date functions such as DateSerial, DateAdd and DateDiff.
- Most operators supported by Visual Basic are also in Basic syntax. For example, string concatenation (&) and date-time literals (#...#).
- Most statements and control structures use the same syntax as in Visual Basic. This includes the If, Select, Do While, Do Until, While and For/Next statements.
- The overall look of the formula will be unmistakably Basic like. For example, Basic style comments and line continuation characters are supported as is the Basic language use of new lines, colons, and the equal sign.

If you already know Crystal syntax

The main adjustment is getting used to the parts of the Basic language that are common to every version of Basic. Read this chapter and the online help for a detailed introduction.

Using the online help

The Crystal Reports online help lists all the functions, operators, and control structures available with Basic syntax and provides a detailed explanation of each of them.

Basic syntax fundamentals

The result of a formula

The result of a formula, or the value that is printed when the formula is placed in a report, is called the *value returned by the formula*. Every formula in Crystal Reports must return a value. Basic syntax does this by setting the value of the special variable *formula*. For example, here is a simple Basic syntax formula that returns the value 10:

```
formula = 10
```

The value returned by a formula can be one of the seven simple data types supported: Number, Currency, String, Boolean, Date, Time and DateTime. Crystal Reports also supports range types and array types, but these cannot be returned by a formula.

Restrictions on changing formula variable data types

The *formula* variable can be set several times within a single formula. For example, suppose a company has a shipping policy in which orders over $1,000 are insured, but orders below that amount are not insured:

```
Rem A formula that returns a String value
If {Orders.Order Amount} >= 1000 Then
   formula = "Insured shipping"
Else
   formula = "Regular shipping"
End If
```

Tip: The text following the keyword Rem is a comment for someone reading this formula and is ignored by the Basic syntax compiler.

The above formula returns the text string value "Insured shipping" if the value of the database field {Orders.Order Amount} is greater than or equal to 1000; otherwise, it returns the text string value "Regular Shipping." Text strings are usually just referred to as strings. Notice that the *formula* variable appears twice in the above example.

If the *formula* variable is set to a value of one type, it cannot be set to a value of another type later in the same formula. For example, replacing the String "Regular shipping" in the above example with the Number 10 would result in an error since the special variable *formula* was first set to the String value "Insured shipping."

The reason for this restriction is that Crystal Reports needs to know in advance what the return type of a formula will be so that it can allocate enough storage for the returned values. This is because different types have different storage requirements. Another reason is that the formatting options available for a formula field depend on its type. For example, a Number field has Number formatting options, such as the number of decimals to display, which do not make sense for a String field.

Note: The special variable *formula* should not be declared, unlike other variables used in a Basic syntax formula. See "Variables" on page 410.

How formulas relate to functions in Microsoft Visual Basic

Consider the following Basic syntax formula:

```
Rem A formula that returns a String value
Rem The function Rnd returns a random number
Rem between 0 and 1
If Rnd > 0.9 Then
   formula = "You won!"
Else
   formula = "Sorry, try again."
End If
```

The above formula returns the text string value "You won!" if the random number returned by Rnd is greater than 0.9 and the text string value "Sorry, try again." otherwise.

The use of the *formula* variable is similar to writing a function named *formula* in Visual Basic.

For example, the above formula could be written as a Visual Basic function as follows:

```
Rem The following code is in Visual Basic
Function formula()
    If Rnd > 0.9 Then
       formula = "You won!"
    Else
       formula = "Sorry, try again."
    End If
End Function
```

The variable formula must be assigned a value

If the variable *formula* is not assigned a value, it is not a complete Basic syntax formula.

Note: Some examples in this chapter are not complete Basic syntax formulas but rather just fragments intended to explain a particular feature.

What if I'm not interested in using the value returned by a formula?

Sometimes you may want to write a formula that just declares and initializes some global variables. These formulas are commonly inserted into the report header section of a report. In such cases, assign any value to the special variable *formula*. Every formula must return a value, even if you are not interested in using that value.

For example:

```
Rem Some Global variable declarations
Rem Remember to set the value of 'formula'
Global x As String, y As Number, z As DateTime
x = "hello"
y = 10.5
z = #Aug 6, 1976#
formula = 10
```

Basic syntax is not case-sensitive

What this means is that *formula*, *Formula*, and *FORMULA* are all considered to be the same. This is true of all variable names, functions, and keywords used in a Basic syntax formula.

Note: The only exception to this rule is for strings. The string “Hello” is not the same as the string “hello”.

Practice using the Xtreme sample database

Many of the examples in this chapter refer to the Xtreme sample database. This database is included with Crystal Reports.

Browse the Samples\En directory to find the Databases folder that contains the Xtreme database.

Comments

Formula comments are notes included with a formula to explain its design and operation. Comments do not print and they do not affect the formula; they appear only in the Formula Editor. Use comments to explain the purpose of a formula or explain the steps involved in writing it.

Comments work as in Visual Basic. Begin comments with a Rem or an apostrophe.

Note: A comment beginning with a Rem is a separate statement and must either start on a new line or be separated from the previous statement by a colon.

```
Rem This is a comment
Rem This is another comment
formula = 10 'So is any text after an apostrophe
formula = 20 : Rem This is also a comment
'Comments can occur after the formula text
```

Fields

Many of the fields you use when creating your report can also be referred to in your formulas. For example, database, parameter, running total, SQL expression, summary, and group name fields can all be used in a formula. You can also refer to other formula fields in your formula.

The easiest way to insert a field into your report is to double-click a field's name in the Report Fields tree. This ensures that the correct syntax for the field is used.

How fields appear in formulas

Database, parameter, formula, running total and SQL expression fields have their names surrounded by braces. Database field names are taken from the database. For example:

- database field: `{Employee.Last Name}`

Parameter, formula, running total, and SQL expression field names are specified when the fields are created.

- parameter fields also includes a question mark: `{?my parameter field}`
- formula fields include an at sign: `{@another formula }`
- running totals fields include a pound sign: `{#my running total}`

- SQL expression fields include a percent sign: `{%my SQL expression}`

Summary and group name fields look like function calls. However, they are really shorthand notation for a report field.

- sum summary field: `Sum({Orders.Order Amount}, {Orders.Ship Via})`
- group name field: `GroupName({Orders.Ship Via})`

Sample formulas using fields

The formula in this example uses the Xtreme database. To find out how many days it takes to ship the product from the date when the order was placed, subtract the ship date database field from the order date database field:

```
Rem A formula that uses database fields
formula = {Orders.Ship Date} - {Orders.Order Date}
```

To find the total dollar amount of a given product that was ordered, multiply its unit price by the quantity ordered:

```
formula = {Orders Detail.Unit Price} * _
          {Orders Detail.Quantity}
```

Note: The example uses the line continuation character " _ " (space underscore). See "Statements" on page 405.

To calculate a sale price of 80 percent of the original unit price:

```
formula = {Orders Detail.Unit Price} * 0.80
```

Statements

A Basic syntax formula consists of a sequence of statements. Each statement must be separated from the previous statement by either a new line or a colon. Typically, each statement takes one line, but you can continue a statement onto the next line by using the line continuation character, which is a space followed by an underscore.

For example:

```
'Declare a variable x to hold a number
Dim x As Number

'Assign the value of 30 to x
x = 10 + 10 + 10

'This also assigns the value of 30 to x
x = 10 + _
    10 + 10

'Line continuation characters _
 can also be used in comments
Dim y as String

'Three statements separated by two colons
y = "Hello" : x = 30 : formula = True
```

Assignment

Use the equal sign (=) when making assignments. The keyword *Let* can be optionally included as well. For example:

```
x = 10
Let y = 20
```

Simple data types

The simple data types in Crystal Reports are Number, Currency, String, Boolean, Date, Time and DateTime.

Number

Enter numbers without any comma separators or currency symbols. (Generally, you would want to have formatted numbers appearing as the result of a formula and not in the formula itself.)

Examples

```
10000
-20
1.23
```

Currency

Use the CCur function to create a Currency value. The initial C in CCur stands for *convert* and it can be used to convert Number values to Currency values.

Examples

```
CCur (10000)
CCur (-20)
CCur (1.23)
```

String

Strings are used to hold text. The text must be placed between double quotation marks (") and cannot be split between lines. If you want to include double quotes in a string, use two consecutive double quotation marks.

```
"This is a string."
"123"
"The word ""hello"" is quoted."
```

You can extract individual elements or substrings from a string by specifying the character position or a range of character positions. Negative values are allowed; they specify the position starting from the end of the string.

```
"hello" (2) 'Equal to "e"
"hello" (-5) 'Equal to "h"
"604-555-1234" (1 to 3) 'Equal to "604"
"abcdef" (-3 to -1) 'Equal to "def"
```

You can also extract substrings from a string using the Left, Right and Mid functions.

Boolean

The Boolean values are:

```
True
False
```

Note: *Yes* can be used instead of True and *No* instead of False.

Date, Time, and DateTime

The DateTime type can hold date-times, dates only or times only. The Date type holds dates only and the Time type holds times only. The Date and Time types are more efficient than the DateTime type, and so can be used in situations where the added functionality and flexibility of the DateTime type is not needed.

Visual Basic does not support separate types for holding dates only or times only. The Basic syntax DateTime type is similar to Visual Basic's Date type.

You can create DateTime values directly using the date-time literal construction. It is formed by typing in the date-time between 2 pound (#) signs. Many different formats are supported, as in Visual Basic.

Note: Date-time literals cannot be split between lines.

Examples

```
#8/6/1976 1:20 am#
#August 6, 1976#
#6 Aug 1976 13:20:19#
#6 Aug 1976 1:30:15 pm#
#8/6/1976#
#10:20 am#
```

Even though `#10:20 am#` looks like it could have the Time type and `#8/6/1976#` looks like it could have the Date type, they do not. They both have the DateTime type, as do all date-time literals. For example, you can think of `#10:20 am#` as a DateTime value with a null date part. To convert it to the Time type use `CTime (#10:20 am#)`.

Instead of using date-time literals you can use CDateTime to convert a String to a DateTime. For example:

```
CDateTime ("8/6/1976 1:20 am")
CDateTime ("10:20 am")
```

However, there is one key difference between using date-time literals and the above usage of CDateTime. Date-time literals always use U.S. English date formats rather than settings from the locale of the particular computer on which Crystal Reports is running. Thus, the date-time literal examples above would work on all computers. On the other hand, on a French system, you could use constructions like:

```
CDateTime ("22 aout 1997") 'Same as #Aug 22, 1997#
```

Date values can be constructed with CDate and Time values with CTime:

```
CDate ("Aug 6, 1969")
CDate (1969, 8, 6) 'Specify the year, month, day
'Converts the DateTime argument to a Date
CDate (#Aug 6, 1969#)
CTime ("10:30 am")
CTime (10, 30, 0) 'Specify the hour, minute, second
CTime (#10:30 am#)
```

Range data types

Ranges are designed to handle a spectrum of values. Range types are available for all the simple types except for Boolean. That is: Number Range, Currency Range, String Range, Date Range, Time Range and DateTime Range. You can generate ranges using the To, _To, To_, _To_, Is >, Is >=, Is < and Is <= keywords. In general, To is used for ranges with 2 endpoints, and Is is used for open ended ranges (only one endpoint). The underscores are used to indicate whether or not the endpoints are in the range.

Examples of Number Range values

The range of numbers from 2 to 5 including both 2 and 5

```
2 To 5
```

The range of numbers from 2 to 5, not including 2 but including 5

```
2 _To 5
```

All numbers less than or equal to 5

```
Is <= 5
```

All number less than 5

```
Is < 5
```

Examples of DateTime Range values

```
#Jan 5, 1999# To #Dec 12, 2000#
Is >= #Jan 1, 2000#
```

Using ranges in formulas

There are twenty-seven functions in Crystal Reports that specify date ranges. For example, the function LastFullMonth specifies a range of date values that includes all dates from the first to last day of the previous month. So if today's date is September 15, 1999 then LastFullMonth is the same as the range value CDate (#Aug 1, 1999#) To CDate (#Aug 31, 1999#).

Ranges are often used with If or Select statements. The following example computes student letter grades based on their test scores. Scores greater than or equal to 90 receive an "A", scores from 80 to 90, not including 90 receive a "B" and so on.

```
Rem Compute student letter grades
Select Case {Student.Test Scores}
   Case Is >= 90
      formula = "A"
   Case 80 To_ 90
      formula = "B"
   Case 70 To_ 80
      formula = "C"
   Case 60 To_ 70
      formula = "D"
   Case Else
      formula = "F"
End Select
```

The above example uses the Select statement which is discussed in more detail in "Control structures" on page 426. You can check if a value is in a range by using the In operator. For example:

```
formula = 5 In 2 To 10 'True
formula = 5 In 2 To_ 5 'False
formula = 5 In 2 To 5 'True
```

The Maximum and Minimum functions can be used to find the endpoints of a range:

```
formula = Maximum (2 To 10) 'Returns 10
```

Array data types

Arrays in Crystal Reports are ordered lists of values that are all of the same type. These values are known as the array's *elements*. The elements of an array can be any simple type or range type. One way to create an array is using the Array function.

Examples

An array of 3 Number values. The first element is 10, the second is 5 and the third is 20.

```
Array (10, 5, 20)
```

An array of 7 String values:

```
Array ("Sun", "Mon", "Tue", "Wed", "Th", "Fri", "Sat")
```

An array of 2 DateTime Range values (note the line continuation character which is used for readability):

```
Array (#Jan 1, 1998# To #Jan 31, 1998#, _
       #Feb 1, 1999# To #Feb 28, 1999#)
```

You can extract individual elements out of an array using parentheses containing the index of the element you want. This is called *subscripting* the array:

```
Array (10, 5, 20) (2) 'Equal to 5
```

Note: Arrays in Basic syntax are indexed from 1 (this means the first element has index 1). This is unlike in Visual Basic where arrays are indexed from 0 by default. However, in Visual Basic, arrays can be indexed from 1 by using the *Option Base* statement.

Number ranges can also be used to subscript arrays. The result is another array. For example:

```
Array (10, 5, 20) (2 To 3) 'Equal to Array (5, 20)
```

Arrays are most useful when used with variables. Using variables, you can change the individual elements of an array and resize the array to accommodate more elements. This capability significantly expands the capabilities of the formula language to do complex calculations.

For example, you can accumulate database field values into a global array variable in a detail level formula, and then use a formula in a group footer to perform a calculation based on those values. This allows you to perform a wide variety of customized summary operations.

Variables

This section describes the key components of variables and shows you how to create variables and assign values to them.

Variable overview

A variable represents a specific data item, or value, and acts as a placeholder for that value. When a formula encounters a variable, the formula searches for the value of the variable and uses it in the formula. 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. Because of this flexibility, it is necessary for you to declare variables before you use them so that Crystal Reports is aware of them and understands how you intend to use them.

Example of a variable

If you wanted to report on customers by area code, you could create a variable that extracts the area code from a customer fax number. The following is an example of a variable called *areaCode*:

```
Dim areaCode As String
areaCode = Left ({Customer.Fax}, 3)
Rem could also use: areaCode = {Customer.Fax} (1 To 3)
```

The first line of the variable example is the variable declaration; it gives the variable a name and type. The database field {Customer.Fax} is a String field and the Left function extracts the first 3 characters from its current value. The variable areaCode is then assigned this value.

Variable declarations using Dim

Before using a variable in a formula, you must declare it. A variable can hold values of a given type. The allowed types are the seven simple types (Number, Currency, String, Boolean, Date, Time and DateTime), the six range types (Number Range, Currency Range, String Range, Date Range, Time Range and DateTime Range) and variables that hold arrays of the previously mentioned types. This gives a total of 26 different types that a variable can have. When you declare a variable, you also specify its name. A variable cannot have the same name as any function, operator or other keyword that is valid for Basic syntax. For example, your variable cannot be named Sin, Mod or If because Sin is a built in function, Mod is a built in operator and If is a built in keyword. When typing formulas in the formula editor, the names of the built-in functions, operators, and other keywords are highlighted in a different color. This makes it easy to check if the variable name conflicts.

Once a variable is declared, it can be used in the formula. For example, you might want to assign it an initial value:

```
Dim x As Number 'Declare x to be a Number variable
x = 10 'Assign the value of 10 to x
```

Variables can only hold values of one type

A variable can only hold values of one type. For example, if a variable holds a Number value you cannot later use it to hold a String.

Example

```
Dim y As String
y = "hello"

'OK - the Len function expects a String argument
formula = Len (y)

'Error- y can only hold String values
y = #Jan 5, 1993#
```

```
'Error- y can only hold String values
y = Array ("a", "bb", "ccc")
```

```
'Error- the Sin function expects a Number argument
formula = Sin (y)
```

You can declare more than one variable per statement by separating their declaration by commas:

```
Dim x As Number, y as String, z as DateTime Range
x = 10 : y = "hello"
z = #Jan 1, 1999# To #Jan 31, 1999#
```

Declaring variables without immediately specifying their type

In general, the type of a variable does not need to be explicitly given when declaring it. In such cases, the variable's type is determined by the first assignment that is made to it. This is similar to the special variable *formula*. This is different from in Visual Basic where a variable whose type is not given at declaration automatically has the Variant type. However, in practice, it means that you can write formulas in a similar style to what you would do if using a Variant in Visual Basic.

```
Dim p 'The type of p is not known yet
p = "bye" 'The type of p is now set to be String
```

```
Dim q 'The type of q is not known yet
q = Array ("hello", p) 'q is a String Array
```

```
'Error- p is a String variable and cannot hold a Number
p = 25
```

```
Dim r
'r is a Number variable, and holds the value 5
r = (10 + 5) / 3
```

```
'The types of a and c are not known yet
Dim a, b As Boolean, c
b = False
'The type of a is now set to Boolean
'and its value is False
a = b
'The type of c is now set to Number and its value is 17
c = 2 + 3 * 5
```

Examples of declaring and initializing range variables

```
Dim gradeA, quarter
'The type of gradeA is set to Number Range
gradeA = 90 To 100
'The type of quarter is set to Date Range
quarter = CDate (1999, 10, 1) To CDate (1999, 12, 31)
```

Variable Scope

Variable scopes are used to define the degree to which variables in one formula are made available to other formulas. There are three levels of scope in Crystal Reports: local, global and shared. Every variable has a scope, and this scope is specified when the variable is declared.

Local Variables

Variables with local scope, also known as local variables, are declared using either the Dim or Local keywords. For example, all the declarations in the previous section using Dim were declaring local variables. Another example:

```
Local x As Number 'equivalent to Dim x As Number
```

Local variables are restricted to a single formula and a single evaluation of that formula. This means that you cannot access the value of a local variable in one formula from a different formula.

Example

```
Rem Formula A
Local x as Number
x = 10
formula = x
```

```
Rem Formula B
EvaluateAfter ({@Formula A})
Local x as Number
formula = x + 1
```

The function call EvaluateAfter ({@Formula A}) ensures that Formula B will be evaluated after Formula A is evaluated. Formula A returns a value of 10 and Formula B returns a value of 1. Formula B does not have access to Formula A's x and thus cannot use the value of 10 and add 1 to it; instead, it uses the default value for the uninitialized local variable x found in Formula B, which is 0, and adds 1 to it to get 1.

You can also create local variables with the same name but different types in different formulas. For example, the type declarations in formulas A and B do not conflict with:

```
Rem Formula C
Local x as String
x = "hello"
formula = x
```

Local variables are the most efficient of the three scopes. In addition, they do not interfere with one another in different formulas. For these reasons, it is best to declare variables to be local whenever possible.

Global variables

Global variables use the same memory block to store a value throughout the main report. This value is then available to all formulas that declare the variable, except for those in subreports. Declare a global variable as in the following example:

```
Global y As String
```

Since global variables share their values throughout the main report, you cannot declare a global variable in one formula with one type and then declare a global variable with the same name in a different formula with a different type.

Example

```
Rem Formula A
Global z As Date
z = CDate (1999, 9, 18)
formula = 10
```

```
Rem Formula B
Global z As Number
formula = True
```

In this case, if you enter and save Formula A first, Crystal Reports will return an error when you check or try to save Formula B. This is because the declaration of the Global variable z as a Number conflicts with its earlier declaration in Formula A as a Date.

Using Global variables

Global variables are often used to perform complex calculations where the results of a formula depend upon the grouping and page layout of the actual printed report. This is accomplished by creating several formulas, placing them in different sections of the report, and having the different formulas interact via global variables.

Here is an example of the sort of effects that can be produced:

```
Rem Formula C
Global x as Number
x = 10
formula = x
```

```
Rem Formula D
'call the function WhileReadingRecords
WhileReadingRecords
Global x as Number
x = x + 1
formula = x
```

If Formula C is placed in the Report Header and then Formula D is placed in a detail section, Formula C will be evaluated before Formula D. Formula C will be evaluated once and then Formula D will be evaluated for each record appearing in the detail section. Formula C returns 10. For the first detail record, Formula D returns 11. This is because the value 10 of x is retained from when it was set by Formula C. Formula

D then adds 1 to this value, setting x to 11 and then returns 11. For the second detail record, formula D return 12, adding 1 to the previously retained value of x which was 11. This process continues for the remaining detail records.

The call to WhileReadingRecords tells Crystal Reports to re-evaluate Formula D as it reads in each record of the report. Otherwise, since the formula does not contain any database fields, the program will evaluate it only once before reading the records from the database. The formula will then return the value 11 instead of 11, 12, 13, ... as the successive records are processed.

If the statement `x = x + 1` is replaced by `x = x + {Orders Detail.Quantity}`, you create the effect of a running total based on {Orders Detail.Quantity}, although it is one starting at 10 rather than 0 because of Formula C. In this case, you can omit the call to WhileReadingRecords, since it will automatically occur because the formula contains a database field.

Shared variables

Shared variables use the same memory block to store the value of a variable throughout the main report and all of its subreports. Thus shared variables are even more general than global variables. To use a shared variable, declare it in a formula in the main report as in the following example:

```
Shared x As Number
x = 1000
```

and declare it in a formula in the subreport as in the following example:

```
Shared x as Number
```

In order to use shared variables the variable must be declared and assigned a value before it can be passed between the main report and the subreport.

Declaring array variables

There are several different ways of declaring array variables. The first way is to use empty parentheses and explicitly specify the type of the array:

```
'Declare x to be a Global variable
'of Number Array type
Global x () As Number
'Initialize x
x = Array (10, 20, 30)

'Declare y to be a Shared variable
'of String Range Array type
Shared y () As String Range
'Initialize y
y = Array ("A" To "C", "H" To "J")
```

The second way is to declare the variable without specifying that it is an array and without giving its type and waiting for the first assignment to the variable to completely specify its type:

```
'Declare y to be a Local variable
'but do not specify its type
Dim y
'The type of y is now set to be a String Array
y = Array ("Sun", "Mon", "Tue", "Wed", "Th", _
           "Fri", "Sat")
```

The third way is to declare that the variable is an array but not specify its type fully until the first assignment. Assuming the declaration of y above:

```
'Declare z to be a Local variable that is an Array
Local z()
'z is set to Array ("Mon", "Tue") and is a String Array
z = y(2 to 3)
```

The fourth way is to explicitly specify the size of the array during the declaration. If you use this technique, the array is automatically created and default values are used to fill the array. For example, for a Number Array, each element is initialized to 0 and for a String array each element is initialized to the empty string "". Since this type of declaration actually creates the array, you must specify its type with the As clause so that Crystal Reports knows how much storage space to reserve for the array.

```
Dim a(2) As String
'Assign a value to the first element of the array a
a(1) = "good"
a(2) = "bye"
'The & operator can be used to concatenate strings
'the formula returns the String "goodbye"
formula = a(1) & a(2)
```

Arrays and For/Next loops

Arrays are commonly used with For/Next loops. The following example creates and then uses the array `Array (10, 20, 30, ..., 100)` using a For/Next loop. See "For/Next loops" on page 430 for more details.

```
Dim b (10) As Number
Dim i
For i = 1 To 10
   b(i) = 10 * i
Next i
formula = b(2) 'The formula returns the Number 20
```

Using array variables

You can assign values to elements of an array and also use the values of the elements for other computations:

```
Global x() As String
x = Array ("hello", "bye", "again")
'Now x is Array ("hello", "once", "again")
x (2) = "once"
'The statement below would cause an error if not
'commented out since the array has size 3
'x (4) = "zap"
'The formula returns the String "HELLO"
formula = UCase (x (1))
```

The Redim and Redim Preserve keywords can be used to resize an array, which is useful if you want to add extra information to it. Redim erases the previous contents of the array first before resizing it whereas Redim Preserve preserves the previous contents.

```
Dim x () As Number
Redim x (2) 'Now x is Array (0, 0)
x (2) = 20 'Now x is Array (0, 20)
Redim x (3) 'Now x is Array (0, 0, 0)
x (3) = 30 'Now x is Array (0, 0, 30)
Redim Preserve x (4) 'Now x is Array (0, 0, 30, 0)
formula = "finished"
```

Default values for the simple types

An uninitialized variable will have the default value for its type. In general, it is not a good programming practice to rely on the default values of types. For example, initialize all local variables in your formula, initialize all global variables in a formula placed in the Report Header, and initialize all shared variables in a formula placed in the Report Header of the main report.

When an array is resized using the Redim keyword, the entries are filled with default values for the type.

Default values

Number

```
0
```

Currency

```
CCur (0)
```

String

```
"" 'The empty string
```

Date

```
CDate (0, 0, 0) 'The null Date value
```

Time

The null Time value. Value held by an uninitialized Time variable.

DateTime

The null DateTime value. Value held by an uninitialized DateTime variable.

Note: It is not recommended that your formulas rely on the values of uninitialized range or array variables.

Automatic type conversions

Generally in Crystal Reports, values of one type cannot be used where values of another type are expected without explicitly supplying a type conversion function. For example:

```
Dim postalCode as String
'Error- assigning a Number value to a String variable
postalCode = 10025
'OK- use the type conversion function CStr
'to create "10025"
postalCode = CStr (10025, 0)
```

However, there are a few conversions that are made automatically:

- Number to Currency
- Date to DateTime
- Simple type to Range value of the same underlying simple type

For example, the following assignments are correct:

```
Dim cost As Currency
'Same as: cost = CCur (10)
cost = 10
```

```
Dim orderDate As DateTime
'Same as: orderDate = CDateTime (1999, 9, 23, 0, 0, 0)
orderDate = CDate (1999, 9, 23)
```

```
Dim aRange As Number Range
'Same as: aRange = 20 To 20
aRange = 20
```

```
Dim aRangeArray () As Number Range
'Same as :
'aRangeArray = Array (10 To 10, 20 To 25, 2 To 2)
aRangeArray = Array (10, 20 To 25, 2)
```

Note: The opposite conversions are not allowed. For example:

```
Dim num As Number
num = 5 + CCur (10) 'Error
'OK- convert to Number type using the CDbl function
num = CDbl (5 + CCur (10))
```

5 is converted to CCur (5) and added to CCur (10) to make CCur (15). However, this Currency value cannot be automatically assigned to the Number variable num since automatic conversions from Currency to Number are not allowed. Similarly, functions accepting a Currency argument can be supplied a Number argument instead, and the Number argument will be converted to a Currency, whereas functions accepting a Number argument cannot be supplied a Currency argument without first explicitly converting the Currency to a Number using CDbl.

Functions

When using a function in a formula, type the name of the function and supply the arguments required. For example, the Len function requires a String argument and computes the length of the string.

```
Dim x As String
x = "hello"
formula = Len (x) 'The formula returns the Number 5
```

Supplying arguments of the incorrect type required by the function produces an error. For example, calling Len (3) would produce an error since Len does not accept a Number argument. Functions sometimes can accept different numbers of arguments or types of arguments. For example, the CDate function could accept a single String argument to form a Date value or 3 Number values holding the year, month and day respectively and form a Date value from them. See “Date, Time, and DateTime” on page 407.

Example with the Mid function

```
Dim x as String
x = "hello"
'Start at position 2, go to the end of the string
formula = Mid (x, 2) 'formula is now "ello"
'Start at position 2, extract 1 character
formula = Mid (x, 2, 1) 'formula is now "e"
```

The classes of functions are: Math, Summary, Financial, String, Date/Time, Date Range, Array, Type Conversion, Programming Shortcuts, Evaluation Time, Print State, Document Properties and Additional Functions. There are also some functions specific to conditional formatting formulas.

Tip: Descriptions of the functions supported by Crystal Reports can be found in the online help.

Functions similar to Visual Basic functions of the same name

The Math, Financial, String, Date/Time, Type Conversion and Programming Shortcuts groups consist mainly of functions that are familiar to Visual Basic users. Most of the functions are intended to work in the same way as the Visual Basic function of the same name.

Note:

- Sometimes functions will have more overloads than are available in Visual Basic.

  For example, the CDate function supports the Visual Basic overload of creating a Date value from a String value, such as CDate ("Sept 18, 1999") but it also supports an overload of creating a Date value by supplying the year, month and day as Number arguments e.g. CDate (1999, 9, 18). The overloads are indicated in the Functions tree.
- Some functions that are supported by Basic syntax are not listed in the Basic syntax Functions tree. This is because they are equivalent to Basic syntax functions that are already listed in the tree.

  For example, the Len function is the normal Basic syntax and Visual Basic function for finding the length of a string. However, the Length function is listed in the Functions tree and works the same as the Len function. Length is the traditional Crystal syntax function for this action and it is included for the convenience of Crystal syntax users migrating formulas to Basic syntax.

Summary functions

The Summary function group provides functions for creating summary fields such as:

```
Sum({Orders.Order Amount}, {Orders.Ship Via})
```

Summary fields are normally created using the Insert Summary or Insert Grand Total dialogs. They then appear in the Report Fields tree, and can be used in a formula by double clicking there. However, they do not need to be created in this way. You can create a summary field exclusively for use by your formula by filling in the arguments to one of the functions in the Summary functions section. However, any groups that refer to summary fields must already exist in the report.

Tip: Descriptions of the summary functions supported by Crystal Reports can be found in the online help.

Date ranges

For additional information refer to “Range data types” on page 408.

Date ranges produced by these functions depend on the current date. For example, if today's date is September 18, 1999, then LastFullMonth is the Date Range value:

```
CDate(#Aug 1, 1999#) To CDate(#Aug 31, 1999#)
```

This functionality is often useful, but if you want to determine a date range based on a database field such as {Orders.Order Date}? The Date/Time functions can be used instead.

Example

```
Dim d As Date
d = CDate ({Orders.Order Date})
Dim dr As Date Range
dr = DateSerial (Year(d), Month(d) - 1, 1) To _
     DateSerial (Year(d), Month(d), 1 - 1)
'At this point dr is the Date Range value holding
'the last full month before {Orders.Order Date}
```

The DateSerial function makes this easy because you don't have to worry about special cases. It never lets you create an invalid date. For example, DateSerial (1999, 1 - 1, 1) is December 1, 1998. Note that in the above example, {Orders.Order Date} is actually a DateTime field and so the CDate function is used to convert it to a date by truncating the time part.

Array functions

The array functions compute summaries of an array's elements. For example, the Sum function when applied to an array returns the sum of the elements of the array.

Example

The following formula returns 100:

```
formula = Sum (Array (10, 20, 30, 40))
```

Evaluation Time functions

These are the report specific functions: BeforeReadingRecords, WhileReadingRecords, WhilePrintingRecords and EvaluateAfter. You can use these functions to guide Crystal Reports as to when your formula should be evaluated.

Should the formula be evaluated before retrieving the records from the database, while reading the records from the database but before the records have been grouped, sorted and summarized, or while printing the report, when the records are grouped, sorted and summarized? In general, Crystal Reports sets an appropriate evaluation time for your formula, based on how much information the formula needs. For example, if a formula uses a database field, then it cannot be evaluated before the records are read from the database. However, you sometimes need to force a later evaluation time than normal to get the desired effect. See "Using Global variables" on page 414 for an example.

Normally, the returned value of a function is used further in a formula. However, evaluation time functions are called to change the internal behavior of Crystal Reports and their return value is not used. They can be called by just placing their name in a separate statement, optionally preceded by the keyword Call.

```
WhilePrintingRecords
Call WhilePrintingRecords
```

Print State functions

These are once again reporting-specific functions. For example, the notation {Orders.Order Date} refers to the value of the field in the current record where PreviousValue ({Orders.Order Date}) refers to the value in the immediately preceding record. NextValue ({Orders.Order Date}) refers to the value in the next record. IsNull ({Orders.Order Date}) checks if the field's value is null.

Other examples are PageNumber and TotalPageCount. These can be used to access pagination information about your report.

Document Properties functions

These are reporting specific functions that refer to the report document as a whole. For example, PrintDate and ReportTitle.

Additional functions

These are functions that are in User Function Libraries (UFLs). A UFL is a separate dynamic link library or Automation server that you create and Crystal Reports uses to add your own customized functions to the formula language. Writing a UFL is more involved than writing a formula using Basic or Crystal syntax. See Crystal Reports Developer's Help for more information.

Note: Using UFLs makes your reports less portable because you must distribute your UFL along with the report.

Conditional formatting functions

When writing a conditional formatting formula, you may want to use the additional functions that appear at the top of the Functions tree.

Example

If you wanted to format the {Customer.Last Year's Sales} field so that sales of more than $100,000 are printed in green and sales of less than $15,000 are printed in red and all else are printed in black.

```
Rem Conditional formatting example 1
If {Customer.Last Year's Sales} > 100000 Then
   formula = crGreen
ElseIf {Customer.Last Year's Sales} < 15000 Then
   formula = crRed
Else
   formula = crBlack
End If
```

Since this is a font color formatting function, the list of Color Constants appears in the Functions tree. This example uses three: crGreen, crRed and crBlack. You could have used the actual numeric values of the color constants instead. For example, crRed is 255 and crGreen is 32768. However, the formula is easier to understand using the color constants. All constant functions in Basic syntax have the "cr" prefix.

Note: Some formatting attributes do not use constant functions. For example, if you wanted to not print {Customer.Last Year's Sales} values if the sales were less than $50,000, you could write the following conditional formatting formula for the suppress attribute:

```
Rem Conditional formatting example 2
If {Customer.Last Year's Sales} < 50000 Then
   formula = True 'suppress the value
Else
   formula = False 'do not suppress the value
End If
```

Or more simply:

```
Rem Conditional formatting example 3 -
Rem equivalent to example 2
formula = {Customer.Last Year's Sales} < 50000
```

If the last year's sales are less than $50,000, then the expression

```
{Customer.Last Year's Sales} < 50000
```

is True, and so the formula returns True. On the other hand, if the last year's sales are greater than or equal to $50,000, then

```
{Customer.Last Year's Sales} < 50000
```

is False and so the formula returns False.

General purpose conditional formatting functions

There are three general purpose conditional formatting functions:

- CurrentFieldValue
- DefaultAttribute
- GridRowColumnValue.

These functions are displayed at the top of the Functions tree when appropriate. DefaultAttribute can be used for any formatting formula. CurrentFieldValue can be used for any formatting formula where you are formatting a field value. GridRowColumnValue can be used for formatting a formula where you are formatting a field value in a Cross-Tab or OLAP grid.

CurrentFieldValue enables you to conditionally format Cross-Tab or OLAP grid cells based on their value. GridRowColumnValue enables you to conditionally format the cells of a Cross-Tab or OLAP grid based on row or column headings values. These two functions are essential in some situations as there is no other way in the formula language to refer to these fields.

Tip: Descriptions of the conditional formatting functions supported by Crystal Reports can be found in the online help.

Example

If you wanted Cross-Tab cells to be suppressed if the values are less than 50,000:

```
Rem Conditional formatting example 4
formula = CurrentFieldValue < 50000
```

Operators

Arithmetic operators

The arithmetic operators are addition (+), subtraction (-), multiplication (*), division (/), integer division (\), modulus (Mod), negation (-) and exponentiation (^). Arithmetic operators are used to combine numbers, numeric variables, numeric fields and numeric functions to get another number.

Examples

```
'Outstanding preferred stock as a percent of
'common stock
formula = ({Financials.Preferred Stock} / _
           {Financials.Common Stock}) * 100
'The square root of 9, Sqr(9), is 3.
'The formula returns 17.
formula =  7 + 2 * 3 - 2 + Sqr(6 + 3) * Len("up")
```

Tip: Descriptions of the arithmetic operators supported by Crystal Reports can be found in the online help.

Order of precedence

When you create arithmetic expressions that involve several operators, the order that the program evaluates the various elements of the expression becomes important.

In general, the program evaluates expressions in the following order:

- from left to right
- follows the rules of precedence from basic math.

Example

Multiplication and division are performed first from left to right. Then addition and subtraction are performed. For example, 5 + 10 * 3 = 5 + 30 = 35.

You can change this order of precedence by using parentheses. For example, (5 + 10) * 3 = 15 * 3 = 45. If you are unsure of the order of precedence, it is a good idea to clarify your intentions with parentheses.

The arithmetic operators in Crystal Reports have the same order of precedence as in Visual Basic. Here is the list, from highest precedence to lowest:

- Exponentiation (^)
- Negation (-)
- Multiplication and division (*, /)
- Integer Division (\)
- Modulus (Mod)
- Addition and subtraction (+, -).

Comparison operators

The comparison operators are equal (=), not equal (<>), less than (<), less than or equal to (<=), greater than (>) and greater than or equal to (>=).

Comparison operators are usually used to compare operands for a condition in a control structure such as an If statement. Comparison operators as a group all have lower precedence than the arithmetic operators. For example, expressions like 2 + 3 < 2 * 9 are the same as (2 + 3) < (2*9).

Boolean operators

The Boolean operators are, in order of precedence from greatest to lowest: Not, And, Or, Xor, Eqv and Imp. Boolean operators are typically used with comparison operators to generate conditions for control structures. Boolean operators as a group have lower precedence than the comparison operators. Thus for example, the expression 2 < 3 And 4 >= -1 is the same as (2 < 3) And (4 >= -1).

Null fields and how to use IsNull

The {Product.Color} field contains both basic colors such as “red” and “black” and more descriptive two word colors such as “steel satin” and “jewel green”. Here’s an example of a formula that writes out “basic” for the basic colors and “fancy” for the others.

```
If InStr({Product.Color}, " ") = 0 Then
    formula = "basic"
Else
    formula = "fancy"
End If
```

The function call to InStr searches the {Product.Color} string for a space. If it finds a space, it returns the position of the space, otherwise it returns 0. Since basic colors are only one word with no spaces, InStr will return 0 for them.

Dealing with null values

For some products, such as the Guardian Chain Lock, a color value was not recorded and so the {Product.Color} field has a null value in the database for that record. In general, when Crystal Reports encounters a null valued field in a formula, it immediately stops evaluating the formula and produces no value. Thus, the Guardian Chain Lock record does not have any word printed beside it. If you want to handle null field values in your formula, you must explicitly do so using one of the special functions designed for handling them: IsNull, PreviousIsNull or NextIsNull. Here is an example of how to fix the above example using IsNull:

```
If IsNull({Product.Color}) Or _
   InStr({Product.Color}, " ") = 0 Then
   formula = "basic"
Else
   formula = "fancy"
End If
```

Relating to operators, when Crystal Reports evaluates the condition:

```
IsNull({Product.Color}) Or _
InStr({Product.Color}, " ") = 0
```

It first evaluates IsNull ({Product.Color}), and when it determines that this is True, it knows that the whole condition is True, and does not need to check whether

```
InStr({Product.Color}, " ") = 0
```

In other words, Crystal Reports will stop evaluating a Boolean expression when it can predict the results of the whole expression. In the following example, the formula guards against attempting to divide by zero in the case that denom is 0:

```
Dim num As Number, denom As Number
...
If denom <> 0 And num / denom > 5 Then
...
```

Note: Visual Basic does not support this technique, since all parts of a Boolean expression in Visual Basic are evaluated, even if not necessary.

Control structures

Formulas without control structures execute each statement in the formula only once. When this happens the formula is evaluated. The statements are executed in a sequential fashion, from the first statement in the formula to the last. Control structures enable you to vary this rigid sequence. Depending upon which control structure you choose, you can skip over some of the statements or repeatedly evaluate some statements depending on certain conditions. Control structures are the primary means of expressing business logic and typical report formulas make extensive use of them.

Basic syntax supports many of the main control structures from Visual Basic with the same syntax. One of the advantages of the Basic language is it is easy to read block notation for control structures. This simplifies the writing and debugging of complex formulas.

If statements

The If statement is one of the most useful control structures. It enables you to evaluate a sequence of statements if a condition is true and evaluate a different sequence of statements if it is not true.

Note: When formatting with conditional formulas, always include the Else keyword; otherwise, values that don't meet the If condition may not retain their original format. To prevent this, use the DefaultAttribute function (`If...Else formula = DefaultAttribute`).

Example

A company plans to pay a bonus of 4 percent to its employees except for those who work in Sales who will receive 6 percent. The following formula using an If statement would accomplish this:

```
Rem Multi-line If example 1
If {Employee.Dept} = "Sales" Then
   formula = {Employee.Salary} * 0.06
Else
   formula = {Employee.Salary} * 0.04
End If
```

In this example, if the condition `{Employee.Dept} = "Sales"` evaluates as true, then the

```
formula = {Employee.Salary} * 0.06
```

statement is processed. Otherwise the statement following the Else, namely the

```
formula = {Employee.Salary} * 0.04
```

is processed.

Suppose another company wants to give employees a 4% bonus, but with a minimum bonus of $1,000. The following example shows how. Notice that the Else clause is not included; it is optional, and not needed in this case.

```
Rem Multi-line If example 2
formula = {Employee.Salary} * 0.04
If formula < 1000 Then
   formula = 1000
End If
```

Now suppose that the previous company also wants a maximum bonus of $5,000. You now need to use an ElseIf clause. Notice that ElseIf is all one word. The following example has only one ElseIf clause, but you can add as many as you need.

Note: There is a maximum of one Else clause per If statement.

The Else clause is executed if none of the If or ElseIf conditions are true.

```
Rem Multi-line If example 3
formula = {Employee.Salary} * 0.04
If formula < 1000 Then
   formula = 1000
ElseIf formula > 5000 Then
   formula = 5000
End If
```

Example

Suppose that a company wants to compute an estimate of the amount of tax an employee needs to pay and write a suitable message. Income below $8,000 is not taxed, income between $8,000 and $20,000 is taxed at 20%, income between $20,000 and $35,000 is taxed at 29%, and income above $35,000 is taxed at 40%.

```
Rem Multi-line If example 4
Dim tax As Currency, income As Currency
income = {Employee.Salary}
Dim message As String
If income < 8000 Then
    tax = 0
    message = "no"
ElseIf income >= 8000 And income < 20000 Then
    message = "lowest"
    tax = (income - 8000)*0.20
ElseIf income >= 20000 And income < 35000 Then
    message = "middle"
    tax = (20000 - 8000)*0.20 + (income - 20000)*0.29
Else
    message = "highest"
    tax = (20000 - 8000)*0.20 + (35000 - 20000)*0.29 + _
        (income - 35000)*0.40
End If

Dim taxStr As String
Rem use 2 decimal places
Rem and use the comma as a thousands separator
taxStr = CStr (tax, 2, ",")
formula = "You are in the " & message & _
          " tax bracket. " & _
          "Your estimated tax is " & taxStr & "."
```

Notice, the use of variables to simplify the logic of the computation. Also, notice that there are two statements that are executed when one of the conditions are met; one assigns the tax variable, and the other assigns the message variable. It is often useful to have multiple statements executed as a result of a condition.

Single-line and multi-line If statements

There are two kinds of If statement, the single-line if statement and the multi-line if statement. Starting on a new line after the first *Then* turns your If statement into a multi-line If statement. Otherwise it is a single-line If statement. The multi-line If statement always includes an End If whereas the single line If statement does not.

Note: Because of the use of line-continuation characters, single-line If statements do not need to be on a single line. In general, it is preferable to use multi-line If statements since they have a clearer layout. However, for simple situations, the single-line If statement is sometimes used.

```
Rem Single-line If example 1
Rem Same result as multi-line If example 1
If {Employee.Dept} = "Sales" Then _
   formula = {Employee.Salary} * 0.06 _
Else _
   formula = {Employee.Salary} * 0.04
```

Here is an example showing various forms of single-line If statements:

```
Rem Single-line If example 2
Dim per As Number, extra As Boolean
per = 2 : extra = False

'An example with no Else clause
If {Employee.Dept} = "Sales" Then per = 10

'More than 1 statement in the Then or Else part can
'be included by separating them with colons
If {Employee.Dept} = "R&D" Then _
   per = 5 : extra = True _
Else _
   per = 3
```

Select statements

The Select statement is similar to an If statement. Sometimes however, you can write formulas that are clear and less repetitive using the Select statement. This example evaluates the {Customer.Fax} field to determine if the area code is for Washington state (206, 360, 509) or British Columbia, Canada (604, 250):

```
Rem Select example 1
Select Case Left ({Customer.Fax}, 3)
   Case "604", "250"
      formula = "BC"
   Case "206", "509", "360"
      formula = "WA"
End Select
```

The expression right after the Select Case keywords is called the *Select condition*. In the above example it is Left ({Customer.Fax}[1 To 3]). The Select statement tries to find the first Case that matches the Select condition, and then executes the statements following it, up until the next Case.

```
Rem Same effect as Select example 1
Dim areaCode As String
areaCode = Left ({Customer.Fax}, 3)
If areaCode In Array ("604", "250") Then
   formula = "BC"
ElseIf areaCode In Array ("206", "509", "360") Then
   formula = "WA"
End If
```

Example

This formula groups the number of Oscar nominations a movie received into low, medium, high or extreme categories and in the process, shows some of the possibilities for the expression lists following the Case labels. Notice the optional Case Else clause. If none of the Case expression lists are matched by the preceding Case clauses, then the Case Else clause is matched. For example, in the following example, if {movie.NOM} is 11, then the formula returns “extreme”.

```
Rem Select example 2
Select Case {movie.NOM}
   Case 1,2,3, Is < 1
      Rem Can have multiple statements in the
      Rem statement blocks
      formula = "low"
   Case 4 To 6, 7, 8, 9
      formula = "medium"
   Case 10
      formula = "high"
   Case Else
      formula = "extreme"
End Select
```

For/Next loops

For/Next loops enable you to evaluate a sequence of statements multiple times. This is unlike the If and Select statements where the program passes through each statement at most once during the formula's evaluation. For/Next loops are best when you know the number of times that the statements needs to be evaluated in advance.

The syntax of the For loop through examples

Sample 1

Suppose you want to reverse the {Customer.Customer Name} string. For example, “City Cyclists” becomes “stsilcyC ytiC”.

```
Rem Reverse a string version 1
formula = ""
Dim strLen
strLen = Len ({Customer.Customer Name})
Dim i
For i = 1 To strLen
   Dim charPos
   charPos = strLen - i + 1
   formula = formula & _
             Mid({Customer.Customer Name}, charPos, 1)
Next i
```

Let us examine how this formula works assuming that the current value of the field {Customer.Customer Name} is "Clean Air". The variable strLen is assigned to be the length of "Clean Air", namely 9. At this time it is also typed to be a Number variable. The variable *i* is known as a *For counter variable* since its value changes with each iteration of the For loop. In other words, it is used to count the iterations of the loop. The For loop will iterate 9 times, during the first time, i is 1, then i is 2, then i is 3 and so on until finally i = 9. During the first iteration, the ninth character of {Customer.Customer Name} is appended to the empty special variable *formula*. As a result formula equals "r" after the first iteration. During the second iteration, the eighth character of {Customer.Customer Name} is appended to formula and so formula equals "ri". This continues until after the ninth iteration, formula equals, "riA naelC" which is the reversed string.

Sample 2

Here is a simpler version of the above formula, that uses a Step clause with a negative Step value of -1. For the "Clean Air" example, i is 9 for the first iteration, 8 for the second, 7 for the third and so on until it is 1 in the final iteration.

```
Rem Reverse a string version 2
formula = ""
Dim i
For i = Len ({Customer.Customer Name}) To 1 Step -1
   formula = formula + _
             Mid({Customer.Customer Name}, i, 1)
Next i
```

Sample 3

The simplest version is to use the built in function StrReverse:

```
Rem Reverse a string version 3
formula = StrReverse ({Customer.Customer Name})
```

The built in String functions in Crystal Reports 8.5 can handle many of the string processing applications which would traditionally be handled using a For/Next loop or some other kind of loop. However, For/Next loops provide the most flexibility and power in processing strings and arrays. This can be essential if the built-in functions do not cover your intended application.

For/Next loop example

Here is a more detailed example of Crystal Reports' string processing capabilities. The Caesar cipher is a simple code that is traditionally credited to Julius Caesar. In this code, each letter of a word is replaced by a letter five characters further in the alphabet. For example, "Jaws" becomes "Ofbx". Notice that "w" is replaced by "b". Since there are not 5 characters after "w" in the alphabet, it starts again from the beginning.

Here is a formula that implements applying the Caesar cipher to the field {Customer.Customer Name} in the Xtreme database:

```
Rem The Caesar cipher

Dim inString 'The input string to encrypt
inString = {Customer.Customer Name}
Dim shift
shift = 5

formula = ""

Dim i
For i = 1 To Len(inString)
   Dim inC, outC
   inC = Mid(inString, i, 1)

   Dim isChar, isUCaseChar
   isChar = LCase(inC) In "a" To "z"
   isUCaseChar = isChar And (UCase (inC) = inC)

   inC = LCase(inC)

   If isChar Then
      Dim offset
      offset = (Asc(inC) + shift - Asc("a")) Mod _
               (Asc("z") - Asc("a") + 1)
      outC = Chr(offset + Asc("a"))
      If isUCaseChar Then outC = UCase(outC)
   Else
      outC = inC
   End If

   formula = formula & outC
Next i
```

In the above example, there is a multi-line If statement nested within the statements block of the For/Next loop. This If statement is responsible for the precise details of shifting a single character. For example, letters are treated differently from punctuation and spaces. In particular, punctuation and spaces are not encoded. Control structures can be nested within other control structures and multiple statements can be included in the statement block of a control structure.

Using the Exit For statement

You can exit from a For/Next loop by using the Exit For statement. The following example searches the Global array names for the name “Fred”. If it finds the name, it returns the index of the name in the array. Otherwise it returns -1. For example, if the names array is:

```
Array ("Frank", "Helen", "Fred", "Linda")
```

Then the formula returns 3.

```
Global names () As String
'The names array has been initialized and filled
'in other formulas
Dim i
formula = -1
'The UBound function returns the size of its array
'argument
For i = 1 to UBound (names)
   If names (i) = "Fred" Then
      formula = i
      Exit For
   End If
Next i
```

Do Loops

Another looping mechanism is the Do loop. A Do loop can be used to execute a fixed block of statement an indefinite number of time.

The 4 different types of Do loops

Type of Do Loop	Explanation	Example

Do While ... Loop	The Do While ... Loop evaluates the condition, and if the condition is true, then it evaluates the statements following the condition.
	When it has finished doing this, it evaluates the condition again and if the condition is true, it evaluates the statements again.
	It continues repeating this process until the condition is false.
		`Do While` *`condition`*
		   `statements`
		`Loop`

Do Until ... Loop	The Do Until ... Loop is similar to the Do While ... Loop except it keeps evaluating the statements *until* the condition is true rather than *while* it is true.	`Do Until` *`condition`*
		   `statements`
		`Loop`

Type of Do Loop	Explanation	Example

Do ... Loop While	The Do ... Loop While evaluates the statements only once.
	It then evaluates the condition, and if the condition is true, evaluates the statements again. This process continues until the condition is false.
		`Do`
		`   statements`
		`Loop While condition`

Do ... Loop Until	Similar to Do ... Loop While except that it evaluates the statements until the condition is true.	`Do`
		`   statements`
		`Loop Until condition`

Note: The Do loops support an Exit Do statement to immediately jump out of the loop. The Exit Do statement is similar to the Exit For in For/Next loops.

Do While ... Loop formula example

The following example searches for the first occurrence of a digit in an input string. If a digit is found, it returns its position, otherwise it returns -1. In this case, the input string is set explicitly to a string constant. If preferred, it could be set to be equal to a String type database field instead.

For example, for the input String, “The 7 Dwarves”, the formula returns 5, which is the position of the digit 7.

```
Dim inString
inString = "The 7 Dwarves"

Dim i, strLen
i = 1
strLen = Len (inString)
formula = -1

Do While i <= strLen And formula = -1
   Dim c As String
   c = Mid (inString, i, 1)
   If IsNumeric (c) Then formula = i
   i = i + 1
Loop
```

While loops

The While loop is similar to the Do While ... Loop except that it does not support an Exit statement. It uses While ... Wend instead of Do While ... Loop as its syntax.

```
While condition
   statements
Wend
```

Safety mechanism for loops

There is a safety mechanism to prevent report processing from hanging due to an infinite loop. Any one evaluation of a formula can have at most 30,000 loop condition evaluations per formula evaluation. For example:

```
Dim i
i = 1
Do While i <= 100000
   If i > {movie.STARS} Then Exit Do
   i = i + 1
Loop
formula = 20
```

If {movie.STARS} is greater than 30,000 then the loop condition (i <= 100000) will be evaluated more than the maximum number of times and an error message is displayed. Otherwise the loop is OK.

Note: The safety mechanism applies on a per formula base, not for each individual loop. For example:

```
Dim i
i = 1
For i = 1 To 10000
   formula = Sin (i)
Next i

Do While i <= 25000
   i = i + 1
Loop
```

The above formula also triggers the safety mechanism since the 30,000 refers to the total number of loop condition evaluations in the formula and this formula will have 10001 + 25001 such evaluations.

Limitations

For reference purposes, here are the sizing limitations of the formula language:

- The maximum length of a String constant, a String value held by a String variable, a String value returned by a function or a String element of a String array is 254 bytes.
- The maximum size of an array is 1000 elements.
- The maximum number of arguments to a function is 1000. (This applies to functions that can have an indefinite number of arguments such as Choose).
- The maximum length of the text of a formula is 64K.
- The maximum number of loop condition evaluations per evaluation of a formula is 30,000. (See safety mechanisms for loops for the precise meaning of this).
- Date-time functions modeled on Visual Basic accept dates from year 100 to year 9999. Traditional Crystal Reports functions accept dates from year 1 to year 9999.

Creating Formulas with Crystal Syntax 24

This chapter provides you with an overview of various parts of a formula. You learn about Crystal syntax and the techniques you can use when creating a formula.

Crystal syntax overview

When creating formulas, you have the option of using either Crystal or Basic syntax. Syntax rules are used to create a correct formula. Almost any formula written with one syntax can be written with the other. Reports can contain formulas that use Basic syntax as well as formulas that use Crystal syntax. For information on whether to use Crystal or Basic syntax, see "Formula syntax" on page 386.

Using the online help

The Crystal Reports online help lists the functions, operators, and control structures available with Crystal syntax and provides a detailed explanation on each of them.

What's new in Crystal syntax

Many new features have been added to Crystal syntax for Crystal Reports version 8.

Note: Equivalent versions of these features are also supported by Basic syntax.

New control structures

Crystal syntax supports three new controls structures: the *Select* expression, the *While* loop, and the *For* loop. Select is a good alternative to the *If* expression when you want to use the result of a single expression to select from a number of alternatives. The While and For loops introduce looping into Crystal Reports. Looping is useful when you want to evaluate some expressions in your formula multiple times. A typical application is to extract information from string database fields for custom formatting in a report. In this case, you loop through the characters of the string field.

New functions inspired by Visual Basic

These new functions are intended to work like their counterparts in Visual Basic. However, you do not need to learn Visual Basic or Basic syntax to use them. They provide functionality that could be useful in many Crystal syntax formulas.

Math functions:

Atn, Cos, Exp, Int, Log, Pi, Rnd, Sin, Sgn, Sqr, Tan

Financial functions:

DDB, FV, IPmt, IRR, MIRR, NPer, NPV, Pmt, PPmt, PV, Rate, SLN, SYD

Programming shortcut functions:

Choose, IFF, Switch

Type conversion functions:

CBool, CCur, CDbl, CStr, CDate, CTime, CDateTime

Date and time functions:

DateValue, TimeValue, DateTimeValue, IsDate, IsTime, IsDateTime, MonthName, WeekdayName, DateAdd, DateDiff, DatePart, DateSerial, Timer, TimeSerial

Array and string functions:

Filter, Replace, Join, Split, UBound, InStrRev, StrReverse

New operators

Some new operators have been added to make writing expressions in Crystal syntax easier and more flexible.

Visual Basic style string concatenation: &

Integer division: \

Exponentiation: ^

Logical operators: Eqv, Imp, Xor

Modulus: Mod

Date-time literals: #Jan 3, 1999 10:20 am#

Expanded capabilities for array processing

Crystal syntax now supports the Redim and Redim Preserve keywords to enable you to resize arrays dynamically. It also supports the ability to assign values to an individual element of an array. In addition, Crystal syntax now uses more efficient algorithms to speed up operations with arrays.

More types of record selection formulas can be pushed down to the server

In particular, there is now support for pushing down a record selection formula that involves the IsNull function and the Or operator. For example, the following selection formulas can be pushed down:

```
{Orders.Order ID} < 1020 Or
{Orders.Order Amount} >= 1000
```

```
IsNull ({Customer.Region}) Or {Customer.Region} = "BC"
```

The formula language now also performs simplification of most constant and parameter expressions in formulas at compile time rather than formula evaluation time. What this means is that record selection formulas such as:

```
{date field} > CurrentDate - 3
```

```
{date field} In CDate ({?year parameter}, 6, 1) To
                CDate ({?year parameter}, 9, 30)
```

can be pushed down to the server since the expressions in italics are replaced by their values when the formula is compiled, and the resulting formula can then be converted to a SQL query.

Formula language support for new functionality specific to Crystal Reports

Conditional formatting of a cell value in a Cross-Tab or OLAP grid based on its row or column values is supported with the new GridRowColumnValue function. The new Percentage summaries are supported in the formula language with the PercentOfSum, PercentOfAverage, PercentOfMaximum, PercentOfMinimum, PercentOfCount and PercentOfDistinctCount functions. You can now use the formula language to conditionally format the font, font style, and font size of a field.

Crystal syntax fundamentals

The result of a formula

The result of a formula, or the value that is printed when the formula is placed in a report, is called the *value returned by the formula*. Every formula in Crystal Reports must return a value. For example, here is a simple Crystal syntax formula that returns a value of 10:

```
10
```

The value returned by a formula can be one of the seven simple data types supported. These are Number, Currency, String, Boolean, Date, Time, and DateTime.

Note: Crystal Reports also supports range types and array types, but these cannot be returned by a formula.

For example, suppose a company has a shipping policy in which orders over $1,000 are insured, but orders below that amount are not insured:

```
//A formula that returns a String value
If {Orders.Order Amount} >= 1000 Then
   "Insured shipping"
Else
   "Regular shipping"
```

Tip: The text following the two forward slashes is a comment for someone reading this formula and is ignored by the Crystal syntax compiler.

The formula returns the text string value “Insured shipping” if the value of the database field {Orders.Order Amount} is greater than or equal to 1000; otherwise, it returns the text string value “Regular Shipping” otherwise.

Crystal syntax is not case-sensitive

For example, the keyword *Then* could equivalently be typed in as *then* or *THEN.* This is true of all variable names, functions, and keywords used in a Crystal syntax formula.

Note: The only exception to this rule is for strings. The string “Hello” is not the same as the string “hello”.

Practice using the Xtreme sample database

Many of the examples in this chapter refer to the Xtreme sample database. This database is included with Crystal Reports.

Browse the Samples\En directory to find the Databases folder that contains the Xtreme database.

Comments

Formula comments are notes included 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. You can use comments to explain the purpose of a formula or explain the steps involved in writing it.

Comments begin with two forward slashes (//) and are followed by the text of the comment. Everything that follows the slashes on the same line is treated as being part of the comment:

```
//This formula returns the string "Hello"
//This is another comment
"Hello" //Comments can be added at the end of a line
//Comments can occur after the formula text
```

Fields

Many of the fields used in the construction of a report can be referred to in your formulas. For example, database, parameter, running total, SQL expression, summary, and group name fields can all be used in a formula. You can also refer to other formula fields in your formula.

The easiest way to insert a field into your report is to double-click a field’s name in the Report Fields tree. This ensures that the correct syntax for the field is used.

How fields appear in formulas

Database, parameter, formula, running total and SQL expression fields have their names surrounded by braces. Database field names are taken from the database. For example:

- database field: `{Employee.Last Name}`

Parameter, formula, running total, and SQL expression field names are specified when the fields are created.

- parameter fields also includes a question mark: `{?my parameter field}`
- formula fields include an at sign: `{@another formula }`
- running totals fields include a pound sign: `{#my running total}`
- SQL expression fields include a percent sign: `{%my SQL expression}`

Summary and group name fields look like function calls. However, they are really shorthand notation for a report field.

- sum summary field: `Sum({Orders.Order Amount}, {Orders.Ship Via})`
- group name field: `GroupName({Orders.Ship Via})`

Sample formulas using fields

The formula in this example uses the Xtreme database. To find out how many days it takes to ship the product from the date when the order was placed, you can just subtract the ship date database field from the order date database field:

```
//A formula that uses database fields
{Orders.Ship Date} - {Orders.Order Date}
```

To find the total dollar amount of a given product that was ordered, multiply its unit price by the quantity ordered:

```
{Orders Detail.Unit Price} * {Orders Detail.Quantity}
```

To calculate a sale price of 80 percent of the original unit price:

```
{Orders Detail.Unit Price} * 0.80
```

Expressions

An *expression* is any combination of keywords, operators, functions, and constant values that result in a value of a given type. For example:

```
//An expression that evaluates to the Number value 25
10 + 20 - 5

//An expression that evaluates to the String value
//"This is a string."
"This is a string."
```

A Crystal syntax formula consists of a sequence of expressions. The value of the final expression is the value returned by the formula and what gets printed. Each expression must be separated from the previous expression by a semicolon (;).

Multiple expressions

Typically, each expression takes one line, but you can continue an expression onto the next line if you need more space.

The formula below consists of five expressions. It returns the Number value 25 since that is the value of the last expression in the formula.

Example

```
//Expressions example
//The first expression. Its value is the Number
//value 30
10 + 20;

//The second expression. Its value is the String
//"Hello World". It takes up two lines.
"Hello " +
"World";

//The third expression. Its value is of Number type
{Orders Detail.Quantity} * 2 - 5;

//The fourth expression. Its value is of String type
If {Orders Detail.Quantity} > 1 Then
   "multiple units"
Else
   "one unit";

//The fifth and final expression. Its value is the
//Number value 25
20 + 5
```

Placing a semicolon after the last expression in the formula is also allowed, but optional. For example, the above formula could have ended with:

```
20 + 5;
```

Some of the sample formulas in the preceding section do not have semicolons. This is because they consist of a single expression, and a semicolon is optional after the last expression. Many formulas in Crystal syntax can be written as a single expression.

Notice that there is no semicolon after the “multiple units” String. In fact, if you put a semicolon there, the program will report an error. This is because a semicolon separates expressions, and the

```
Else
   "one unit";
```

is not a separate expression. It does not stand alone apart from the If. In fact, it is an integral part of the If expression since it describes the value that the If expression will return under certain circumstances.

Note: The example is not a practical example because the first four expressions in the formula did not have any effect on the last expression.

How earlier expressions affect later expressions

The fact that a Crystal syntax formula is a sequence of expressions whose result is the value of the final expression is the most important concept in understanding Crystal syntax. This expression-based syntax allows you to write very short formulas with a lot of functionality.

Example

```
//First expression. It declares the Number variable x
//and then returns the value of an uninitialized
//Number variable, which is 0.
NumberVar x;
//Second expression. It assigns the value of 30 to x,
//and returns 30.
x := 30
```

The above formula would give an error if the first expression were omitted. This is because the second expression refers to the Number variable x, and the program needs to have x declared before it understands expressions involving it.

In general, you use variables to get the earlier expressions in a formula to affect the final expression. See “Variables” on page 450 for more information.

Using the If expression

The *If* expression is one of the most widely used features of Crystal syntax. It also provides insight into the nature of expressions. Consider the earlier If expression as a separate formula. Notice that because this formula is a single expression, it does not need a semicolon:

```
If {Orders Detail.Quantity} > 1 Then
   "multiple units"
Else
   "one unit"
```

Suppose you wanted to modify this formula so that it either prints “multiple units” or the number 1.

```
//An erroneous formula
If {Orders Detail.Quantity} > 1 Then
   "multiple units"
Else
   1
```

This formula will result in an error. This is because the values in this expression are different types: “multiple units” is a String value and 1 is a Number value. Crystal Reports requires that the value of an expression always be of a single type.

Note: This example can be corrected by using the CStr function to convert the Number 1 to a String value. For example, the Number 1 is converted to the string “1” by calling CStr (1, 0).

```
//A correct formula
If {Orders Detail.Quantity} > 1 Then
   "multiple units"
Else
   CStr (1, 0) //Use 0 decimals
```

See “Control structures” on page 465 for more information on the If expression.

Assignment

The assignment operator is a colon followed by an equals sign (:=).

Example

```
//Assign the Number value of 10 to the variable x
x := 10;
//Assign the String value of "hello" to the
//variable named greeting
greeting := "hello";
```

The equality operator (=) is used to check when two values are equal. A common error is to use the equality operator when the assignment operator is actually intended. This can generate a mysterious error message or no error message at all since it is often syntactically correct to use the equality operator. For example:

```
greeting = "hello";
```

The above formula checks if the value held by the variable greeting is equal to the value “hello”. If it is, then the expression's value is *True*, and if is not then the expression's value is *False*. In any case, it is a perfectly correct Crystal syntax expression (assuming that the greeting is a String variable).

Simple data types

The simple data types in Crystal Reports are Number, Currency, String, Boolean, Date, Time, and DateTime.

Number

Enter numbers without any comma separators or currency symbols (generally you would want to have formatted numbers appearing as the result of a formula and not in the formula itself).

Example

```
10000
-20
1.23
```

Currency

Use the dollar sign ($) to create a Currency value.

Example

```
$10000
-$20
$1.23
```

You can also use the CCur function. The initial C in CCur stands for convert and it can be used to convert Number values to Currency values:

```
CCur (10000)
CCur (-20)
CCur (1.23)
```

String

Strings are used to hold text. The text must be placed between double quotation marks (") or apostrophes (') and cannot be split between lines. If you want to include double quotation marks in a string delimited by double quotation marks, use two consecutive double quotation marks. Similarly, if you want to include an apostrophe in a string delimited by apostrophes, use two consecutive apostrophes.

Example

```
"This is a string."
"123"
"The word ""hello"" is quoted."
'This is also a string.'
'123'
'Last Year''s Sales'
```

If you use double quotes for the left side of the string, you must use double quotes on the right side. Similarly for apostrophes. The following example is incorrect:

```
'Not a valid string."
```

You can extract individual elements or substrings from a string by specifying the character position or a range of character positions. Negative values are allowed; they specify the position starting from the end of the string.

```
"hello" [2] //Equal to "e"
"hello" [-5] //Equal to "h"
"604-555-1234" [1 to 3] //Equal to "604"
"abcdef" [-3 to -1] //Equal to "def"
```

You can also extract substrings from a string using the Left, Right and Mid functions.

Boolean

The valid Boolean values are:

```
True
False
```

Note: *Yes* can be used instead of True and *No* instead of False.

Date, Time, and DateTime

The DateTime type can hold date-times, dates only, or times only and thus is rather versatile. The Date type holds dates only and the Time type holds times only. The Date and Time types are more efficient than the DateTime type, and so can be used in situations where the added functionality and flexibility of the DateTime type is not needed.

You can create DateTime values directly using the date-time literal construction, which is formed by typing in the date-time between two pound (#) signs. Many different formats are supported.

Note: Date-time literals cannot be split between lines.

Examples

```
#8/6/1976 1:20 am#
#August 6, 1976#
#6 Aug 1976 13:20:19#
#6 Aug 1976 1:30:15 pm#
#8/6/1976#
#10:20 am#
```

Even though #10:20 am# looks like it could have the Time type and #8/6/1976# looks like it could have the Date type, they do not. They both have the DateTime type, as do all date-time literals. For example, you can think of #10:20 am# as a DateTime value with a null date part. To convert it to the Time type use CTime (#10:20 am#).

Instead of using date-time literals, you can use CDateTime to convert a String to a DateTime. For example,

```
CDateTime ("8/6/1976 1:20 am")
CDateTime ("10:20 am")
```

However, there is one key difference between using date-time literals and the above usage of CDateTime. Date-time literals always use U.S. English date formats rather than settings from the locale of the particular computer on which Crystal Reports is running. Thus, the date-time literal examples above would work on all computers. On the other hand, on a French system, you could use constructions like:

```
CDateTime ("22 aout 1997") //Same as #Aug 22, 1997#
```

Date values can be constructed with CDate and Time values with CTime:

```
CDate ("Aug 6, 1969")
CDate (1969, 8, 6) //Specify the year, month, day
//Converts the DateTime argument to a Date
CDate (#Aug 6, 1969#)
```

```
CTime ("10:30 am")
CTime (10, 30, 0) //Specify the hour, minute, second
CTime (#10:30 am#)
```

Range data types

Ranges are designed to handle a spectrum of values. Range types are available for all the simple types except for Boolean. That is: Number Range, Currency Range, String Range, Date Range, Time Range and DateTime Range. You can generate ranges using the To, _To, To_, _To_, UpTo, UpTo_, UpFrom, and UpFrom_ keywords. In general, To is used for ranges with 2 endpoints, and UpTo and UpFrom are used for open ended ranges (only one endpoint). The underscores are used to indicate whether or not the endpoints are in the range.

Examples of Number Range values

The range of numbers from 2 to 5 including both 2 and 5

```
2 To 5
```

The range of numbers from 2 to 5, not including 2 but including 5

```
2 _To 5
```

All numbers less than or equal to 5

```
UpTo 5
```

All number less than 5

```
UpTo_ 5
```

Examples of DateTime Range values:

```
#Jan 5, 1999# To #Dec 12, 2000#
UpFrom #Jan 1, 2000#
```

Using ranges in formulas

There are twenty-seven functions in Crystal Reports that specify date ranges. For example, the function LastFullMonth specifies a range of date values that includes all dates from the first to last day of the previous month. So if today's date is September 15, 1999 then LastFullMonth is the same as the range value CDate (#Aug 1, 1999#) To CDate (#Aug 31, 1999#).

Ranges are often used with If or Select expressions. The following example computes student letter grades based on their test scores. Scores greater than or equal to 90 receive an “A”, scores from 80 to 90, not including 90, receive a “B” and so on.

```
//Compute student letter grades
Select {Student.Test Scores}
   Case UpFrom 90 :
      "A"
   Case 80 To_ 90 :
      "B"
   Case 70 To_ 80 :
      "C"
   Case 60 To_ 70 :
      "D"
   Default :
      "F";
```

The above example uses the Select expression which is discussed in more detail in the control structures section. You can check if a value is in a range by using the In operator. For example:

```
5 In 2 To 10; //True
5 In 2 To_ 5; //False
5 In 2 To 5; //True
```

The Maximum and Minimum functions can be used to find the endpoints of a range:

```
Maximum (2 To 10) //Returns 10
```

Array data types

Arrays in Crystal Reports are ordered lists of values that are all of the same type. These values are known as the array's *elements*. The elements of an array can be any simple type or range type. Arrays can be created using square brackets ([ ]).

Examples

An array of 3 Number values. The first element is 10, the second is 5 and the third is 20.

```
[10, 5, 20]
```

An array of 7 String values:

```
["Sun", "Mon", "Tue", "Wed", "Th", "Fri", "Sat"]
```

An array of 2 DateTime Range values:

```
[#Jan 1, 1998# To #Jan 31, 1998#,
 #Feb 1, 1999# To #Feb 28, 1999#]
```

You can extract individual elements out of an array using square brackets containing the index of the element you want. This is called *subscripting* the array:

```
[10, 5, 20] [2] //Equal to 5
```

Number ranges can also be used to subscript arrays. The result is another array. For example:

```
[10, 5, 20] [2 To 3] //Equal to [5, 20]
```

Arrays are most useful when used with variables and so will be discussed in more detail in the "Variables" section. Using variables, you can change the individual elements of an array and resize the array to accommodate more elements. For example, you can accumulate database field values into a global array variable in a detail level formula, and then use a formula in a group footer to perform a calculation based on those values. This enables you to perform a wide variety of customized summary operations.

Variables

This section describes the key components of variables and shows you how to create variables and assign values to them.

Variables overview

A variable represents a specific data item, or value, and acts as a placeholder for that value. When a formula encounters a variable, the formula searches for the value of the variable and uses it in the formula. 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. Because of this flexibility, it is necessary for you to declare variables before you use them so that Crystal Reports is aware of them and understands how you intend to use them.

Example

If you wanted to report on customers by area code, you could create a variable that extracts the area code from a customer fax number. The following is an example of a variable called *areaCode*:

```
Local StringVar areaCode;
areaCode := {Customer.Fax} [1 To 3];
```

The first line of the variable example is the variable declaration; it gives the variable a name and type. The database field {Customer.Fax} is a String field and [1 To 3] extracts the first 3 characters from its current value. The variable areaCode is then assigned this value.

Variable declarations

Before using a variable in a formula, you must declare it.

A variable can hold values of a given type. The allowed types are the seven simple types (Number, Currency, String, Boolean, Date, Time and DateTime), the six range types (Number Range, Currency Range, String Range, Date Range, Time Range and DateTime Range) and variables that hold arrays of the previously mentioned types. This gives a total of 26 different types that a variable can have.

When you declare a variable, you also specify its name. A variable cannot have the same name as any function, operator or other keyword that is valid for Crystal syntax. For example, your variable cannot be named Sin, Mod or If because Sin is a built in function, Mod is a built in operator and If is a built in keyword. When typing formulas in the formula editor, the names of the built in functions, operators and other keywords are highlighted in a different color and so it is easy to check if the variable name conflicts.

Once a variable is declared, it can be used in the formula. For example, you might want to assign it an initial value:

```
Local NumberVar x; //Declare x to be a Number variable
x := 10; //Assign the value of 10 to x
```

Note: The keyword for declaring the Number variable has a *Var* at the end. This is true for all the variable types in Crystal syntax.

A variable can only hold values of one type. For example, if a variable holds a Number value, you cannot later use it to hold a String.

Example

```
Local StringVar y;
y := "hello";

//OK- the Length function expects a String argument
Length (y);

//Error- y can only hold String values
y := #Jan 5, 1993#;

//Error- y can only hold String values
y := ["a", "bb", "ccc"];

//Error- the Sin function expects a Number argument
Sin (y);
```

You can combine declaring a variable and assigning it a value in a single expression. For example:

```
Local NumberVar x := 10 + 20;
Local StringVar y := "Hello" + " " + "World";
Local DateVar z := CDate (#Sept 20, 1999#);
Local NumberVar Range gradeA := 90 To 100;
```

This is a good practice because it is more efficient and helps prevent the common mistake of having incorrectly initialized variables.

Here are some more examples of declaring and initializing range variables:

```
Local NumberVar Range gradeA;
Local DateVar Range quarter;
gradeA := 90 To 100;
quarter := CDate (1999, 10, 1) To CDate (1999, 12, 31);
```

Variable Scope

Variable scopes are used to define the degree to which variables in one formula are made available to other formulas. There are three levels of scope in Crystal Reports: local, global and shared. Every variable has a scope, and this scope is specified when the variable is declared.

Local Variables

Variables with local scope, also known as local variables, are declared using the Local keyword followed by the type name (with the Var suffix) followed by the variable name as in the above examples.

Local variables are restricted to a single formula and a single evaluation of that formula. This means that you cannot access the value of a local variable in one formula from a different formula.

Example

```
//Formula A
Local NumberVar x;
x := 10;
```

```
//Formula B
EvaluateAfter ({@Formula A})
Local NumberVar x;
x := x + 1;
```

The function call EvaluateAfter ({@Formula A}) ensures that Formula B will be evaluated after Formula A is evaluated. Formula A returns a value of 10 and Formula B returns a value of 1. Formula B does not have access to Formula A's x and thus cannot use the value of 10 and add 1; instead, it uses the default value for the uninitialized local variable x found in Formula B, which is 0, and adds 1 to it to get 1.

You can also create local variables with the same name but different types in different formulas. For example, the type declarations in formulas A and B do not conflict with:

```
//Formula C
Local StringVar x := "hello";
```

Local variables are the most efficient of the three scopes. They also do not interfere with one another in different formulas. For these reasons, it is best to declare variables to be local whenever possible.

Global variables

Global variables use the same memory block to store a value throughout the main report. This value is then available to all formulas that declare the variable, except for those in subreports. You declare a global variable as in the following example:

```
Global StringVar y;
```

You can also omit the Global keyword which creates a Global variable by default:

```
StringVar y; //Same as: Global StringVar y;
```

However, even though global variables are easy to declare, it is recommended that you use them only when local variables do not suffice.

Since global variables share their values throughout the main report, you cannot declare a global variable in one formula with one type and then declare a global variable with the same name in a different formula with a different type.

Example

```
//Formula A
Global DateVar z;
z := CDate (1999, 9, 18)
```

```
//Formula B
NumberVar z;
z := 20
```

In this case, if you enter and save Formula A first, Crystal Reports will return an error when you check or try to save Formula B. This is because the declaration of the Global variable z as a Number conflicts with its earlier declaration in Formula A as a Date.

Using Global variables

Global variables are often used to perform complex calculations where the results of a formula depend upon the grouping and page layout of the actual printed report. This is accomplished by creating several formulas, placing them in different sections of the report, and having the different formulas interact via global variables.

Example

```
//Formula C
Global NumberVar x;
x := 10;
```

```
//Formula D
//Call the function WhileReadingRecords
WhileReadingRecords;
Global NumberVar x;
x := x + 1
```

If Formula C is placed in the Report Header and then Formula D is placed in a detail section, Formula C will be evaluated before Formula D. Formula C will be evaluated once and then Formula D will be evaluated for each record appearing in the detail section. Formula C returns 10. For the first detail record, Formula D returns 11. This is because the value 10 of x is retained from when it was set by Formula C. Formula D then adds 1 to this value, setting x to 11 and then returns 11. For the second detail record, formula D return 12, adding 1 to the previously retained value of x which was 11. This process continues for the remaining detail records.

The call to WhileReadingRecords tells Crystal Reports to re-evaluate Formula D as it reads in each record of the report. Otherwise, since the formula does not contain any database fields, the program evaluates it only once before reading the records from the database. The formula will then return the value 11 instead of 11, 12, 13, ... as the successive records are processed.

If the expression x := x + 1 is replaced by x := x + {Orders Detail.Quantity}, you create the effect of a running total based on {Orders Detail.Quantity}, although it is one starting at 10 rather than 0 because of Formula C. In this case, you can omit the call to WhileReadingRecords, since it will automatically occur because the formula contains a database field.

Shared variables

Shared variables use the same memory block to store the value of a variable throughout the main report and all of its subreports. Thus shared variables are even more general than global variables. To use a shared variable, declare it in a formula in the main report as in the following example:

```
Shared NumberVar x := 1000;
```

and declare it in a formula in the subreport as in the following example:

```
Shared NumberVar x;
```

In order to use shared variables, the variable must be declared and assigned a value before it can be passed between the main report and the subreport.

Declaring array variables

You can declare array variables by following the type name with the keyword Array.

Example

```
//Declare x to be a Global variable of
//Number Array type
Global NumberVar Array x := [10 , 20, 30];

//cost is a Global variable of Currency Array type
//It is automatically Global since the scope specifier
//(one of Local, Global or Shared) is omitted.
CurrencyVar Array cost := [$19.95, $79.50, $110.00,
                           $44.79, $223.99];
```

```
//payDays is a Global variable of Date Array type
Global DateVar Array payDays := [CDate(1999, 5, 15),
CDate(1999, 5, 31)];
```

```
//y is a Shared variable of String Range Array type
Shared StringVar Range Array y := ["A" To "C",
                                   "H" To "J"];
```

```
//days is a Local variable of String Array type
Local StringVar Array days;
days := ["Sun", "Mon", "Tue", "Wed", "Th",
         "Fri", "Sat"];
```

Using array variables

You can assign values to elements of an array and also use the values of the elements for other computations.

Example

```
StringVar Array x := ["hello", "bye", "again"];
x [2] := "once"; //Now x is ["hello", "once", "again"]
//The expression below would cause an error if not
//commented out since the array has size 3
//x [4] := "zap";
```

```
//The formula returns the String "HELLO"
UpperCase (x [1])
```

The Redim and Redim Preserve keywords can be used to resize an array if you want to add extra information to it. Redim erases the previous contents of the array first before resizing it whereas Redim Preserve preserves the previous contents.

```
Local NumberVar Array x;
Redim x [2]; //Now x is [0, 0]
x [2] := 20; //Now x is [0, 20]
Redim x [3]; //Now x is [0, 0, 0]
x [3] := 30; //Now x is [0, 0, 30]
Redim Preserve x [4]; //Now x is [0, 0, 30, 0]
"finished"
```

```
Local StringVar Array a;
Redim a [2];
//Assign a value to the first element of the array a
a[1] := "good";
a[2] := "bye";
//The & operator can be used to concatenate strings
a[1] & a[2] //The formula returns the String "goodbye"
```

Using Arrays with For loops

Arrays are commonly used with For loops. The following example creates and then uses the Array [10, 20, 30, ..., 100] using a For loop. See "For loops" on page 470 for more details.

```
Local NumberVar Array b;
Redim b[10];
Local NumberVar i;
For i := 1 To 10 Do
(
   b[i] := 10 * i
);
b [2] //The formula returns the Number 20
```

Default values for the simple types

An uninitialized variable will have the default value for its type. In general, it is not a good programming practice to rely on the default values of types. For example, initialize all local variables in your formula, initialize all global variables in a formula placed in the Report Header and initialize all shared variables in a formula placed in the Report Header of the main report.

When an array is resized using the Redim keyword, the entries are filled with default values for the type. It is useful to know about default values when using If and Select expressions.

Default values

Number

```
0
```

Currency

```
$0
```

String

```
"" //The empty string
```

Date

```
Date (0, 0, 0) //The null Date value
```

Time

The null Time value. Value held by an uninitialized Time variable.

DateTime

The null DateTime value. Value held by an uninitialized DateTime variable.

Note: It is not recommended that your formulas rely on the values of uninitialized range or array variables.

Automatic type conversions

Generally in Crystal Reports, values of one type cannot be used where values of another type are expected without explicitly supplying a type conversion function. For example:

```
Local StringVar postalCode;
//Error- assigning a Number value to a String
postalCode := 10025;
//OK - use the type conversion function CStr
//to create "10025"
postalCode := CStr (10025, 0);
```

However, there are a few conversions that are made automatically:

- Number to Currency
- Date to DateTime
- Simple type to Range value of the same underlying simple type

For example, the following assignments are correct:

```
Local CurrencyVar cost;
//Same as: cost := $10
cost := 10;
```

```
Local DateTimeVar orderDate;
//Same as: orderDate := CDateTime (1999, 9, 23, 0, 0, 0)
orderDate := CDate (1999, 9, 23);
```

```
Local NumberVar Range aRange;
//Same as: aRange := 20 To 20
aRange := 20;
```

```
Local NumberVar Range Array aRangeArray;
//Same as : aRangeArray := [10 To 10, 20 To 25, 2 To 2]
aRangeArray := [10, 20 To 25, 2];
```

Note: The opposite conversions are not allowed. For example:

```
Local NumberVar num;
num := 5 + $10; //Error
//OK- convert to Number type using the CDbl function
num := CDbl (5 + $10) //Could also use ToNumber
```

5 is converted to $5 and added to $10 to make $15. However, this Currency value cannot be automatically assigned to the Number variable num since automatic conversions from Currency to Number are not allowed. Similarly, functions accepting a Currency argument can be supplied a Number argument instead, and the Number argument will be converted to a Currency, whereas functions accepting a Number argument cannot be supplied a Currency argument without first explicitly converting the Currency to a Number using CDbl.

Functions

When using a function in a formula, type the name of the function and supply the arguments required. For example, the Length function requires a String argument and computes the length of the string.

```
Local StringVar x := "hello";
Length (x) //The formula returns the Number 5
```

Supplying arguments of the incorrect type required by the function produces an error. For example, calling Length (3) would produce an error since Length does not accept a Number argument. Functions sometimes can accept different numbers of arguments or types of arguments. For example, the CDate function which could accept a single String argument to form a Date value or 3 Number values holding the year, month and day respectively and form a Date value from them. See "Date, Time, and DateTime" on page 447.

Example with the Mid function

```
Local StringVar x := "hello";
Local StringVar y;
//Start at position 2, go to the end of the string
y := Mid (x, 2); //y is now "ello"
//Start at position 2, extract 1 character
y := Mid (x, 2, 1) //y is now "e"
```

These classes of functions are: Math, Summary, Financial, String, Date/Time, Date Range, Array, Type Conversion, Programming Shortcuts, Evaluation Time, Print State, Document Properties and Additional Functions. There are also some functions specific to conditional formatting formulas.

Tip: Descriptions of the functions supported by Crystal Reports can be found in the online help.

Non reporting-specific functions

The Math, Financial, String, Date/Time, Type Conversion, and Programming Shortcuts groups consist mainly of functions that are not specific to reporting, but might be found in any full featured programming environment. Many of the functions are similar in functionality to Visual Basic functions of the same or similar name.

Note: Some functions that are supported by Crystal syntax are not listed in the Crystal syntax Functions tree. This is because they are equivalent to Crystal syntax functions already listed in the tree.

For example, the Length function is the traditional Crystal syntax function for finding the length of a string. Crystal syntax also supports Len as a synonym. Len is the Visual Basic and Basic syntax function for this and its inclusion is for the convenience of Visual Basic and Basic syntax users who want to write or modify Crystal syntax formulas.

Summary functions

The Summary function group provides functions for creating summary fields such as:

```
Sum({Orders.Order Amount}, {Orders.Ship Via})
```

Summary fields are normally created using the Insert Summary or Insert Grand Total dialogs. They then appear in the Available Fields tree, and can be used in a formula by double clicking there. However, they do not need to be created in this way. You can create a summary field exclusively for use by your formula by filling in the arguments to one of the functions in the Summary functions section appropriately. However, any groups in the report that the summary field refers to must already exist in the report. See the online help on summary fields for details.

Date Ranges

This category of functions was discussed in the section "Range data types" on page 448. One additional comment is that the Date ranges produced by these functions depend on the current date. For example, if today's date is September 18, 1999, then LastFullMonth is the Date Range value:

```
CDate(#Aug 1, 1999#) To CDate(#Aug 31, 1999#)
```

This functionality is often useful, but what if you want to determine a date range based on a database field such as {Orders.Order Date}? The Date/Time functions can be used instead.

For example:

```
Local DateVar d := CDate ({Orders.Order Date});
Local DateVar Range dr;
dr := DateSerial (Year(d), Month(d) - 1, 1) To
      DateSerial (Year(d), Month(d), 1 - 1);
//At this point dr is the Date Range value holding
//the last full month before {Orders.Order Date}
```

The DateSerial function makes this easy because you don't have to worry about special cases. It never lets you create an invalid date. For example, DateSerial (1999, 1 - 1, 1) is December 1, 1998.

Note: In the above example, {Orders.Order Date} is actually a DateTime field and so the CDate function is used to convert it to a date by truncating the time part.

Array functions

The array functions compute summaries of an array's elements. For example, the Sum function when applied to an array returns the sum of the elements of the array. For example, the following formula returns 100:

```
Sum ([10, 20, 30, 40])
```

Evaluation Time functions

These are the reporting specific functions BeforeReadingRecords, WhileReadingRecords, WhilePrintingRecords and EvaluateAfter. You can use these functions to guide Crystal Reports as to when your formula should be evaluated.

Should the formula be evaluated before retrieving the records from the database, while reading the records from the database but before the records have been grouped, sorted and summarized, or while printing the report, when the records are grouped, sorted and summarized? In general, Crystal Reports sets an appropriate evaluation time for your formula, based on how much information the formula needs. For example, if a formula uses a database field, then it cannot be evaluated before the records are read from the database. However, you sometimes need to force a later evaluation time than normal to get the desired effect. See "Global variables" on page 453 for an example.

Print State functions

These are once again reporting specific functions. For example, the notation {Orders.Order Date} is used to refers to the value of the field in the current record whereas Previous ({Orders.Order Date}) refers to the value in the immediately preceding record and Next ({Orders.Order Date}) in the next record. IsNull ({Orders.Order Date}) checks if the field's value is null.

Other examples are PageNumber and TotalPageCount that you can use to access pagination information about your report.

Document Properties functions

These are specific functions, such as PrintDate and ReportTitle, that refer to the report document as a whole.

Additional functions

These are functions that are in User Function Libraries or UFLs. A UFL is a separate dynamic link library or Automation server that you create and that Crystal Reports can use to add your own customized functions to the formula language. Writing a UFL is more involved than writing a formula using Basic syntax or Crystal syntax. See Crystal Reports Developer's Help for details.

Note: Using UFLs makes your reports less portable because you must distribute your UFL along with the report.

Conditional formatting functions

When writing a conditional formatting formula, certain additional functions appear at the top of the Functions tree to help you with this. For example, you can format the {Customer.Last Year’s Sales} field to print sales of more than $100,000 in green, sales of less than $15,000 in red, and all other sales in black.

Example

```
//Conditional formatting example 1
If {Customer.Last Year's Sales} > 100000 Then
   crGreen
Else If {Customer.Last Year's Sales} < 15000 Then
   crRed
Else
   crBlack
```

Since this is a font color formatting function, the list of Color Constants appears in the Functions Tree. This example uses three: crGreen, crRed and crBlack. You could have used the actual numeric values of the color constants instead. For example, crRed is 255 and crGreen is 32768. However, the formula is more understandable using the color constants. All constant functions in Crystal syntax can have the “cr” prefix.

Crystal syntax still supports constant functions from previous versions without the “cr” prefix. For example, you can use “Red” instead of “crRed”. However, using the “cr” prefix organizes constant functions and is recommended.

Note: Some formatting attributes do not use constant functions. For example, if you wanted to not print {Customer.Last Year's Sales} values if the sales were less than $50,000, you could write the following conditional formatting formula for the suppress attribute:

```
//Conditional formatting example 2
If {Customer.Last Year's Sales} < 50000 Then
   True //suppress the value
Else
   False //do not suppress the value
```

Or more simply:

```
//Conditional formatting example 3 -
//equivalent to example 2
{Customer.Last Year's Sales} < 50000
```

If the last year's sales are less than $50,000, then the expression

```
{Customer.Last Year's Sales} < 50000
```

is True, and so the formula returns True. On the other hand, if the last year's sales are greater than or equal to $50,000, then

```
{Customer.Last Year's Sales} < 50000
```

is False and so the formula returns False.

General purpose conditional formatting functions

There are three general purpose conditional formatting functions:

- CurrentFieldValue
- DefaultAttribute
- GridRowColumnValue.

These functions are displayed at the top of the Functions tree whenever appropriate. DefaultAttribute can be used for any formatting formula, CurrentFieldValue for any formatting formula where you are formatting a field value, and GridRowColumnValue for any formatting formula where you are formatting a field value in a Cross-Tab or OLAP grid.

In particular, CurrentFieldValue allows you to conditionally format the cells of a Cross-Tab or OLAP grid based on their value while GridRowColumnValue lets you conditionally format the cells of a Cross-Tab or OLAP grid based on the values of the row or column headings. These two functions are essential in this situation since there is no other way in the formula language to refer to the values of these fields. For example, if you wanted Cross-Tab cells to be suppressed if the values are less than 50,000:

```
//Conditional formatting example 4
CurrentFieldValue < 50000
```

Tip: Descriptions of the conditional formatting functions supported by Crystal Reports can be found in the online help.

Operators

Arithmetic operators

The arithmetic operators are addition (+), subtraction (-), multiplication (*), division (/), integer division (\), percent (%), modulus (Mod), negation (-) and exponentiation (^). See the online help for details on what each operator does. Arithmetic operators are used to combine numbers, numeric variables, numeric fields and numeric functions to get another number.

Examples

```
//Outstanding preferred stock as a percent of
//common stock
{Financials.Preferred Stock} %
{Financials.Common Stock};
```

```
//The square root of 9, Sqr(9) is 3
//The formula returns 17
7 + 2 * 3 - 2 + Sqr(6 + 3) * Length("up");
```

Tip: Descriptions of operators supported by Crystal Reports can be found in the online help.

Order of precedence

When you create arithmetic expressions that involve several operators, the order that the program evaluates the various elements of the expression becomes important. In general, the program evaluates expressions from left to right. However, it also follows the rules of precedence from basic math.

Example

Multiplication and division are performed first from left to right and then addition and subtraction are performed.

For example, 5 + 10 * 3 = 5 + 30 = 35. You can change this order of precedence by using parentheses. For example, (5 + 10) * 3 = 15 * 3 = 45. If you are unsure of the order of precedence, it is a good idea to clarify your intentions with parentheses.

List of arithmetic operators, from highest precedence to lowest

- Exponentiation (^)
- Negation (-)
- Multiplication, division, and percent (*, /, %)
- Integer Division (\)
- Modulus (Mod)
- Addition and subtraction (+, -).

Comparison operators

The comparison operators are equal (=), not equal (<>), less than (<), less than or equal (<=), greater than (>) and greater than or equal (>=).

Comparison operators are usually used to compare operands for a condition in a control structure such as an If expression. Comparison operators as a group all have lower precedence than the arithmetic operators. Thus, expressions like 2 + 3 < 2 * 9 are the same as (2 + 3) < (2*9).

Boolean operators

The Boolean operators are, in order of precedence from greatest to lowest: Not, And, Or, Xor, Eqv and Imp. Boolean operators are typically used with comparison operators to generate conditions for control structures. Boolean operators as a group have lower precedence than the comparison operators. Thus for example, the expression 2 < 3 And 4 >= -1 is the same as (2 < 3) And (4 >= -1).

Null fields and how to use IsNull

The {Product.Color} field contains both basic colors such as “red” and “black” and fancy two word colors such as “steel satin” and “jewel green”. Suppose that you want to write a formula that writes out “basic” for the basic colors and “fancy” for the others.

```
If InStr({Product.Color}, " ") = 0 Then
   "basic"
Else
   "fancy"
```

The function call to InStr searches the {Product.Color} string for a space. If it finds a space, it returns the position of the space, otherwise it returns 0. Since basic colors are only one word with no spaces, InStr will return 0 for them.

Dealing with null values

For some products, such as the Guardian Chain Lock, a color value was not recorded and so the {Product.Color} field has a null value in the database for that record. In general, when Crystal Reports encounters a null valued field in a formula, it immediately stops evaluating the formula and produces no value. Thus, the Guardian Chain Lock record does not have any word printed beside it. If you want to handle null field values in your formula, you must explicitly do so using one of the special functions designed for handling them: IsNull, PreviousIsNull or NextIsNull.

Here is how to fix up the previous example using IsNull:

```
If IsNull({Product.Color}) Or
   InStr({Product.Color}, " ") = 0 Then
   "basic"
Else
   "fancy"
```

How this relates to operators, is that when Crystal Reports evaluates the condition:

```
IsNull({Product.Color}) Or
InStr({Product.Color}, " ") = 0
```

It first evaluates IsNull ({Product.Color}), and when it determines that this is True, it knows that the whole condition is True, and thus does not need to check whether

```
InStr({Product.Color}, " ") = 0
```

In other words, Crystal Reports will stop evaluating a Boolean expression when it can deduce the results of the whole expression.

In the following example, the formula guards against attempting to divide by zero in the case that denom is 0:

```
Local NumberVar num;
Local NumberVar denom;
...
If denom <> 0 And num / denom > 5 Then
...
```

Control structures

Formulas without control structures execute each expression in the formula exactly once when the formula is evaluated. The expressions are executed in a sequential fashion, from the first expression in the formula to the last. Control structures enable you to vary this rigid sequence. Depending upon which control structure you choose, you can skip over some of the expressions or repeatedly evaluate some expressions depending on if certain conditions hold. Control structures are the primary means of expressing business logic and typical report formulas make extensive use of them.

If expressions

The *If* expression is one of the most useful control structures. It allows you to evaluate an expression if a condition is true and evaluate a different expression otherwise.

Note: When formatting with conditional formulas, always include the Else keyword; otherwise, values that don't meet the If condition may not retain their original format. To prevent this, use the DefaultAttribute function (`If...Else DefaultAttribute`).

Example

A company plans to pay a bonus of 4 percent to its employees except for those who work in Sales who will receive 6 percent. The following formula using an If expression would accomplish this:

```
//If example 1
If {Employee.Dept} = "Sales" Then
   {Employee.Salary} * 0.06
Else
   {Employee.Salary} * 0.04
```

In this example, if the condition `{Employee.Dept} = "Sales"` evaluates as true, then the

```
{Employee.Salary} * 0.06
```

expression is processed. Otherwise the expression following the *Else*, namely the

```
{Employee.Salary} * 0.04
```

is processed.

Suppose another company wants to give employees a 4% bonus, but with a minimum bonus of $1,000. The following example shows how. Notice that the *Else* clause is not included; it is optional, and not needed in this case.

```
//If example 2
Local CurrencyVar bonus := {Employee.Salary} * 0.04;
If bonus < 1000 Then
   bonus := 1000;
//The final expression is just the variable 'bonus'.
//This returns the value of the variable and is the
//result of the formula
bonus
```

Another way of accomplishing example 2 is to use an Else clause:

```
//If example 3
Local CurrencyVar bonus := {Employee.Salary} * 0.04;
If bonus < 1000 Then
   1000
Else
   bonus
```

Now suppose that the previous company also wants a maximum bonus of $5,000. You now need to use an Else If clause. The following example has only one Else If clause, but you can add as many as you need. Note, however, that there is a maximum of one Else clause per If expression. The Else clause is executed if none of the If or Else If conditions are true.

```
//If example 4
Local CurrencyVar bonus := {Employee.Salary} * 0.04;
If bonus < 1000 Then
   1000
Else If bonus > 5000 Then
   5000
Else
   bonus;
```

If example

Suppose that a company wants to compute an estimate of the amount of tax an employee needs to pay and write a suitable message. Income below $8,000 is not taxed, income from $8,000 to $20,000 is taxed at 20% income from $20,000 to $35,000 is taxed at 29% and income above $35,000 is taxed at 40%.

```
//If example 5
Local CurrencyVar tax := 0;
Local CurrencyVar income := {Employee.Salary};
Local StringVar message := "";

If income < 8000 Then
(
   message := "no";
   tax := 0
)
Else If income >= 8000 And income < 20000 Then
(
   message := "lowest";
   tax := (income - 8000)*0.20
)
Else If income >= 20000 And income < 35000 Then
(
   message := "middle";
   tax := (20000 - 8000)*0.20 + (income - 20000)*0.29
)
Else
(
   message := "highest";
   tax := (20000 - 8000)*0.20 + (35000 - 20000)*0.29 +
          (income - 35000)*0.40
);
//Use 2 decimal places and the comma as a
//thousands separator
Local StringVar taxStr := CStr (tax, 2, ",");
"You are in the " & message & " tax bracket. " &
"Your estimated tax is " & taxStr & "."
```

Note: The use of variables is to simplify the logic of the computation. Also, there are 2 expressions that are executed when one of the conditions are met; one assigns the tax variable, and the other assigns the message variable. It is often useful to have multiple expressions executed as a result of a condition.

More details on If expressions

The *If* expression is an expression. In other words it evaluates to a value of a given type. If there is no Else clause, and the condition is not true, then the value is the default value for the type. For example:

```
If Length ({Employee.First Name}) < 5 Then
   "short"
```

The above If expression returns a String value. The string value is “short” if the Employee's first name has fewer than 5 letters and the empty String "" otherwise.

Consider the formula:

```
If Year({Orders.Order Date}) >= 1995 Then
   {Orders.Order Date}
```

For order dates before 1995, the above If expression returns the null DateTime value. It is a DateTime value rather than a Date value since {Orders.Order Date} is a DateTime database field. The null DateTime value is not printed by Crystal Reports so if the above formula is placed in a report, the formula field would be blank for order dates before 1995. Null Time values and null Date values behave similarly.

Here is an example that illustrates the use of parentheses to have more than one expression executed as the outcome of an If condition. A company charges a 5 percent fee for orders shipped within 3 days and a 2 percent fee otherwise. It wants to print messages such as “Rush shipping is $100.00” or “Regular shipping is $20.00” as appropriate.

```
Local StringVar message;
Local CurrencyVar ship;
If {Orders.Ship Date} - {Orders.Order Date} <= 3 Then
(
   message := "Rush";
   //A semicolon at the end of the next line
   //is optional
   ship := {Orders.Order Amount} * 0.05
) //A semicolon cannot be placed here
```

```
Else
(
   message := "Regular";
   ship := {Orders.Order Amount} * 0.02;
);
//The preceding semicolon is required to separate the
//If expression from the final expression below
message & " shipping is " & CStr (ship)
```

When expressions are grouped together with parentheses, the whole group is considered as a single expression, and its value and type are the value and type of the final expression inside the parentheses.

```
//The parentheses group expression as a whole has
//Currency type
(
   //The first expression in the parentheses has
   //String type
   message := "Rush";
   //The second and final expression in parentheses
   //has Currency type
   ship := {Orders.Order Amount} * 0.05;
)
```

Thus, for example, the following formula gives an error. The reason is that the Then part of the If expression returns a Currency value while the Else part returns a String value. This is not allowed, since the If expression is an expression and so must always return a value of a single type.

```
//An erroneous formula
Local StringVar message;
Local CurrencyVar ship;

If {Orders.Ship Date} - {Orders.Order Date} <= 3 Then
(
   message := "Rush";
   ship := {Orders.Order Amount} * 0.05
)
Else
(
   //The following 2 lines were interchanged
   ship := {Orders.Order Amount} * 0.02;
   message := "Regular";
);
message & " shipping is " & CStr (ship)
```

One way of fixing up the erroneous formula without being careful about expression order is just to make the If expression return a constant value of the same type in every branch. For example, the If expression now returns the Number value 0:

```
//Repaired the erroneous formula
Local StringVar message;
Local CurrencyVar ship;
```

```
If {Orders.Ship Date} - {Orders.Order Date} <= 3 Then
(
   message := "Rush";
   ship := {Orders.Order Amount} * 0.05;
   0
)
Else
(
   ship := {Orders.Order Amount} * 0.02;
   message := "Regular";
   0
);
message & " shipping is " & CStr (ship)
```

Select expressions

The Select expression is similar to an If expression. Sometimes however, you can write clearer and less repetitive formulas using the Select expression. For example, to evaluate the {Customer.Fax} field to determine if the area code is for Washington state (206, 360, 509) or British Columbia, Canada (604, 250):

```
//Select example 1
Select {Customer.Fax}[1 To 3]
   Case "604", "250" :
      "BC"
   Case "206", "509", "360" :
      "WA"
   Default :
      "";
```

The expression right after the Select keyword is called the *Select condition*. In the above example it is {Customer.Fax}[1 To 3]. The Select expression tries to find the first Case that matches the Select condition, and then executes the expression following the colon for that Case. The Default case is matched if none of the preceding cases match the Select condition. Notice that there is also a colon after the Default.

```
//Same effect as Select example 1
Local StringVar areaCode := {Customer.Fax}[1 To 3];
If areaCode In ["604", "250"] Then
   "BC"
Else If areaCode In ["206", "509", "360"] Then
   "WA"
Else
   "";
```

Example

This formula groups the number of Oscar nominations a movie received into low, medium, high or extreme categories and in the process, shows some of the possibilities for the expression lists following the Case labels:

```
//Select example 2
Select {movie.NOM}
   Case 1,2,3, Is < 1 :
   (
      //Can have expression lists by using
      //parentheses
      10 + 20;
      "low"
   )
   Case 4 To 6, 7, 8, 9 :
      "medium"
   Case 10 :
      "high"
   Default :
      "extreme"
```

The Default clause of the Select expression is optional. If the Default clause is missing and none of the cases are matched, then the Select expression returns the default value for its expression type. For example, if in the above example the Default clause were omitted and {movie.NOM} = 11, it would return the empty string "". The Select expression is an expression, and similar comments as in the *More details on If expressions* section apply to it as well.

For loops

For loops enable you to evaluate a sequence of expressions multiple numbers of times. This is unlike the If and Select expressions where the program passes through each expression at most once during the formula's evaluation. For loops are best when you know the number of times that the expressions needs to be evaluated in advance.

The syntax of the For loop through examples

Example 1

Suppose you want to reverse the {Customer.Customer Name} string. For example, “City Cyclists” becomes “stsilcyC ytiC”.

```
//Reverse a string version 1
Local StringVar str := "";
Local NumberVar strLen :=
   Length ({Customer.Customer Name});
Local NumberVar i;
For i := 1 To strLen Do
(
   Local NumberVar charPos := strLen - i + 1;
   str := str + {Customer.Customer Name}[charPos]
);
str
```

Examine how this formula works assuming that the current value of the field {Customer.Customer Name} is “Clean Air”. The variable strLen is assigned to be the length of “Clean Air”, namely 9. The variable *i* is known as a *For counter variable* since its value changes with each iteration of the For loop. In other words, it is used to count the iterations of the loop. The For loop will iterate 9 times, during the first time, i is 1, then i is 2, then i is 3 and so on until finally i equals 9. During the first iteration, the ninth character of {Customer.Customer Name} is appended to the empty string variable str. Thus str equals “r” after the first iteration. During the second iteration, the eighth character of {Customer.Customer Name} is appended to str and so str equals “ri”. This continues until after the ninth iteration, str equals, “riA naelC” which is the reversed string.

Example 2

Here is a simpler version of the above formula that uses a Step clause with a negative Step value of -1. For the “Clean Air” example, i is 9 for the first iteration, 8 for the second, 7 for the third and so on until it is 1 in the final iteration.

```
//Reverse a string version 2
Local StringVar str := "";
Local NumberVar strLen :=
   Length ({Customer.Customer Name});
Local NumberVar i;
For i := strLen To 1 Step -1 Do
(
   str := str + {Customer.Customer Name}[i]
);
str
```

Example 3

The simplest version is to use the built in function StrReverse:

```
//Reverse a string version 3
StrReverse ({Customer.Customer Name})
```

The built in String functions in Crystal Reports 8.5 can handle many of the string processing applications that would traditionally be handled using a For loop or some other kind of loop. However, For loops provide the most flexibility in processing strings and also power in processing arrays, which can be essential if the built-in functions do not cover your intended application.

For loop example

Here is a more full featured example of Crystal Reports' string processing capabilities. The Caesar cipher is a simple code that is traditionally credited to Julius Caesar. In this code, each letter of a word is replaced by a letter five characters further in the alphabet. For example, "Jaws" becomes "Ofbx". Notice that "w" is replaced by "b"; since there are not 5 characters after "w" in the alphabet, it starts again from the beginning. Here is a formula that implements applying the Caesar cipher to the field {Customer.Customer Name} in the Xtreme database:

```
//The Caesar cipher

//The input string to encrypt
Local StringVar inString := {Customer.Customer Name};
Local NumberVar shift := 5;
Local StringVar outString := "";

Local NumberVar i;
For i := 1 To Length(inString) Do
(
   Local StringVar inC := inString [i];
   Local StringVar outC;

   Local BooleanVar isChar :=
   LowerCase(inC) In "a" To "z";
   Local BooleanVar isUCaseChar :=
   isChar And (UpperCase (inC) = inC);

   inC := LCase(inC);

   If isChar Then
   (
      Local NumberVar offset :=
         (Asc(inC) + shift - Asc("a")) Mod
         (Asc("z") - Asc("a") + 1);
      outC := Chr(offset + Asc("a"));
      If isUCaseChar Then outC := UpperCase(outC)
   )
   Else
      outC := inC;

   outString := outString + outC
);
outString
```

In the above example there is an If expression nested within the expression block of the For loop. This If expression is responsible for the precise details of shifting a single character. For example, letters are treated differently from punctuation and spaces. In particular, punctuation and spaces are not encoded. The general points here are that control structures can be nested within other control structures and that multiple expressions can be included in the (parentheses enclosed) expression blocks of other control structures.

Using Exit For

You can exit from a For loop by using Exit For. The following example searches the Global array names for the name “Fred”. If it finds the name, it returns the index of the name in the array. Otherwise it returns -1.

For example, if the names array is:

```
["Frank", "Helen", "Fred", "Linda"]
```

Then the formula returns 3.

```
Global StringVar Array names;

//The names array has been initialized and filled
//in other formulas
Local NumberVar i;
Local NumberVar result := -1;

//The UBound function returns the size of its array
//argument
For i := 1 to UBound (names) Do
(
   If names [i] = "Fred" Then
   (
      result := i;
      Exit For
   )
);
result
```

When considered as an expression, the For loop always returns the Boolean value True. Thus you will almost never want a For loop to be the last expression in a formula, since then the formula will then just display the value True rather than your intended result.

While Loops

Another looping mechanism is the While loop. A While loop can be used to execute a fixed block of statement an indefinite amount of time.

The 2 different types of While loops

Type of Do Loop	Explanation	Example

While ... Do	The While ... Do loop evaluates the condition, and if the condition is true, then it evaluates the expression following the Do.
	When it has finished doing this, it evaluates the condition again and if the condition is true, it evaluates the expression following the Do again. It continues repeating this process until the condition is false.
		`While` *`condition`* `Do`
		`expression`

Type of Do Loop	Explanation	Example

Do ... While	The Do ... While loop evaluates the expression once no matter what.
	It then evaluates the condition, and if the condition is true, evaluates the expression again. This process continues until the condition is false.
		`Do`
		`   `*`expression`*
		`While `*`condition`*

Note:

- The While loops support an Exit While statement to immediately jump out of the loop. Its use is analogous to the use of Exit For in For loops.
- As with the For loop, the While loop when considered as an expression always returns the Boolean value True.

While ... Do loop example

The following example searches for the first occurrence of a digit in an input string. If a digit is found, it returns its position, otherwise it returns -1. In this case, the input string is set explicitly to a string constant, but it could be set equal to a String type database field instead. For example, for the input String, “The 7 Dwarves”, the formula returns 5, which is the position of the digit 7.

```
Local StringVar inString := "The 7 Dwarves";
Local NumberVar strLen := Length (inString);
Local NumberVar result := -1;
Local NumberVar i := 1;

While i <= strLen And result = -1 Do
(
   Local StringVar c := inString [i];
   If NumericText (c) Then
      result := i;
   i := i + 1;
);
result
```

Safety mechanism for loops

There is a safety mechanism to prevent report processing from hanging due to an infinite loop. Any one evaluation of a formula can have at most 30,000 loop condition evaluations per formula evaluation. This will be explained by the example below.

For example:

```
Local NumberVar i := 1;
While i <= 100000 Do
(
   If i > {movie.STARS} Then
      Exit While;
   i := i + 1
);
20
```

If {movie.STARS} is greater than 30,000 then the loop condition (i <= 100000) will be evaluated more than the maximum number of times and an error message is displayed. Otherwise the loop is OK.

Note: The safety mechanism applies on a per formula base, not for each individual loop. For example:

```
Local NumberVar i := 1;
For i := 1 To 10000 Do
(
   Sin (i);
);
While i <= 25000 Do
(
   i := i + 1;
)
```

The above formula also triggers the safety mechanism since the 30,000 refers to the total number of loop condition evaluations in the formula and this formula will have 10001 + 25001 such evaluations.

Limitations

For reference purposes, here are the sizing limitations of the formula language:

- The maximum length of a String constant, a String value held by a String variable, a String value returned by a function or a String element of a String array is 254 bytes.
- The maximum size of an array is 1000 elements.
- The maximum number of arguments to a function is 1000. (This applies to functions that can have an indefinite number of arguments such as Choose).
- The maximum length of the text of a formula is 64K.
- The maximum number of loop condition evaluations per evaluation of a formula is 30,000.

  See “Safety mechanism for loops” on page 475.
- Date-time functions modeled on Visual Basic accept dates from year 100 to year 9999. Traditional Crystal Reports functions accept dates from year 1 to year 9999.

Parameter Fields 25

This chapter explains what parameter fields are and how they can be applied to create a single report that can be used to access different types of data depending on the user's needs.

Parameter overview

Parameters prompt the user of a report to enter information. Think of a parameter as a question that the user needs to answer before the report is generated. The information users enter, or the way they respond, determines what appears in the report. For example, in a report used by salespeople, there might be a parameter that asks the user to choose a region. The report would return the results for the specific region, instead of returning the results for all of the regions.

By using parameter fields in formulas, selection formulas, and in the report itself, you can create a single report that you can modify whenever your needs change. Parameter fields can also be used in subreports.

Parameter field considerations

There are a number of things to keep in mind when working with parameter fields:

- Parameter fields support the following data types:

  Boolean: Requires a yes/no or true/false answer.
  Example: Include planned budget numbers in the summary?

  Currency: Requires a dollar amount.
  Example: Display customers with sales over XXXXX.

  Date: Requires an answer in a date format.
  Example: Enter the start and end dates of the quarter.

  DateTime: Requires both date and time.
  Example: Display statistics for 07/04/1999 between 1:00pm-2:00pm.

  Number: Requires a numeric value.
  Example: Enter the customer identification number.

  String: Requires a text answer.
  Example: Enter the region.

  Time: Requires an answer using a time format
  Example: Display the total number of calls from 1:00pm-2:00pm.
- Parameter field prompting text can be up to four lines long with approximately 60-70 characters per line (depending on character width, up to the 254 character limit). Text over one line in length will automatically word wrap.
- You can create a pick list for the user to choose the parameter value rather than having them enter it manually.
- A parameter field does not have to be placed in a report in order to be used in a record or group selection formula. You create the parameter field and then enter it in your formula as you would any other field.

Creating a parameter field

Use the following steps to create a parameter that enables the user to specify a list of customers for a specific country. This procedure is made up of two sets of steps. The first is creating the parameter, and the second is using the Select Expert to incorporate the parameter.

To create a parameter field

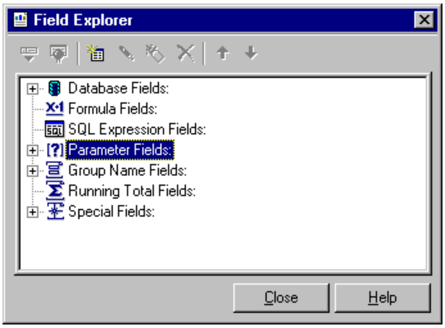
1 Check to make sure your report is open in the **Design** tab.

This procedure uses Group.rpt, a sample report that is included with the Crystal Reports software.

2 On the **Insert** menu, click **Field Object.**

Tip: Another way to do this is to click the Insert Fields button on the Standard toolbar.

The Field Explorer dialog box appears.



3 Select **Parameter Fields** and click **New**.

The Create Parameter Field dialog box appears.

4 Enter a name for the parameter in the **Name** field (up to 255 alphanumeric characters).

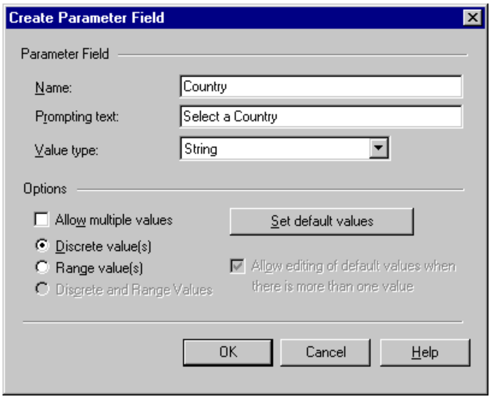
This example uses Country.

5 Enter the desired prompting text in the **Prompting text** field (up to 255 alphanumeric characters).

This is the text that appears in the Enter Parameter Values dialog box when the report is refreshed. This example uses “Select a Country.”

6 Select the appropriate **Value type** from the list.

This example uses String.



Create Parameter Field

Parameter Field

Name: Country

Prompting text: Select a Country

Value type: String

Options

☐ Allow multiple values

● Discrete value(s)

○ Range value(s)

○ Discrete and Range Values

☑ Allow editing of default values when there is more than one value

Note: When creating a parameter whose Value type is either Date or DateTime, you can change the date format to suit your needs. For details, see "Changing your default field formats" on page 227.

7 Click **Set Default Values**.

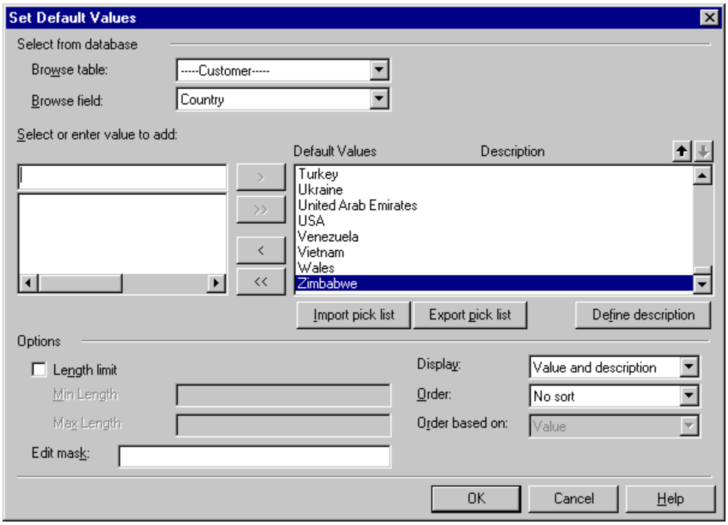
The Set Default Values dialog box appears.

8 Check to make sure the **Browse table** is set to **Customer**.

9 From the **Browse field** drop-down list, select **Country**.

10 Click **>>** to move all of the countries to the **Default Values** area.

This example will enable the user to choose from any of the countries. If you want to limit the selection, move only the countries you would like the user to choose from.



Set Default Values

Select from database

Browse table: -----Customer-----

Browse field: Country

Select or enter value to add:

Default Values	Description
Turkey	
Ukraine	
United Arab Emirates	
USA	
Venezuela	
Vietnam	
Wales	
Zimbabwe	

Options

☐ Length limit

Min Length

Max Length

Edit mask:

Display: Value and description

Order: No sort

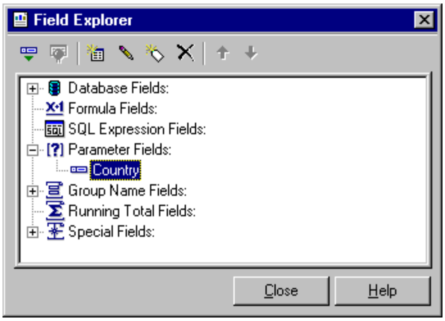
Order based on: Value

11 Click **OK**.

The Create Parameter Field dialog box appears.

12 Click **OK**.

The Field Explorer dialog box appears with the Country parameter selected.



13 Drag and drop the **Country** parameter into your report.

Note: If you don't want to see the parameter field you dropped in your report, place it in a section you can suppress, such as a report header or footer.

14 Click **Close**.

To incorporate the parameter into the selection

1 On the **Report** menu, click **Select Expert**.

Tip: Another way to do this is to click the Select Expert button on the Standard toolbar.

The Choose Field dialog box appears.

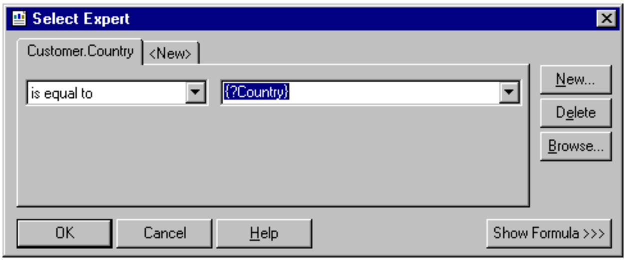
2 Select **Country** from the Customer table, then click **OK**.

The Select Expert appears.

3 Choose **is equal to** from the drop-down list.

4 Choose the parameter from the adjacent drop-down list.

This example uses {?Country}.



is equal to

{?Country}

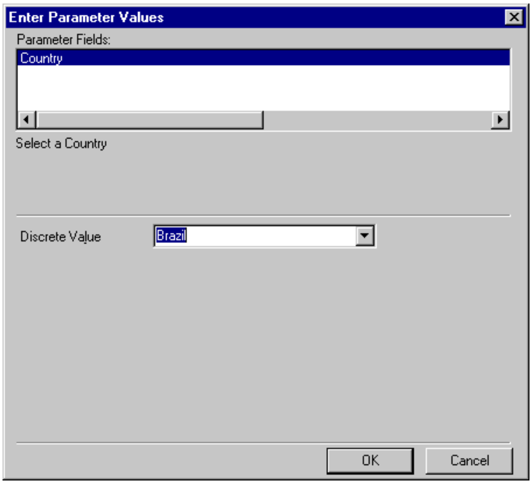
5 Click **OK**.

6 Click the **Preview** tab.

The Enter Parameter Values dialog box appears.

7 Select the country to base the report on.

This example uses Brazil.



Enter Parameter Values

Parameter Fields:

Country

Select a Country

Discrete Value: Brazil

8 Click **OK**.

The Change in Record Selection Formula Detected dialog box appears.

9 Click **Refresh Data**.

The report appears with the information for Brazil.

With parameter fields, you can create a single report that can be customized quickly to meet a variety of needs.

Deleting parameter fields

There are several methods for deleting parameters within a report. The type of parameter you are deleting determines the method you can use.

To delete a parameter that is not used in a formula

1 On the **Insert** menu, click **Field Object.**

The Field Explorer appears.

2 Expand the **Parameter Fields** folder and click the parameter you want to delete.

3. Click **Delete**.

   A Crystal Reports dialog box appears confirming whether you want to delete the parameter.

4. Click **Yes**.

   This parameter is removed from the Parameter Fields folder and from your report.

To delete a parameter used with the Select Expert

1. On the **Report** menu, click **Select Expert**.

   Tip: Another way to do this is to click the Select Expert button on the Standard toolbar.

   The Select Expert dialog box appears.

2. Choose the tab whose selection criteria uses the parameter you want to delete.
3. Click **Delete**.
4. Click **OK** to close the Select Expert.
5. On the **Insert** menu, click **Field Object**.
6. Expand the **Parameter Fields** folder and click the parameter you want to delete.
7. Click **Delete**.

To delete a parameter that is used in a formula

1. On the **Insert** menu, click **Field Object**.

   The Field Explorer appears.

2. In the **Formula Fields** folder, select the formula that contains the parameter you want to delete.
3. Click **Edit** and delete the parameter field from the formula.

   Note: If the parameter is used in more than one formula, it must be deleted from each formula.

4. Close the Formula Editor.
5. Expand the **Parameter Fields** folder and click the parameter you want to delete.
6. Click **Delete**.

Responding to parameter field prompts

Previewing a report for the first time

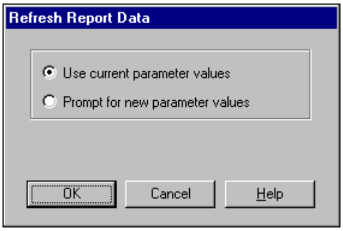
When you preview a report for the first time, the Enter Parameter Values dialog box appears, prompting you for a value.

- 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, the program will not refresh the data until you supply a new value.

Note: If the parameter is a string value type, without a default value, and nothing is entered into the discrete value field, an empty string appears.

Refreshing report data

When you refresh data from the Preview tab, the Refresh Report Data dialog box appears.



Refresh Report Data

● Use current parameter values

○ Prompt for new parameter values

Select the Use current parameter values option to use the current parameter value.

Select the Prompt for new parameter values option to enter a new parameter value. When you select this option and click OK, the Enter Parameter Values dialog box appears.

- Enter String values exactly as they will appear in the field. If the parameter field allows multiple values, you can enter additional parameter values using the Add.
- Enter Boolean values using the following format: TRUE or FALSE.
- Enter Number values exactly as they will appear in the field.
- Enter Currency values exactly as they will appear in the field.
- Enter Date values to match the format used on-screen. If the format is unspecified, enter as Date (Year, Month, Day). For example, Date (1997, 5, 21). To access the calendar, click the drop-down arrow beside the date.
- Enter Time values to match the format used on-screen. If the format is unspecified, enter as Time (Hour, Minutes, Seconds AM/PM). For example, Time (4:32:12 PM). You can also select the unit of time and then use the up and down arrows to scroll through the numbers.

- Enter DateTime values using the following format: Date (Year, Month, Day), Time (Hour, Minutes, Seconds AM/PM). For example, Date (1997, 5, 21), Time (4:32:12 PM). You can also enter DateTime values by using the associated drop-down arrow to access the calendar, and the up and down arrows to scroll through the time.

  Note: The drop-down arrow to access the calendar for dates and the up and down arrows to scroll for times will only be available if you have the correct version (4.70 or later) of comctl32.dll.

To use a different value than the default displayed, type a new value in the text box, and click OK.

To use the default value, click OK.

- If the parameter field is range limited, then you can only enter values within a certain range. The range limit is specified in the Set Default Value dialog box when creating or editing a parameter.
- If the parameter is a string value type, you can limit the values the user is able to enter by using the length limit option or by using an edit mask to restrict the format.

The program now runs the report using the new value(s) you specified.

Advanced parameter features

There are a variety of ways that parameters can be used within a report. This section covers some of the advanced methods of using parameters:

- Creating a parameter with multiple values
- Applying conditional formatting using parameter fields
- Creating a report title using parameter fields
- Specifying single or ranges of values
- Incorporating a parameter into a formula
- Defining sort order using parameter fields
- Defining entry type and format using the Edit Mask.

Creating a parameter with multiple values

1. On the **Insert** menu, click **Field Object**.
2. Select **Parameter Fields** and click **New**.
3. Enter a Name and Prompting text.
4. Select a Value type.
5. Select the **Allow multiple values** check box to enable more than one value to be entered into the parameter.

6. Click **Set default values** to determine the type of entries that are allowed to be entered.

   The Set Default dialog box appears.

7. From the **Browse table** drop-down list, select the table for the default values.
8. From the **Browse field** drop-down list, select the field for the default values.

   The values associated with this table and field appear in the “Select or enter value to add” area.

9. Select the values listed, or create you own, and move them to the Default Values area using the arrow buttons.
10. Adjust the display, order, and length limit as required.
11. Click **OK**.

Applying conditional formatting using parameter fields

Parameter fields can be used to create conditional formatting formulas. You can customize these formulas whenever you refresh the report data. A conditional formatting formula could be used for color-flagging data that meets certain conditions. For example:

- sales representatives who sell more than 10% over quota
- customers who have not ordered in the last quarter
- inventory items that have not had any movement in the last month.

If the conditions under which you flag these items never change, you do not need to use parameter fields. You can just use formulas (for text flags) or conditional formatting (for border flags). However, to change the conditions from report to report, you need to use parameter fields in formulas and conditional formatting formulas.

To apply conditional formatting using parameter fields

1. Create the parameter field of 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.
3. For example, to be prompted for all the customers whose last year’s sales were over a certain value, and to print their names in red, select the Last Year’s Sales field and click **Format** from the Supplementary toolbar.

   The Format Editor appears.

4. Click the Conditional formula button next to the Color property on the **Font** tab, and format the field using a conditional formatting formula like this:

```
If {customer.LAST YEAR’S SALES} > {?SalesTarget} Then
     Red
Else
     Black
```

Now, when you refresh the data, the program will prompt you for the value that triggers the color flag (known as the threshold value). It then runs the report and flags all the customers that had sales last year above the threshold figure. You can change the figure each time you run the report and the program will flag a different set of Customer Names.

For more information see "Conditional formatting functions" on page 422.

Creating a report title using parameter fields

Crystal Reports allows you to use parameter fields to create a report title that can be changed each time the report is refreshed.

To creating a report title using parameter fields

1. On the **Insert** menu, click **Field Object**.
   The Field Explorer appears.
2. Select **Parameter Fields** and click **New**.
   The Create Parameter Field dialog box appears.
3. Type a name for the parameter field in the **Name** field.
4. Type in any prompting text you wish in the **Prompting text** field.
5. Select String from the **Value type** drop-down list.
6. Click **Set default values**.
   The Set Default Values dialog box appears.
7. To specify a default title, type the desired text into the **Select or enter value to add** field and click > to add the title to the **Default Values** area.

   Note: Continue adding titles as required.
8. Click **OK**.
   The Create Parameter Field dialog box appears.
9. Click **OK**.
   The Field Explorer appears with the new parameter selected.
10. Place the parameter field in the Page Header section of the report to have the title appear on every page, or in the Report Header section if you want the title to appear on only the first page of the report.

Now, when you refresh the data, the program will prompt you for a report title. If you wish, you can change the title each time you run the report.

Specifying single or ranges of values

1 Select the **Discrete value(s)** option or the **Range value(s)** option to specify whether the parameter field will accept a range of values.

- If you select **Discrete value(s)**, the parameter field will accept single values (rather than ranges of values).
- If you select **Range value(s)**, then when you are prompted for parameter values, you can enter a start value and an end value, and the report will then display all records within that range. For example, if you enter the values "5" and "10", the range is 5-10, and the report will display all records with values between 5 and 10. This also works for string parameter fields. With a start value of "A" and an end value of "E", the report will display all records within an alphabetical range of A-E.

Note: If the "Allow multiple values" and the "Discrete value(s)" options are selected, the parameter field will accept multiple single values. In this case, you can enter more than one value, but these values will be evaluated individually and will not be interpreted as a range. If the "Allow multiple values" and "Range value(s)" options are selected, the parameter field will accept multiple ranges.

2 If applicable, click **Set default values** to select the **Length limit** check box to designate the length of the field. If you select **Length limit**:

- for a Currency or Number parameter field, the "Min Value" and "Max Value" fields appear
- for a DateTime parameter field, the "Start Date-time" and "End Date-time" fields appear
- for a Date parameter field, the "Start Date" and "End Date" fields appear
- for a Time parameter field, the "Start Time" and "End Time" fields appear.

Incorporating a parameter into a formula

1 On the **Insert** menu, click **Field Object**.

The Field Explorer appears.

2 Select **Parameter Fields** and click **New**.

The Create Parameter Field dialog box appears.

3 Create a parameter field and save it.

4 Select **Formula Fields** and click **New**.

The Formula Name dialog box appears.

5 Enter the name of the formula, then click **OK**.

6 Create a formula using the parameter field as you would any constant value. For example, rather than creating a formula that hard-codes the country name:

{customer.COUNTRY} = "USA"

Use a parameter field instead of “USA”.

{customer.COUNTRY} = {?Country}

To do this double-click the database field, press =, then double-click the parameter.

Tip: Identify parameter fields easily by looking for (?).

7. Click **Save and close**.

   The Field Explorer dialog box appears. The name of the formula you have just created is highlighted in the Formula list box.

8. Drag and drop the formula into the report.

9. Click **Close** to exit the Field explorer dialog box.

10. Click **Refresh** to generate the report.

    A dialog box appears, prompting you for values.

Defining sort order using parameter fields

To set the sort order using parameter fields, you need to first create a formula that includes a parameter field and then sort based 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.

1. Create a parameter field and call it **SortField**.

2. In the **Prompting text** edit box, enter 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. Select **String** from the **Value type** drop-down list.

4. You may want to limit the number of characters the user can type to one. To do this, click **Set default values**.

   The Set Default Values dialog box appears.

5. Select the **Length limit** check box and type the numeral 1 into the **Min Length** and **Max Length** text boxes and click **OK**.

   Now the parameter field will only accept single-character values. The field will accept “C” as a value, but not “City.”

6. 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 will sort by the City field. If you enter “R” it will sort by the Region field. If you enter anything else, or do not enter anything at all, the formula will sort by the Country field.

For more information see “If statements” on page 427.

7. Place the formula in the Report Header section of the report and select **Suppress (No Drill-Down)** in the Section Expert so that it does not print.
8. Click **Sort Records**.
9. Choose your formula and click **Add**.
10. Click **OK**.

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.

Defining entry type and format using the Edit Mask

1. On the **Insert** menu, click **Field Object**.
2. Select **Parameter Fields** and click **New**.

   The Create Parameter Field dialog box appears.
3. Enter the Name, Prompting text, and Value type.
4. Click **Set default values**.

   For a string parameter field that you are setting the default values for, you can choose to enter an Edit Mask in the Edit Mask field, rather than specifying a range. An Edit Mask can be any of a set of masking characters used to restrict the values you can enter as parameter values (the Edit Mask also limits the values you can enter as default prompting values).

   You can enter any of the following masking characters, or any combination of them:

   - “A” (allows an alphanumeric character and requires the entry of a character in the parameter value).
   - “a” (allows an alphanumeric character and does not require the entry of a character in the parameter value).
   - “0” (allows a digit [0 to 9] and requires the entry of a character in the parameter value).
   - “9” (allows a digit or a space, and does not require the entry of a character in the parameter value).
   - “#” (allows a digit, space, or plus/minus sign, and does not require the entry of a character in the parameter value).
   - “L” (allows a letter [A to Z], and requires the entry of a character in the parameter value).

- "?" (allows a letter, and does not require the entry of a character in the parameter value).
- "&" (allows any character or space, and requires the entry of a character in the parameter value).
- "C" (allows any character or space, and does not require the entry of a character in the parameter value).
- ". , : ; - /" (separator characters). Inserting separator characters into an Edit Mask is something like hard coding the formatting for the parameter field. When the field is placed on the report, the separator character will appear in the field object frame, like this: LLLL/0000. This example depicts an edit mask that requires four letters followed by four numbers.
- "<" (causes subsequent characters to be converted to lowercase).
- ">" (causes subsequent characters to be converted to uppercase).
- "\" (causes the subsequent character to be displayed as a literal). For example, the Edit Mask "\A" would display a parameter value of "A." If the Edit Mask is "00\A00," then a valid parameter value would consist of two digits, the letter "A," and then two additional digits.
- "Password". Allows you to set the Edit Mask to "Password," you can create conditional formulas specifying that certain sections of the report become visible only when certain user passwords are entered.

Note: Some of the Edit Mask characters require that you enter a character in their place (when entering a parameter value), while others allow you to leave a space, if needed. For example, if the Edit Mask is 000099, you can enter a parameter value with four digits, five digits, or six digits, since the '9' Edit Mask character does not require the entry of a character. However, since '0' does require such an entry, you could not enter a parameter value with less than four digits.

5 Enter the default prompting values by adding to or highlighting values on the "Select or enter value to add" list and using the **Add** and **Add All** buttons to add these values to the **Default Values** list.

You can use the Remove and Remove All buttons to remove prompting values from the list. The items you add appear in a drop-down list on the Enter Parameter Values dialog box with the default prompting values you specify.

6 Click **OK**.

The Create Parameter Field dialog box appears.

Note: When you have more than one item in the "Default Values" list box, the "Allow editing of default values" check box appears. This check box is selected by default to specify that you can edit or enter new values when prompted for parameter values. Select or clear this check box as needed.

7 Click **OK**.

The Field Explorer appears with the parameter selected.

8 Drag and drop the parameter into the report.

Subreports 26

A subreport is a report within a report. With subreports, unrelated reports can be combined into a single report. You can coordinate data that otherwise cannot be linked and present different views of the same data in a single report. This chapter shows you how to create and use subreports.

What are subreports?

A subreport is a report within a report. The process for creating a subreport is similar to the process of creating a regular report. A subreport can have most of the characteristics of a report, including its own record selection criteria. The only differences between a subreport and a primary report are that a subreport:

- is inserted as an object into a primary report; it cannot stand on its own (although a subreport can be saved as a primary report)
- can be placed in any report section and the entire subreport will print in that section
- cannot contain another subreport.

There are four instances in which a subreport would typically be used:

- To combine unrelated reports into a single report. See "Combining unrelated reports by using subreports" on page 503.
- To coordinate data that cannot otherwise be linked. See "Using subreports with unlinkable data" on page 504.
- To present different views of the same data within a single report. See "Showing different views of the same data in a report" on page 508.
- To perform one-to-many lookups from a field that is not indexed on the lookup field. See "One-to-many links" on page 565.

Note: You can increase the performance of reports containing subreports by using on-demand subreports instead of regular, in-place subreports.

Unlinked vs. linked subreports

Unlinked

Unlinked subreports are free-standing; their data is not in any way coordinated with the data of the primary report.

In unlinked subreports, there is no attempt to match up the records in one report with records in the other. An unlinked subreport does not have to use the same data as the primary report; it can use the same data source or a different data source entirely. In addition, the subreport is not limited to reporting on a single table. An unlinked subreport can be based on a single table or on multiple tables. Regardless of the underlying data sources, the reports are treated as unrelated.

Linked

Linked subreports are just the opposite; their data is coordinated. The program matches up the records in the subreport with 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 that customer.

How subreport linking works

When you link a subreport to a primary report, the program creates the link by using a parameter field.

When a subreport link field is selected, the program creates:

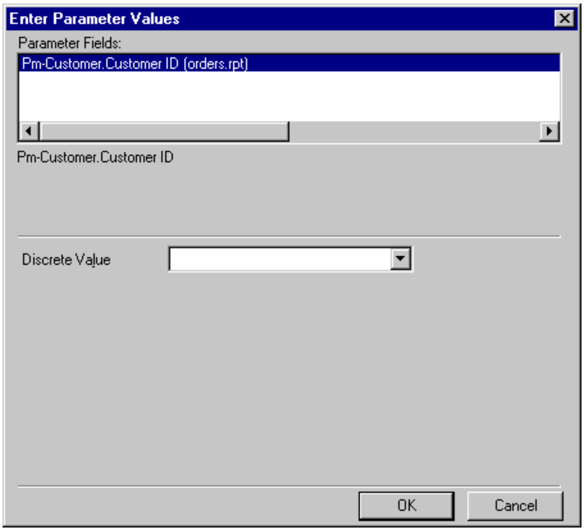
- a parameter field in the subreport which is then used to retrieve values passed to it by the primary report
- a record selection formula for the subreport using the parameter field
- the selection formula limits the subreport to those records in which the link field is equal to the parameter field value.

When the report is run, 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:

- You create a report that shows customer data and a subreport that shows order data and then you link the two reports using the Customer ID field.
- 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.
- 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.
- When the subreport is finished, the program locates 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.
- The program then runs a subreport including only those order records for the second customer.
- The process continues until the report is finished.
- All of this parameter field manipulation takes place behind the scenes. You simply pick the fields that will 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 on the Standard toolbar (from the Subreport Design tab), the program runs the subreport on its own, without waiting to receive a parameter field value from the primary report and without evaluating the tab text formula. In this case, the program displays the Enter Parameter Values dialog box and prompts you for a value.

The value entered in the drop-down box is the value the program uses to run the subreport.



Database links vs. subreports in one-to-many situations

When two tables in a report have a one-to-many relationship, the program retrieves the data in different ways depending on:

- data source
- index situation
- record selection criteria
- 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 a subreport or linked tables, you need to understand the ramifications of each. These issues are discussed fully in "Performance considerations in one-to-many links" on page 517.

As a general rule, if you have:

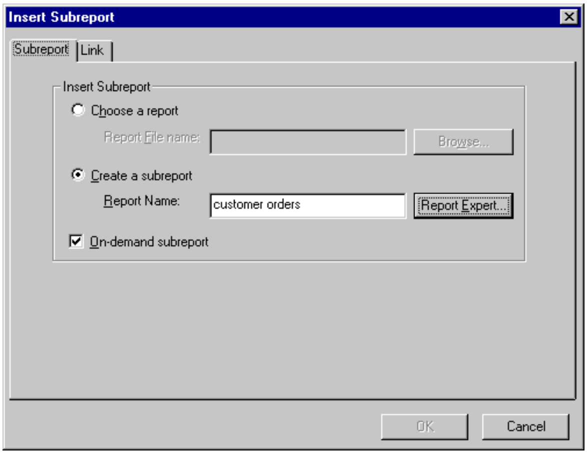
- indexed tables
- linked indexed fields
- range limiting record selection criteria based on the indexed fields

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, linked tables may have a performance advantage. See "Indexed tables" on page 514.

Inserting subreports

1 On the **Insert** menu, click **Subreport**.

The Insert Subreport dialog box appears.



○ Choose a report

Report File name:

● Create a subreport

Report Name: customer orders

☑ On-demand subreport

2 To choose an existing subreport, click **Choose a report** and type the name. 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, click “Create a subreport” and type a name. If you would like assistance in creating the subreport, click the Report Expert button.

3 Choose **On-demand subreport** to have the ability to retrieve the data on the subreport when needed. Otherwise, all the subreport data will appear with the report.

4 Click **OK**.

The program displays an object frame.

5 Move the frame where you want it to appear in the report and click once to place it.

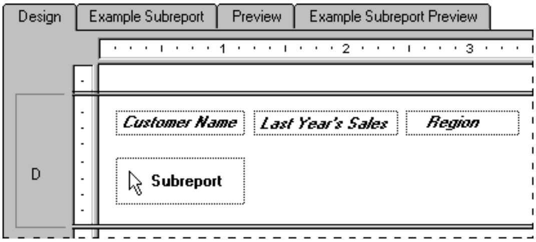
If you imported the subreport, the program creates a Subreport Design tab, which is labeled with the subreport name.

- To edit the report, click the Subreport Design tab and make your modifications.
- If you do not want to edit the report, you are finished.

If you are creating a new subreport, the program creates a Subreport Design tab labeled with the subreport name.

- To edit the subreport further, after exiting the Report Expert, click the Subreport Design tab and finish modifying the subreport as you would with any other report.

- If you do not want to edit the report, you are finished.



6 Click the **Preview** tab to see your report.

7 If you chose the “On-demand subreport” option, click the subreport preview tab to see your subreport.

This tab is labeled with the name of your subreport.

Note: Using on-demand subreports will increase the performance of reports that contain subreports.

For information about creating a custom caption for the Subreport Preview Tab, see “Adding captions to on-demand subreports” on page 507.

Previewing subreports

There may be times when you want to preview a subreport on its own instead of previewing it as a part of the main report. For example, you may want to preview the subreport in order to view and analyze the data for a particular set of parameter values.

To preview a subreport

1 Click the subreport design tab.

This tab is labeled with the name of your subreport.

2 Click **Print Preview** on the Standard toolbar.

The program displays a preview of the selected subreport.

Saving a subreport as a primary report

You may find it advantageous to save a subreport as a primary report for the sake of distributing the information to a variety of audiences. For example, the primary report containing the subreport may be relevant for a stockholders meeting at the end of the fiscal year; however, the data contained in the subreport may be relevant for everyday use by your managers. In such cases, it is easy to save a subreport as a primary report.

To save a subreport as a primary report

1. In the Design tab, right-click the subreport and click **Save Subreport As** from the shortcut menu.

   The Save As dialog box appears.
2. Highlight the appropriate directory in the list box.
3. Type a new name for the subreport.
4. Click **Save**.

   The program saves the subreport as a primary report so that you can open it separately when necessary.

Updating subreports

In order to maintain the most up-to-date subreports, you may want to re-import a subreport automatically when opening the main report.

Note: Re-importing is available only for subreports that were created from a report file.

Re-importing not only updates the data, but updates the formatting, grouping, and structure of the subreport if any changes have been made. If you change the report the subreport was originally based on, you can have these changes reflected in the report containing the subreport.

You can either globally specify that all subreports should be re-imported, or set the automatic re-import for an individual subreport.

To globally update subreports when opening a main report

1. On the **File** menu, click **Options**.
2. In the Options dialog box, click the **New Report** tab.
3. Click **Re-import Subreports When Opening Reports**.
4. Click **OK**.

   Any subreport will be updated when its main report is opened and refreshed.

To update a specific subreport when opening a main report

1. From the **Format** menu, click **Format Subreport**.
2. In the Format Editor dialog box, click the **Subreport** tab.
3. Click **Re-import when opening**.

   The current subreport will be updated when the main report is opened and refreshed.

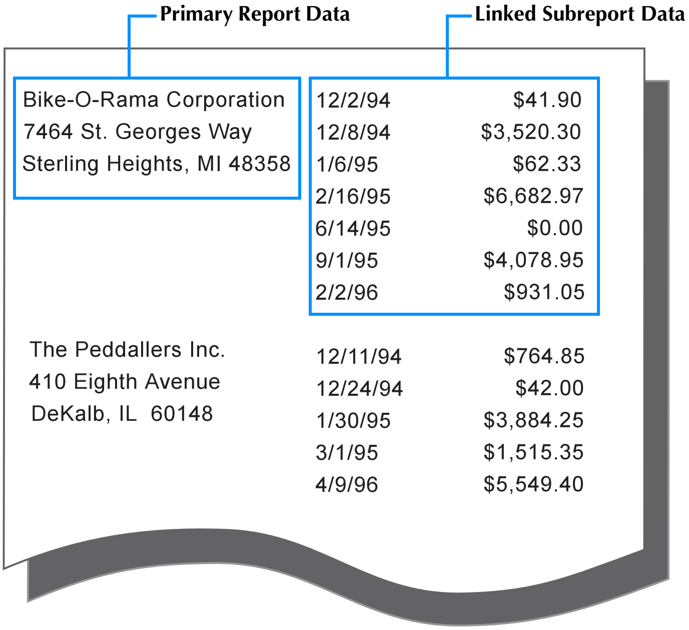
Manually updating subreport data

You can update your subreport data at any time.

1. On the **Design** tab, right-click the subreport.
2. Click **Re-import subreport** from the shortcut menu.
3. Click **Yes** to update the subreport data.

Linking a subreport to the data in the primary report

Frequently, the data in a subreport supplements the data in the primary report. You might, for example, have customer data in a primary report and then use subreports to show the orders for each customer.



Bike-O-Rama Corporation
7464 St. Georges Way
Sterling Heights, MI 48358

12/2/94	$41.90
12/8/94	$3,520.30
1/6/95	$62.33
2/16/95	$6,682.97
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 cases, you will need to coordinate the data in the primary report with the data in the subreport so that the orders in each subreport match up with the correct customer.

To do this, you need to specify a field that is common to both the subreport and the primary report. With the Subreport Links dialog box, you create a link between the two common fields. Crystal Reports uses the link to match up records from the primary report to those in the subreport. The link makes certain that the "orders" data in the subreport sits on the same row as the corresponding "customer" data in the primary report.

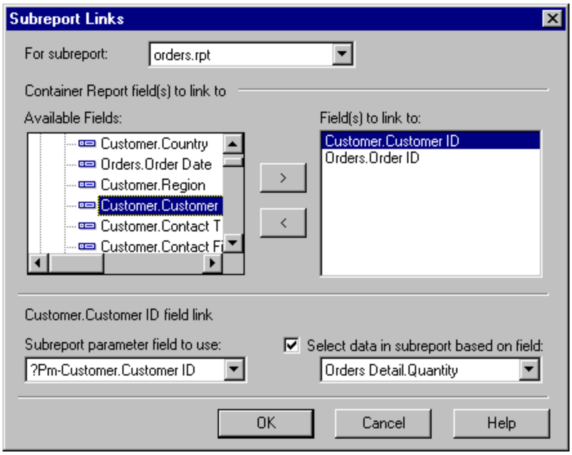
To link a subreport to the data in the primary report

1 If you are creating a new subreport or importing an existing report as a subreport, from the **Insert** menu, click **Subreport**. Choose or create a report and click the **Link** tab.

- or -

If you have already placed a subreport in the primary report, but did not create a link at setup, navigate to the Subreport Links dialog box by choosing **Subreport Links** from the **Edit** menu.

The Subreport Links dialog box appears.



Subreport Links

For subreport: orders.rpt

Container Report field(s) to link to

Available Fields:
Customer.Country
Orders.Order Date
Customer.Region
Customer.Customer
Customer.Contact T
Customer.Contact Fi

Field(s) to link to:
Customer.Customer ID
Orders.Order ID

Customer.Customer ID field link

Subreport parameter field to use: ?Pm-Customer.Customer ID

☑ Select data in subreport based on field: Orders Detail.Quantity

2 Choose the subreport you want to link from the **For subreport** list (if it is not already selected).

3 Select the field you want used as a link field in the primary (containing) report from the **Available fields** list.

4 Click the arrow (>) button.

The field is added to the Field(s) to link to list box, and is now selected as a link field.

5 Repeat steps 3 and 4 for each additional link, as desired.

6 Use the **Field link** section (which will only appear if you have selected a link field) to set up the link for each link field:

- select the field you want linked to the primary report from the **Subreport parameter field to use**,
- select the **Select data based on field** check box on and select a field from the adjacent drop-down list to organize the subreport data based on a specific field (this is the quick equivalent of using the Select Expert). If nothing is specified here, the subreport will adopt the organization of the primary report.

7 Click **OK**.

When you run the report, the program will coordinate the data in the primary report with the data in the subreport.

Note: The field type of the Containing Report field determines which subreport fields are visible. Because the Report Designer reads dates as either strings, dates, or date/time fields, you must make sure your subreport parameter field type matches the field type set up in Report Options in the main report for the field you want linked.

Linking a subreport to the main report without modifying the selection formula

Crystal Reports uses a parameter field mechanism for linking subreports to main reports.

When linking a main report field that is not a parameter field to a subreport field, the program:

- automatically creates a parameter field to complete the link
- modifies the subreport record selection formula to select those records in which the subreport field is equal to the parameter field value.

The need for a parameter field is implied; it is called an “Implicit Link” situation.

At times, you may wish to use a linked parameter field in a subreport without using it as part of the selection formula for the subreport. For instance, you may want the main report to pass in a summary value that can be used in calculations by the subreport, or you may want the main report to pass in the title of the subreport.

When you link a field in the main report to a parameter field that you have created in the subreport, the program:

- checks the link you have specified
- does not create any additional parameter fields
- does not modify the subreport record selection formula.

Specifying a link is called an “Explicit Link” situation.

To link a subreport to a main report without modifying the selection formula

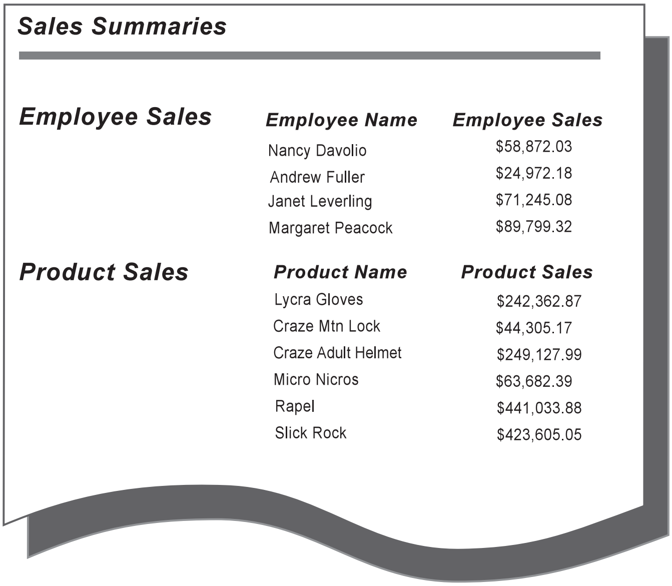
1. Create a parameter field in the subreport.
2. Link a field in the main report to that parameter field.

Combining unrelated reports by using subreports

At times, you may wish to combine unrelated reports into a single report. For example, you may want to create a single report that presents:

- sales grouped by sales representative
- sales grouped 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

Subreports can be used to combine unrelated reports into a single report like this. 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 free-standing; the data in any of the reports is not linked in any way to data in another report. This is the easiest of the subreport options to work with.

Combining two or more unrelated reports

To combine two unrelated reports

1. Create the report you want printed first as the primary report.
2. Import an existing report for use as a subreport or create a new subreport.
3. Place the subreport into the Report Footer and it will print immediately after the primary report.

To combine three or more unrelated reports

1. Create the report you want printed first as the primary report.
2. Import or create each of the other reports you want to use as subreports.
3. Use the Section Expert to 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 that you have a total of three Report Footer sections.
4. In Report Footer A, place the subreport you want printed immediately after the primary report. In Report Footer B, place the subreport you want printed next, and so forth.

   The primary report will print first and then the subreports in the order that you placed them in the report.

   Note: Subreports can be placed side-by-side in the same Report Footer section. They will print next to each other at the end of the report.
5. Place the subreports into the Report Footer sections and they will print sequentially after the primary report.

Related topics

"Working with sections" on page 182.

Using subreports with unlinkable data

Tables can be linked in a report as long as the following criteria are met:

- the link fields are both database fields
- the link fields contain similar data
- the link fields are the same length
- the link field in the link to (lookup) table is indexed (PC databases only).

Linking tables is rarely a problem. However, there are some circumstances in which you cannot coordinate data from different tables because the data does not meet the linking criteria.

For example, linking to or from a formula field, or linking two unindexed tables cannot be done in a single report. Subreports must be used.

Linking to/from a formula field

There are situations in which you may need to link to or from a formula (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).

The formula language makes it 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, by using a subreport.

To link to/from a formula field

1. Create the primary report using a table that includes the Social Security Number field.
2. Create (or import) a subreport using the formula that extracts the Social Security Number from the Employee ID field (for this example, {@EXTRACT}).

   See "Inserting subreports" on page 497.
3. Place the subreport where you want it to appear in the primary report.
4. Link the subreport to the primary report by linking the Social Security Number field in the primary report ({file.SSN}) to the formula that extracts the number in the subreport ({@EXTRACT}). See "Linking a subreport to the main report without modifying the selection formula" on page 502.

Linking unindexed tables

When using PC (not SQL or ODBC) databases, the link field in the lookup database needs to be indexed in order to create a valid link. You can not link the tables in a single report when two tables contain related data yet neither is indexed on the field which you want to use as a link field, or when the primary table is indexed but the lookup table is not. 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 may cause inefficient reporting. If your data set is large, this kind of report will take considerable time to run. Use this technique only if you do not have other options.

To link unindexed tables

1 Create the primary report.

2 Create (or import) the subreport and insert it into the primary report.
See "Inserting subreports" on page 497.

3 Use the unindexed fields (or the indexed field in the primary table and the unindexed field in the lookup table) to link the subreport to the primary report. See "Linking a subreport to the data in the primary report" on page 500.

Creating an on-demand subreport

On-demand subreports can be especially useful when you want to create a report that contains multiple subreports. In this case, you can choose to have these subreports appear only as hyperlinks.

The actual data is not read from the database until the user drills down on the hyperlink. This way only data for on-demand subreports that are actually viewed will be retrieved from the database. This makes the subreports much more manageable.

Note: Data for an on-demand subreport is not saved unless the subreport is actually open in a preview window.

To create an on-demand subreport

1 Place an ordinary subreport in your primary report.

2 Click the **Format** button on the Supplementary toolbar.
The Format Editor dialog box appears.

3 Click the **Subreport** tab and select the **On-demand subreport** check box on.

4 Click **OK**.

Adding captions to on-demand subreports

To further organize a report, captions can be created for the Subreport Preview tab and for the placeholder frame of an on-demand subreport.

Captions are written by using formulas. Both placeholder frame captions and Subreport Preview tab captions can include field names from the main report.

A tab text caption replaces the subreport file name caption on the Subreport Preview tab.

Note: Placeholder frame captions only apply to on-demand subreports, while tab text captions apply to both on-demand subreports and regular subreports. Since the data from a regular subreport is visible on the Preview tab, there is no need for a frame caption when you format a regular subreport.

To add a caption

1 Select the subreport and click the **Format** button on the Supplementary toolbar.

The Format Editor dialog box appears.

2 Click the Subreport Tab.

3 You can enter either an on-demand subreport caption or a tab text caption by clicking the appropriate **Formula** button and opening the Format Formula Editor.

4 Enter your formula in the **Formula text** box.

Crystal syntax formula example:

```
"More Information About " + {Customer.Customer Name}
```

Basic syntax formula example:

```
formula = "More Information About" + {Customer.Customer Name}
```

Using the Xtreme.mdb sample database, these formulas would give you a caption like “More Information About Pathfinders” or “More Information About Rockshocks for Jocks.”

5 Click **Check** to check the formula for errors. If the program finds an error, it will prompt you with a message box detailing the nature of the error.

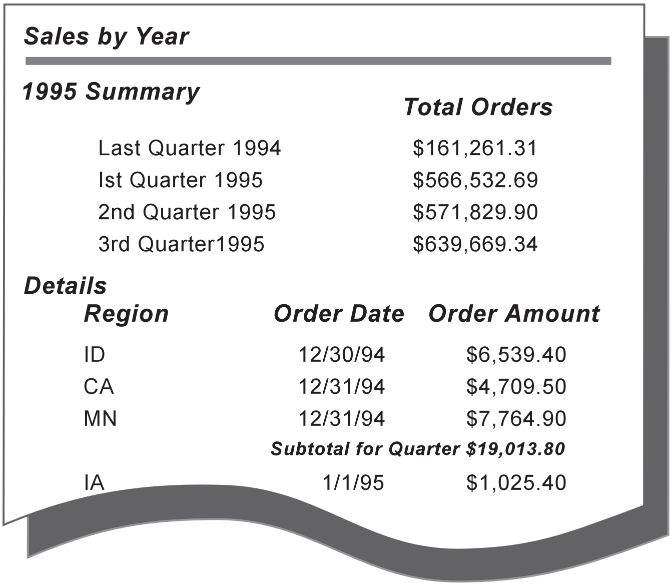
6 After fixing any errors, click **Save and Close**.

7 Click **OK** to return to the report.

Note: A tab text formula is only evaluated when you drill-down on a subreport. If you preview a subreport separately from the primary report, the formula will not be evaluated.

Showing different views of the same data in a report

Subreports can be used to provide a different view of the data in the primary report. For example, assume you want to show summary values at the top of a report and details at the bottom, like this:



Sales by Year

1995 Summary

	Total Orders
Last Quarter 1994	$161,261.31
Ist Quarter 1995	$566,532.69
2nd Quarter 1995	$571,829.90
3rd Quarter1995	$639,669.34

Details

Region	**Order Date**	**Order Amount**
ID	12/30/94	$6,539.40
CA	12/31/94	$4,709.50
MN	12/31/94	$7,764.90
	Subtotal for Quarter	**$19,013.80**
IA	1/1/95	$1,025.40

This can be done in a variety of ways. The two easiest methods are:

- By creating the summary report as the primary report and the details report as the subreport. In this method, the details subreport would be placed in the Report Footer section.
- By creating the details report as the primary report and the summary report as the subreport. In this method, the summary report would be placed in the Report Header section.

Use the appropriate link fields to link the report and coordinate the data.

Understanding Databases 27

This chapter describes the essentials you'll need to know to understand database design and use. After describing relational databases (including those from ODBC data sources), indexing, and table linking, the chapter introduces the concepts of server-side processing (pushing report processing to the server level) and field mapping (re-establishing report and database field mappings after a database changes). Finally, the chapter focuses on common database-related tasks such as changing a database driver and creating an ODBC data source.

Databases overview

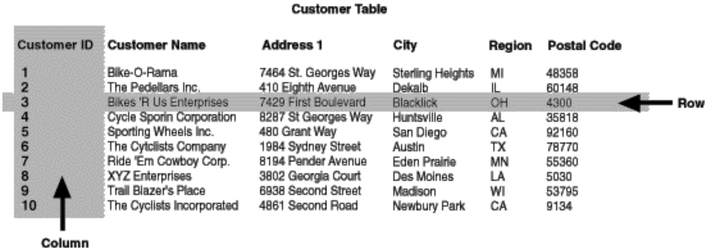
Though there are hundreds of Database Management Systems (DBMS) available, Crystal Reports eliminates many of the differences once it connects to the actual database files. The process of working with database files, tables, fields, and records is much the same, regardless of the actual type of data 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. "Using SQL and SQL databases" on page 535, 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:

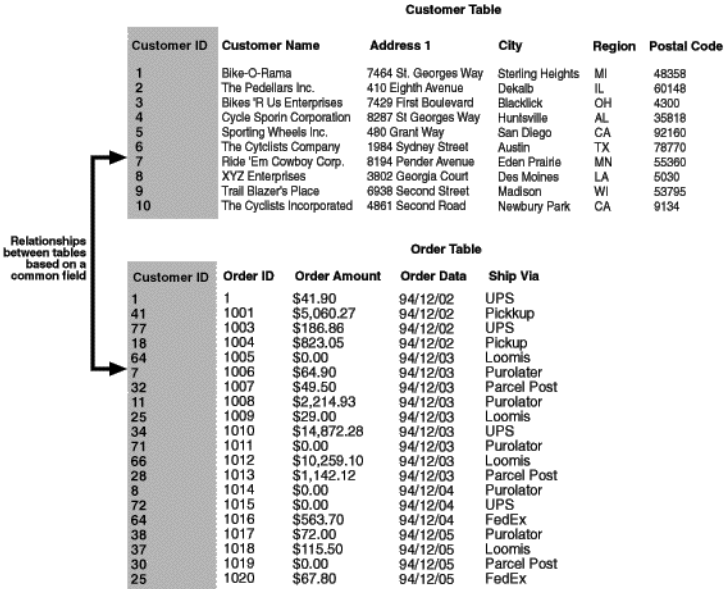


Customer Table

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 Cytclists 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	LA	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, demonstrating a relationship between tables. The two tables can be linked by a common field see "Linking tables" on page 515.

The following diagram displays how two tables can have a relationship:



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 Cytclists 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	LA	5030
9	Trail Blazer's Place	6938 Second Street	Madison	WI	53795
10	The Cyclists Incorporated	4861 Second Road	Newbury Park	CA	9134

Relationships between tables based on a common field

Order Table

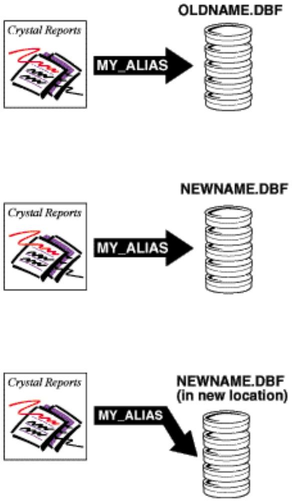
Customer ID	Order ID	Order Amount	Order Data	Ship Via
1	1	$41.90	94/12/02	UPS
41	1001	$5,060.27	94/12/02	Pickkup
77	1003	$186.86	94/12/02	UPS
18	1004	$823.05	94/12/02	Pickup
64	1005	$0.00	94/12/03	Loomis
7	1006	$64.90	94/12/03	Purolater
32	1007	$49.50	94/12/03	Parcel Post
11	1008	$2,214.93	94/12/03	Purolator
25	1009	$29.00	94/12/03	Loomis
34	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

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, the Report Designer 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 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, the Report Designer uses aliases to refer to database tables and files. Aliases are pointers, internal devices that tell the program 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” on page 512. The name of the alias does not change, so your formulas are not affected. The Report

Designer 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.

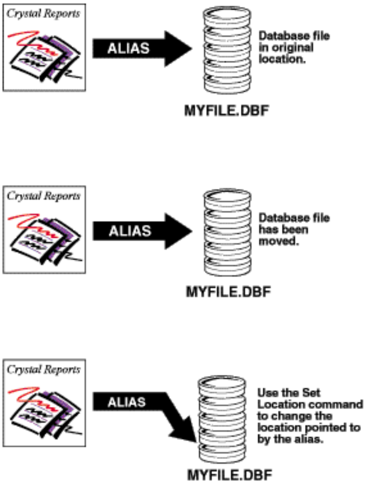


The Report Designer automatically assigns default alias names to database tables when you first select the table or file. By default, an alias matches the original name of the table. In databases where the database table is a separate file (for instance, dBASE), the name of the database file is used without the file name extension. For example, if you are using the dBASE database file Company.dbf, the program 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.


Crystal Reports
ALIAS
Database file in original location.
MYFILE.DBF
Crystal Reports
ALIAS
Database file has been moved.
MYFILE.DBF
Crystal Reports
ALIAS
Use the Set Location command to change the location pointed to by the alias.
MYFILE.DBF

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, the program 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 automatically converts your database driver to the data source you have chosen. For example, you can automatically convert a direct access data source to an ODBC data source using the Set Location command. The following guidelines apply:

- If you are moving only one table, the Set Location command can convert it to the same type or any other type of data source.
- If you are moving more than one table, the Set Location command can convert all of the tables to the same type of data source. To move more than one table to a different data source, use the Convert Database Driver option.

Related topics

“Changing the ODBC data source accessed by a report” on page 561.

Indexed tables

Creating indexes for database tables can increase the speed of data access and reduce the time it takes for the program 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 Visual Linking Expert can also help you determine if your tables include indexes.

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
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 want to look up information specific to a certain customer?

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 are 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 advantage 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 you link, you are using 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 this 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. 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, the Report Designer 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, select the Allow partial text matches check box on 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 works only 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” cannot 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
- 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 Xtreme.mdb database, 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, hiring information, etc. 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 Xtreme.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 are 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 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.

The tables 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, an 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 "Accessing Data Sources" on page 573.

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 on the field you are going to use indexed to match up the records?
- **Index B**
  Is Table B on the field you are going to use indexed 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?

PC Data						
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
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 you link tables in a single report or when you add a subreport to your report.

Linking data files

When retrieving data from linked data files in one-to-many situations, the program uses the following process:

- 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. 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” operator contains range selection criteria for the Region field. The region must fall alphabetically between “CA” and “IL.” The program passes this kind of condition down to either the database DLL (for PC data) or the server (for SQL data). See “Record Selection” on page 125.

  The second half of the selection formula, however, requires processing that must be done in the Report Engine. It uses a built-in function to manipulate and evaluate a field value and it cannot 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 this example), the program goes directly to the record it is seeking in Table A (the first CA record) and reads it.
  - For that record, the program locates the first matching record in Table B, using the Table B index.
  - The program passes this merged record (A+B) back to the Report Designer, which tests the record against the entire selection formula.
  - The program then reads the second matching record and passes the merged record on, and then reads the third matching record, and so on, until it has read all of the matching records.
  - The program then 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; the field is indexed and the records are in alphabetic order. But the program 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, the program begins the matching process again for that record.
  - The program 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; otherwise, 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 first record it finds.
  - for that record, the program uses the Table B index to locate the first matching record in Table B
  - the program passes this merged record (A+B) back to the Report Engine, which tests it against the entire selection formula
  - the program 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
  - the program then moves on 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, the subreport is based on Table B, and the records are linked, your primary considerations are as follows:

- the number of subreports that are run by the program is determined by the index and the 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 two subreports
  - 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 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) when it runs a subreport
  - if you do not have an index on Table B, the program will always read every record in Table B (2600) when it runs a subreport.

SQL database considerations

Since indexes are not critical with SQL data, the primary concern with both linked tables and 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" on page 519.

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 does not 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 either case, the Report Engine will then apply the entire selection formula to the merged records.

Subreports and SQL databases

If you are creating a primary report based on Table A and a subreport based on Table B:

- The number of subreports that are 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 remains the same regardless of whether there was range limit selection on Table A. Each subreport will read only those records in Table B that match each record read in Table A (100).

Performance considerations for all reports

Consideration 1

With both data files and SQL databases, the program parses the entire selection formula and passes down whatever parts of the criteria it is able to translate (pass), wherever they may physically appear in the formula. Thus, if the formula finds criteria it can pass, then criteria that it cannot, 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, the program passes down the criteria that it can to the database translation layer.
- In the case of SQL databases, the program passes down to the server the criteria that it can in the form of a WHERE clause.

While there are exceptions, as a general rule the program 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 Expert (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 you need to consider. In these situations, the record selection formula includes multiple conditions, some of which can be passed down while others cannot.

- 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 operator but not the condition after. Since the only records that will meet the second condition will have to meet the first as well, the program passes down the first condition, retrieves the data set that satisfies the condition, and then applies the second condition only to 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 operator but not the condition after. Since there are records that can satisfy the second condition without satisfying the first, passing the first condition down does not 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) before it can apply the second condition in Report Designer. 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 Report Designer. The rule for OR situations is that the program either passes down all the tests, or none of the tests.

  Note: If all the tests in an OR situation can be performed on the server or in the database DLL, the program passes them all down.

Consideration 2

To make certain the program can use the index on Table A to enhance performance, make certain:

- There is a selection formula.
- There are range limits in the selection formula on the key (indexed) field in Table A.
- Use Indexes is selected in the Options dialog box.

Consideration 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 that field. 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 U.S. 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 improve speed by limiting your report to orders placed 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 and afterward (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

The Visual Linking Expert lets you easily link two or more tables. When you choose Add Database To Report from the Database menu and select an additional database table, the Visual Linking Expert appears and displays that database table.

The easiest way to link database tables is to select Smart Linking in the Visual Linking Expert. 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 used to link two tables must have the same data type. For example, you can 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 cannot link a numeric field in one table to a string field in another table.

Note:

- Some DBMS applications allow you to convert the field value 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 choose to use that field to link to another table, you must link to a field of the original data type. You cannot link a string value to a numeric field that has been converted to a string in the index.
- 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 cannot link different data sources in Crystal Reports using anything other than string values.

- When linking direct-access database tables using native drivers (non-SQL), the only join type available is Left Outer join.

Changing the index used in linking

When using 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, use the Index section of the Link Options dialog box.

To call up the dialog box, use one of the following methods:

- double-click the link line of interest
- select the link line of interest and choose Options at the bottom of the Visual Linking Expert
- right-click the link line of interest and choose Options from the shortcut menu.

The Index section of the Link Options dialog box has two parts:

- **The Index In Use text box**.
  This text box displays the index that is currently in use. If you click the arrow, it 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 would expect to see on this list, use the Add Index button.
- **The Fields in Index text box**.
  This text box displays the fields that are included in the index that is currently selected in the Index In Use list 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 to learn 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 options for looking up records in two or more lookup tables from a single primary table:

- "Look up both at the same time" on page 525.
- "Look up all of one, then all of others (A to B, A to C)" on page 526.
- "Look up all the combinations of the two files" on page 526.

Note: These options are not available if you are using data from an ODBC data source.

These options are available only 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 lookup 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 this example, for each method demonstrated, three fields from three different linked tables will be 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 actual 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.

Customer Table	Credit Table	Orders Table
Customer Name	**Amount**	**Order Amount**
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 if the Orders table runs out of records.

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 looks for all the matching records in the Orders table (Table C). Then it repeats the process with the next record in the Customer table, and continues with the rest of the records.

Customer Table	Credit Table	Orders Table
Customer Name	**Amount**	**Order Amount**
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 the report before Table B data (Credit table), you need to change the links so that the A to C link comes first, followed by the A to B link. You do this in the Visual Linking Expert. To change the order of the links, delete the existing links and set up new links in the desired order.

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, and then 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.

Customer Table Customer Name	Credit Table Amount	Orders Table Order Amount
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 (the Orders table in this example), and then find all matching records in Table B (Credit table), the reverse of the current process, you will need to change your links so that the A to C link comes first, followed by the A to B link. You do this in the Visual Linking Expert. 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 linking SQL tables. An SQL join indicates how linked fields in two SQL tables are compared when records are read. The SQL Join Type options can be specified in the Link Options dialog box.

Note: When you link fields using SQL joins, no indexed fields are required.

The join types are:

- "Equal [=] join" on page 528
- "Left Outer [=(+), *=] join" on page 529
- "Right Outer [(+)=, =*] join" on page 529

- "Greater Than [>] join" on page 530
- "Less Than [<] join" on page 531
- "Greater Than Or Equal [>=] join" on page 532
- "Less Than Or Equal [<=] join" on page 533
- "Not Equal [!=, <>] join" on page 534.

Equal [=] 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 the program finds a Customer ID in the Orders table that matches a Customer ID in the Customer table, it displays information 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:

Customer Table	Customer Table	Orders Table
Customer ID	**Customer Name**	**Order Amount**
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 in which the linked field value in both tables is an exact match. It also includes a row for every record in the primary (left) table for which the linked field value has no match in the lookup table. For instance, you can use a Left Outer join to view all customers and the orders they have placed, but you also get a row for every customer who has not placed any orders. These customers appear at the end of the list with blanks in the fields that would otherwise hold order information:

Customer Table	Customer Table	Orders Table
Customer ID	**Customer Name**	**Order Amount**
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
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 uses ODBC syntax in the SQL statement. If you are connecting to an SQL database directly (not through ODBC), Crystal Reports uses a syntax native to the database. For more information about what an Outer join looks like in an SQL statement, refer to Microsoft ODBC™ documentation or to the documentation for your SQL database.

Right Outer [(+)=, =*] join

The result set from a Right Outer join includes all the records in which the linked field value in both tables is an exact match. It also includes a row for every record in the lookup (right) table for which the linked field value has no match in the primary table. If you link the Customer table to the Orders table, you get one row in the table for each order a customer has placed, as with an Equal join. You also get a row for every order found that cannot be linked to a customer. Theoretically, this should not 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 of the Customer fields for the order without a customer.

Customer Table Customer ID	Orders Table Order ID	Orders Table Order Amount
52	6	25141.50
53	11	19164.30
53	21	1683.60
57	4	15716.40
58	20	1956.20
60	16	24580.50
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 uses ODBC syntax in the SQL statement. If you are connecting to a SQL database directly (not through ODBC), Crystal Reports uses a syntax native to the database. For more information about what an Outer join looks like in an SQL statement, refer to Microsoft ODBC documentation or to 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 their sales representatives to the salaries made by all 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:

SalesRep Table	SalesRep Table	Manager Table	Manager Table
Last Name	**Salary**	**Last Name**	**Salary**
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 established between sales representatives and sales managers. 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 in the primary table is less than the linked field value in the lookup table. Using the Less Than join, you 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, you 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 that produced by the Greater Than join:

Manager Table	Manager Table	SalesRep Table	SalesRep Table
Last Name	Salary	Last Name	Salary
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 in 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, except that 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:

SalesRep Table	SalesRep Table	Manager Table	Manager Table
Last Name	Salary	Last Name	Salary
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
Dodsworth	$48,300.00	Fuller	$32,000.00
Dodsworth	$48,300.00	Brid	$30,000.00

SalesRep Table Last Name	SalesRep Table Salary	Manager Table Last Name	Manager Table Salary
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 in 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, except that 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 similar to the following:

Manager Table Last Name	Manager Table Salary	SalesRep Table Last Name	SalesRep Table Salary
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
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 in 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 considers it 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:

Product1	**Product2**
Product Name	**Product Name**
Xtreme Adult Helmet	Xtreme Mtn Lock
Xtreme Adult Helmet	InFlux Lycra Glove
Xtreme Adult Helmet	Roadster Micro Mtn Saddle
Xtreme Mtn Lock	Xtreme Adult Helmet
Xtreme Mtn Lock	InFlux Lycra Glove
Xtreme Mtn Lock	Roadster Micro Mtn Saddle
InFlux Lycra Glove	Xtreme Adult Helmet
InFlux Lycra Glove	Xtreme Mtn Lock
InFlux Lycra Glove	Roadster Micro Mtn Saddle
Roadster Micro Mtn Saddle	Xtreme Adult Helmet
Roadster Micro Mtn Saddle	Xtreme Mtn Lock
Roadster Micro Mtn Saddle	InFlux Lycra Glove

Note: The symbol != is used to represent a Not Equal 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 join.

Using SQL and SQL databases

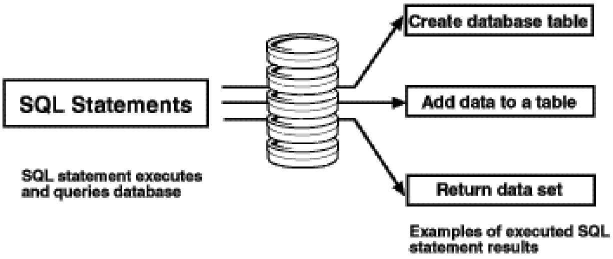
Perhaps the most popular and most powerful database formats are DBMS applications based on the Structured Query Language (SQL). SQL databases usually work over a client/server network architecture, providing an SQL Server to create, store, and manipulate database files, tables, fields and records, and an SQL Client interface allowing workstation users not only to design and work with database files, but also to 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 in 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 in order 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.

SQL is not a true computer language. It cannot be used to create stand-alone computer applications or operating systems. SQL is often referred to as a sub-language, since it can be used within other languages or applications. Most importantly, the purpose of the SQL language is specific to working with relational databases.

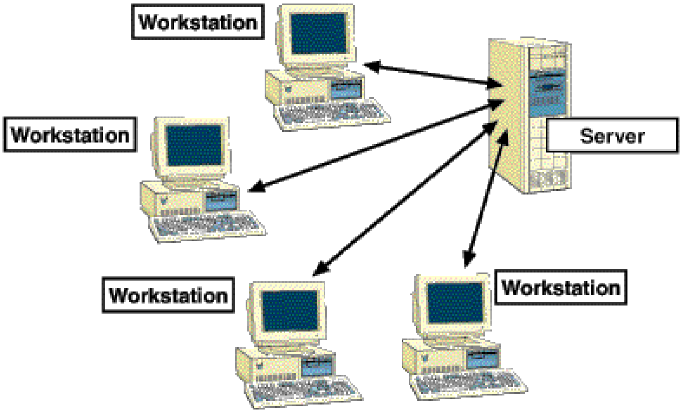
The syntax of the SQL language is built on a system of sending SQL statements to the SQL database server. Each statement is a request to perform a database operation, such as creating a database file, adding tables and fields to a database, adding records to tables, or retrieving data from databases. The SQL server analyzes the SQL statement and performs the required operation. For example, 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.


SQL Statements
SQL statement executes and queries database
Create database table
Add data to a table
Return data set
Examples of executed SQL statement results

An SQL query is an SQL statement designed specifically to request data from one or more SQL databases. Some SQL applications require that you type in an SQL query directly using a text editor, while others provide graphical user interfaces that lead you through the process of querying an SQL database. In the latter case, the application must create an SQL statement based on the information you provide. This statement is the actual SQL query, and it is the SQL query that is used to request the data. Crystal Reports falls into this second category of SQL-compliant applications.

Client/server architecture

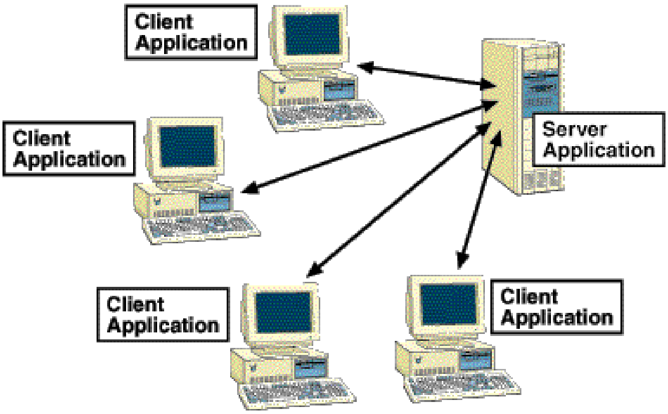
One of the most powerful features of SQL DBMS applications is their ability to efficiently use the client/server architecture of a network.



A simple network structure consists of one or more network servers that provide a common location where all users on the network can 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’s power and resources, while the client application provides an easy-to-use user interface designed to help get the work done faster and better than it could otherwise be done.



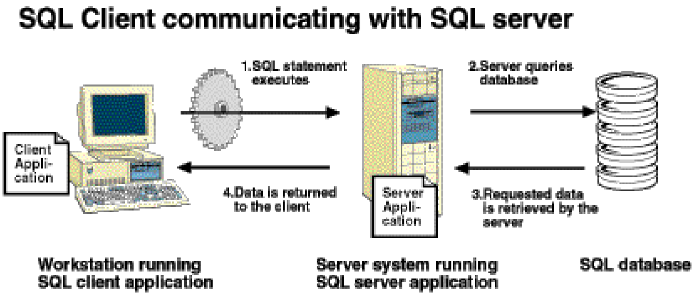
Often, a client/server application will be available with a certain number of seats, depending on the cost of the application. Each seat represents either a single client workstation, or a single client user (depending on the software vendors' specifications) that can be connected to the client/server 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 network server computers. 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, while a network server is a physical machine to which network clients are connected by cables or some other connection device.

SQL DBMS

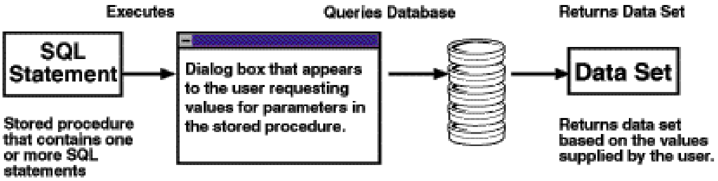
An SQL Database Management System is a common example of a client/server software package. A standard SQL DBMS will include an SQL server application that handles all the actual work of building and working with databases and database data. The DBMS will also include at least one set of SQL client software (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 server returns any data, the client software displays the data to the user.


SQL Client communicating with SQL server
1.SQL statement executes
2.Server queries database
Client Appli-cation
4.Data is returned to the client
Server Appli-cation
3.Requested data is retrieved by the server
Workstation running SQL client application
Server system running SQL server application
SQL database

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 an SQL query that you can 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.


Executes
Queries Database
Returns Data Set
SQL Statement
Dialog box that appears to the user requesting values for parameters in the stored procedure.
Data Set
Stored procedure that contains one or more SQL statements
Returns data set based on the values supplied by the user.

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 an SQL database and use the returned data 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 "Selecting an SQL stored procedure and changing parameters" on page 562.

Note:

- Only one stored procedure can be used per report or subreport (a report containing subreports can have a different stored procedure for the main report and each subreport).
- Stored procedures cannot be used in combination with other database objects.
- Linking is not possible for stored procedures.

How does Crystal Reports use SQL?

When you connect to an SQL database, Crystal Reports acts as an SQL client application, connecting to your SQL server through your network.

When you design a report that accesses SQL data, Crystal Reports builds an SQL query. This query can be edited if you know SQL and you feel that the query can be further optimized. If you choose Show SQL Query from the Database menu, the Show SQL Query dialog box displays the SQL query that Crystal Reports has built.

Note: If you edit your SQL query and then make changes to your report's record selection (or anything else that normally affects an SQL query statement), you will lose your edits and will have to reapply them.

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 the report design into an SQL query, Crystal Reports can off-load much of the report processing onto the server machine. Rather than having to sift through an entire database to find the 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 in order 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 the 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 forces the query to retrieve only unique (distinct) sets of data. When using the DISTINCT clause, a row of results will be retrieved only once. The previous SELECT statement can be modified to use the DISTINCT clause:

```
SELECT DISTINCT
    TABLEA.'CUSTNAME',
    TABLEA.'STATE'
```

FROM

The FROM clause indicates the sources of the database fields specified in the SELECT clause. FROM lists actual database tables that include 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 example illustrates the FROM clause used with the SELECT clause:

```
SELECT
    TABLEA.'CUSTNAME',
    TABLEA.'STATE'
FROM
    ‘TABLEA’ TABLEA
```

WHERE

The WHERE clause has two purposes:

- to specify record selection criteria
- to show how two database tables are joined.

When WHERE is used to specify record selection criteria, it includes a search condition in order 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
```

When WHERE is used to specify how two tables are linked, an SQL join operator sits between the two table names. See “SQL join types (ODBC data sources)” on page 527.

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 specifies that the database records retrieved be sorted according to the values in a specific field. If you do not use the ORDER BY clause, the program retrieves records in the order in which they appear in the original database. If you specify more than one field after the ORDER BY clause, the program sorts the records according to the values in the first field specified, then, within that sort, the program sorts the records 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 with an SQL aggregate function. The server returns only the summarization information for each group to Crystal Reports.

For example:

```
SELECT
    MYTABLE.'STATE',
    MYTABLE.'ZIPCODE',
    SUM (MYTABLE.'SALES')
FROM
    'MYTABLE' MYTABLE
GROUP BY
    MYTABLE.'STATE',
    MYTABLE.'ZIPCODE'
```

Server-side processing

Server-side processing allows you to set up a report that performs the majority of its processing on the server and pushes only relevant details to your computer.

Server-side processing provides you with a number of benefits:

- less time connected to the server
- less memory needed to process the report on your computer
- lower transfer time from the server to the client.

Here’s how server-side processing works: by using SQL pass-through technology to send an SQL statement to the database server and retrieve an initial set of data, Crystal Reports off-loads much of the data retrieval and sorting work onto the server system, thus freeing up local memory and resources for more important tasks. That is why server-side processing works only for reports that have been sorted and grouped; if a report has not been sorted and grouped (for example, if it is a simple list report), then there is no processing to push to the server. You should also note that server-side processing works only for reports based on SQL data sources (you could not, for example, use server-side processing for a report based on a query, since a query is not an SQL data source).

Keep in mind that in order to perform the grouping on the server, your report must conform to the following conditions:

- The Perform Grouping on Server option (in the Options dialog box) is enabled. For more information, see "Enabling server-side processing" on page 543.
- The report uses some form of grouping.
- The report is at least partially hidden (at the very least, the Details section must be hidden). Since the server will process those sections that are hidden, the greater the portion of the report that is visible, the greater the amount of processing that must take place on the client side. If the Details section is shown, server-side processing will not be possible.
- In some cases, formula fields must be processed on the client side. If grouping is based on a formula field, or if a formula is used in a summary field, then all the records must be transferred to the client side before the formula can be evaluated. This will increase the amount of time required to run the report. Therefore, you may want to use SQL expressions as an alternative to formulas.

  Note: Formulas used for record selection are an exception and can be pushed down to the server.
- In order for a report to be processed on the server, any running totals appearing in the report must be based on summary fields (since the data needed for the running totals will be pulled over to the client side).
- In order for a report to be processed on the server, the report must not contain Average or Distinct count summaries.
- The report does not contain specified value grouping.

Note:

- When you drill-down on a hidden section of a report, with the processing being done on the server, connection to the server will be automatically initiated. If the client is disconnected from the server (for example, if you download a report onto your laptop and you work on it from a remote location), then drilling-down on data will produce an error since the database is not available.
- If you save a report that has been partially processed on the server using the Save Data with Report option, the program will save only those records that have been transferred to the client side. In other words, if you have drilled-down on a hidden section and there is a tab for that data in Crystal Reports (indicating the data has been transferred), those records will be saved with the report.

How server-side grouping affects the SQL query

When a report pushes most of its processing to the server, this by necessity alters the SQL query. Thus, when the Perform Grouping on Server option is enabled, individual aspects of server-side processing will modify the SQL statement in different ways.

- If you select Use Indexes Or Server For Speed (in the Report Options dialog box), the program adds an ORDER BY clause to the SQL statement and a WHERE clause for the record selection formula, if possible.
- If you group on a linkable data type in the DBMS, the program adds a GROUP BY clause to the SQL statement. The program uses the GROUP BY clause to perform the grouping on the server.
- If you summarize on a linkable data type, the program adds a summary field to the SELECT clause of the SQL statement.
- If you drill-down on a linkable data type, the program adds a WHERE clause to the SQL statement.
- If you group in descending order on a linkable data type, the program adds an ORDER BY clause to the SQL statement.

The statement also varies between tabs:

- If you are working in the Preview tab, the statement includes the GROUP BY clause, as well as any aggregates that the report is pushing to the server.
- If you are drilling-down, the statement varies depending on the underlying data and the level of drill-down. With each drill-down, the WHERE criteria changes. Also, if you drill-down to the details, the statement will not include a GROUP BY clause (since you no longer have any groups on that drill-down tab).

To view the current SQL statement for the active tab, choose Show SQL Query from the Database menu. The Show SQL Query dialog box appears, displaying the SQL statement.

Note:

- you can use the SQL Expression editor to edit SQL expressions to be processed on the server
- SQL statements that have been edited in the Show SQL Query dialog box disable server-side processing as GROUP BY is no longer pushed down.

Enabling server-side processing

1. On the **File** menu, click **Report Options**.
2. Select **Perform Grouping on Server** on the Report Options dialog box.

   This check box is inactive if Use Indexes or Server for Speed is not selected.
3. Click **OK**.

Note: You can also enable or disable this option quickly by selecting or deselecting Perform Grouping on Server on the Database menu as needed. If Use Indexes or Server for Speed is not selected in the Report Options dialog box, this command is inactive.

Mapping database fields

The Map Fields dialog box allows you to link report fields to their corresponding database fields when you have made changes to the structure of the database, or when you have created a report based on one database from a template of another report based on some other database that has the same table and field structure. In this way, the dialog box helps you to make sure your reports print with the current version of the active database.

When you first create a report, the report draws its fields from the database as it exists at the time. If you change the structure of the database after you create the report, the program needs to adapt the report to the new structure.

About the Map Fields dialog box

The Map Fields dialog box contains four list boxes:

- The upper-left box displays the names of all unmapped report fields (for which the program detects a change in the active database). The top name is selected by default.
- The upper-right box displays the names of unmapped database fields (in which the program detects a change). Since the Match Type box is selected by default, this box displays only the names of unmapped database fields of the same type as the unmapped report field you selected in the upper-left box. To display all unmapped database fields, regardless of type, clear the Match Type box.
- The lower-left box displays the names of mapped report fields. When you map fields in the upper boxes, they appear in the lower boxes.
- The lower-right box displays the names of mapped database fields. When you map fields in the upper boxes, they appear in the lower boxes.

For each database field that you have changed, highlight the report field and the database field in the upper sections and click Map. The field names move from the upper boxes to the lower boxes.

You do not have to remap every report field. For example, if you delete a database field, it is not necessary to remap its corresponding report field, since you will not need that field in your report. You can click OK to close the dialog box without remapping each field listed in the upper-left box.

On some occasions, you might want to unmap or unlink report and database fields that you have already mapped. For example, you might mistakenly map the wrong fields or want to map a report field to a database field that you already mapped. In these cases, you can select a report field in the lower boxes (the program then automatically selects the database field) and click Unmap. The field names will move from the lower boxes to the upper boxes.

Note: Crystal Reports does not automatically refresh the report data when you close the Map Fields dialog box. To refresh your data, click Refresh on the Standard toolbar.

Remapping processes

If you make changes to the active database fields that require you to remap the corresponding report fields, you can display the Map Fields dialog box by choosing any of these commands from the Database Menu:

- **Verify Database**
- **Verify on Every Print**
- **Set Location**
- **Convert Database Driver**.

You can use each of these commands for a specific function; however, any of these commands will open the Map Fields dialog box if the program detects a mismatch between the field names in the report and the field names in the database. In order to detect any possible mismatches, the program checks each field name in the report against the field names in the database. If one of the field names does not match any of the field names in the database, the Map Fields dialog box appears.

Note: Report fields that are left unmapped are removed from the report.

Using the Verify Database process

When you choose Verify Database from the Database menu, the program checks the active databases and reports. If it detects changes, the report must be adapted to prevent errors.

The program displays the Map Fields dialog box when it detects either of these types of changes to the database:

- the name of a database field that is used in the report has changed
- the database has been upsized from a PC data source to an SQL data source.

For a tutorial on the Map Fields dialog box, see "Remapping altered database fields" on page 546. Crystal Reports automatically adapts the report (and does not display the Map Fields dialog box) if it detects any of these changes:

- fields have been added to the database
- fields that are not used in the report have been deleted from the database
- field positions have changed in the database
- data types have changed for fields in the database.

Using the Verify on Every Print process

Verify on Every Print triggers the Verify Database command every time you print your report.

- If there is a check mark beside Verify on Every Print, the option is active. It triggers Verify Database every time you print. See "Using the Verify Database process" on page 545.
- If there is no check mark beside it, the option is inactive. The option is inactive by default.

Using the Set Location process

When you choose Set Location from the Database menu and specify a new location for the active database, the program checks the database for changes.

Set Location displays the Map Fields dialog box when it detects any of the following changes in the database structure:

- a database field has been deleted
- a database field has been renamed
- the database is completely new.

Note: The program checks for these changes only if the databases have different names or if the database name has changed. If the databases have the same name, the Map Fields dialog box does not appear, and you need to verify the database when you have finished setting the location. For more information, see "Using the Verify Database process" on page 545.

Remapping altered database fields

Use the Map Fields dialog box to remap existing report fields in the active database if they have been altered.

To remap an altered database field

1. With the report active in the Design tab, choose **Verify Database** from the **Database** menu.

   The Verify Database message box appears.

   - If the program detects no changes in the active database, the message box displays this message: "The database is up to date." In this case, click OK and return to your work.
   - If the program detects a change(s) in the active database, the message box displays this message: "The database file ["table name"] has changed. Proceeding to fix up the report!"

2. Click **OK**.

   If the program detects that a field name has been altered in the active database, the Map Fields dialog box appears.

   Note: The program automatically adapts the report to changes in other data within the database (number of fields, field position, data type, etc.). It is not necessary to remap fields in which these changes have been made.

3 Highlight the first report field that you want to remap in the upper-left box.

4 In the upper-right box, highlight the unmapped database field to which you want to remap the selected report field.

5 Click **Map**.

The highlighted report and database fields no longer appear in the upper boxes. Instead, they appear in the corresponding lower boxes.

6 Repeat Steps 3 through 5 for each unmapped report field you want remapped.

Note: If the names of any report fields remain in the upper-left box when you exit the dialog box, the program removes them from your report.

7 Click **OK**.

The program remaps the report fields to the altered database fields.

Remapping when converting database drivers

When you choose Convert Database Driver from the Database menu and specify the database driver to use when running your report, the program checks the database for changes. It displays the Map Fields dialog box when it detects changes in the database structure.

Working with databases

This section introduces several common procedures related to accessing database files from within Crystal Reports. Where practical, step-by-step instructions are provided.

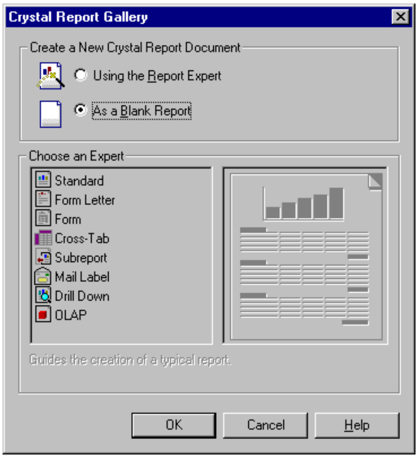
Using Access queries

Opening Access queries through DAO

Microsoft Access queries can be used in Crystal Reports as separate data sets, just like Access tables. When you open an Access database through the DAO engine (see “Microsoft Access” on page 577), any queries in the database can automatically be read.

To open an Access query through DAO

1 On the Standard toolbar, click **New**.
The Crystal Report Gallery appears.



Crystal Report Gallery

Create a New Crystal Report Document

○ Using the Report Expert

● As a Blank Report

Choose an Expert

2 Click **As a Blank Report** and then **OK**.

3 In the Data Explorer dialog box, locate, highlight, and then expand the Microsoft Access database (*.mdb) file that contains the query you want to use in your report.

Note: If your database requires a user name and password, or any other log on information, a log on dialog box appears.

4 Highlight the query and any tables you want to include in your report and click **Add**.

5 Click **Close** when you are finished with the Data Explorer dialog box.
The Field Explorer dialog box appears.

Note: The Visual Linking Expert dialog box appears first if you have selected several queries and tables.

All of the queries and tables you highlighted from your Access database appear in the Field Explorer dialog box.

6 Locate your query in the **Database Fields** list, and double-click its name.
The query expands to display all the fields it contains.

7 Highlight any fields you want to use in your report and add them to the Design tab.

Note: You cannot use Access Action queries or Update queries in Crystal Reports. However, you can use Access Select queries, Cross-tab queries, and Parameter queries. For more information about Parameter queries, see "Opening Access Parameter queries" on page 550.

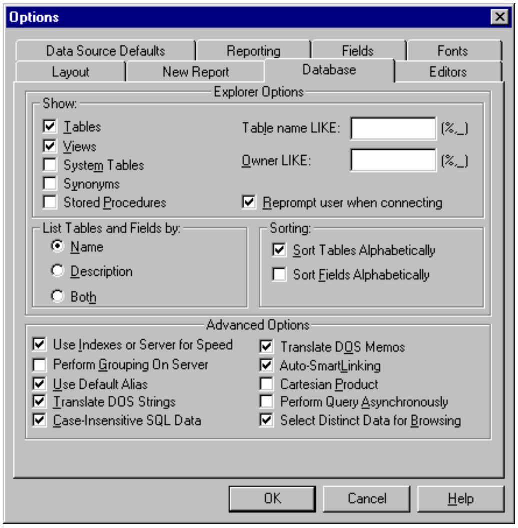
Opening Access queries through ODBC

ODBC gives you more control over which parts of a database you can use. For this reason, using an Access query through ODBC may require a few extra steps.

To open an Access query through ODBC

1 On the **File** menu, click **Options**.

2 Click the **Database** tab.



Explorer Options

Show:

☑ Tables
☑ Views
□ System Tables
□ Synonyms
□ Stored Procedures

Table name LIKE: (%,_)

Owner LIKE: (%,_)

☑ Reprompt user when connecting

List Tables and Fields by:

● Name
○ Description
○ Both

Sorting:

☑ Sort Tables Alphabetically
□ Sort Fields Alphabetically

Advanced Options

☑ Use Indexes or Server for Speed
□ Perform Grouping On Server
☑ Use Default Alias
☑ Translate DOS Strings
☑ Case-Insensitive SQL Data
☑ Translate DOS Memos
☑ Auto-SmartLinking
□ Cartesian Product
□ Perform Query Asynchronously
☑ Select Distinct Data for Browsing

3 Select the **Views** and/or **Reprompt user when connecting** check box(es).

- Selecting "Views" automatically displays any available queries in your Access database.
- Selecting "Reprompt user when connecting" displays the Allow Reporting On dialog box every time you select an ODBC data source. This dialog box provides the same options as the Allow Reporting On section on the SQL tab of the Options dialog box, but provides those options for every ODBC data source you select.

4 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. You can 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% displays only those tables that have a table name beginning with the letter C.

- "Owner LIKE" is also based on the SQL LIKE clause. Owner LIKE allows you to select the Owner (or Creator or Alias) of the table, not the table name itself. For example Owner LIKE C% displays only those tables that have an owner beginning with the letter C.

5 Click **OK** to exit the Options dialog box.

6 Create a new report and choose ODBC as the data source for your Access database.

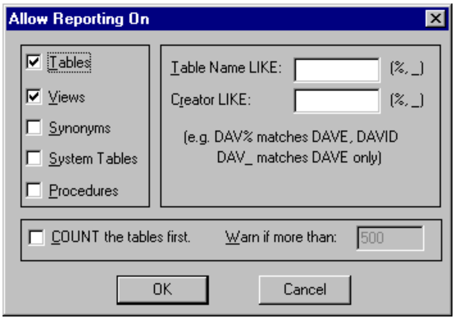
Tip: Choosing ODBC as your data source automatically logs you onto the server.

If you did not specify a particular Access database file with your Access ODBC data source, the Select Database dialog box appears.

Note: If your database requires a user name and password, or any other log on information, a log on dialog box appears.

7 Locate and select the database that contains the Access query you want to use and then expand it.

If you selected the "Reprompt user when connecting" check box in the Options dialog box, the Allow Reporting On dialog box appears. Otherwise, skip to Step 9.



Allow Reporting On

☑ Tables
☑ Views
☐ Synonyms
☐ System Tables
☐ Procedures

Table Name LIKE: (%, _)
Creator LIKE: (%, _)
(e.g. DAV% matches DAVE, DAVID
DAV_ matches DAVE only)

☐ COUNT the tables first. Warn if more than: 500

8 Make sure the **Views** check box is selected and click **OK** when finished.

9 Highlight your query, click **Add** and then **Close**.

The Design tab appears along with the Field Explorer dialog box. Your Access query, and all fields associated with that query, appear under "Database Fields."

Note: You cannot use Access Action queries or Update queries in Crystal Reports. However, you can use Access Select queries and Cross-tab queries.

Opening Access Parameter queries

Access Parameter queries can be opened only when you open an Access database via ODBC. Make sure you have an ODBC data source set up for your Access database before attempting this procedure. See "Setting up an ODBC data source" on page 554.

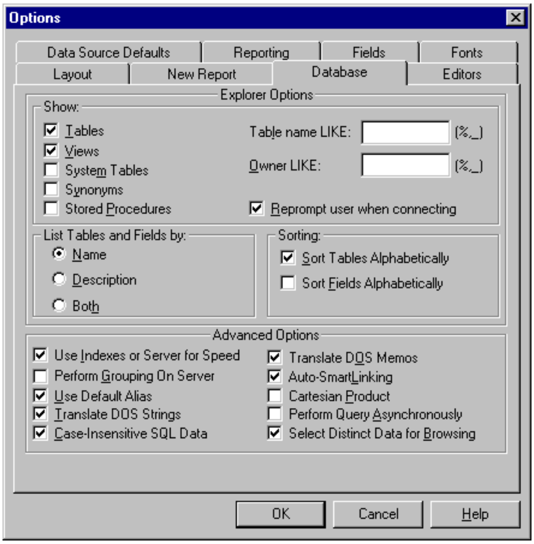
Note: When you design a Parameter query in Access, you must provide a prompt for the query and specify a data type for the parameter. First, with your query open in Design View in Microsoft Access, enter a prompt in the Criteria cell for the field that will act as a parameter. Then, choose Parameters 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.

To open an Access Parameter query

1 In Crystal Reports, choose **Options** from the **File** menu.

The Options dialog box appears.

2 Click the **Database** tab.



Explorer Options

Show:

☑ Tables
☑ Views
☐ System Tables
☐ Synonyms
☐ Stored Procedures

Table name LIKE: (%,_)
Owner LIKE: (%,_)

☑ Reprompt user when connecting

List Tables and Fields by:

● Name
○ Description
○ Both

Sorting:

☑ Sort Tables Alphabetically
☐ Sort Fields Alphabetically

Advanced Options

☑ Use Indexes or Server for Speed
☐ Perform Grouping On Server
☑ Use Default Alias
☑ Translate DOS Strings
☑ Case-Insensitive SQL Data
☑ Translate DOS Memos
☑ Auto-SmartLinking
☐ Cartesian Product
☐ Perform Query Asynchronously
☑ Select Distinct Data for Browsing

3 Select the **Stored Procedures** and/or **Reprompt user when connecting** check box(es).

- Selecting "Stored Procedures" automatically displays any available stored procedures when you log on to an 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 check box must be selected.
- Selecting "Reprompt user when connecting" displays the Allow Reporting On dialog box every time you select an ODBC data source. This dialog box provides the same options as the Allow Reporting On section on the SQL tab of the Options dialog box, but provides those options for every ODBC data source you select.

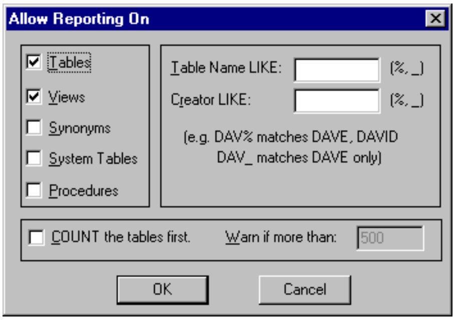
4. In addition, you can specify **Table name LIKE** and **Owner LIKE** options if you wish.
   - "Table name LIKE" is based on the SQL LIKE clause. This option allows you to specify the kinds of table names you want to appear in the Choose SQL Table dialog box. You can use the underscore character (_) 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% displays only those tables that have a table name beginning with the letter C.
   - "Owner LIKE" is also based on the SQL LIKE clause. Owner LIKE allows you to select the Owner (or Creator or Alias) of the table, not the table name itself. For example Owner LIKE C% displays only those tables that have an owner beginning with the letter C.
5. Click **OK** to exit the Options dialog box.
6. Create a new report and choose ODBC as the data source for your Access database.

   Tip: Choosing ODBC as your data source automatically logs you onto the server.

   If you did not specify a particular Access database file with your Access ODBC data source, the Select Database dialog box appears.

   Note: If your database requires a user name and password, or any other log on information, a log on dialog box appears.
7. Locate and select the database that contains the Access Parameter query you want to use and then expand it.

   If you selected the "Reprompt user when connecting" check box in the Options dialog box, the Allow Reporting On dialog box appears. Otherwise, skip to Step 9.



Allow Reporting On

☑ Tables
☑ Views
☐ Synonyms
☐ System Tables
☐ Procedures

Table Name LIKE: (%, _)
Creator LIKE: (%, _)

(e.g. DAV% matches DAVE, DAVID
DAV_ matches DAVE only)

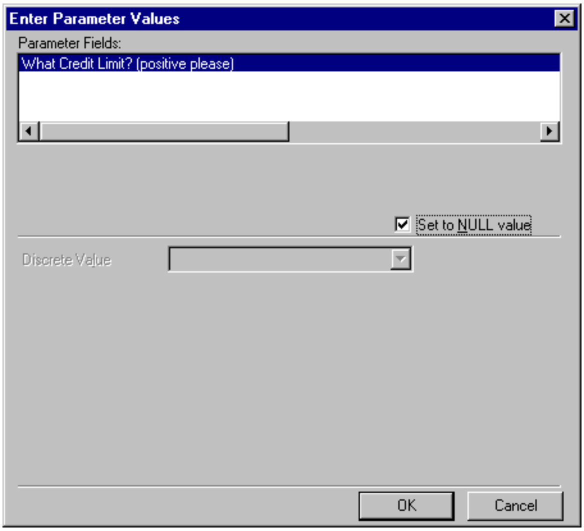
☐ COUNT the tables first. Warn if more than: 500

8 Make sure the **Procedures** check box is selected and click **OK** when finished.

9 Highlight your Parameter query and click **Add**.

Tip: Your Parameter query is found under the Stored Procedures branch of your ODBC data source in the Data Explorer.

The Enter Parameter Values dialog box appears.



10 Highlight a parameter in the **Parameter Fields** list.

Set to NULL Value is selected by default. If you want to assign a value, clear the Set to NULL Value check box.

11 Assign a value by typing into the **Discrete Value** box and then click **OK**.

12 Repeat Steps 10 and 11 for each parameter in your Access Parameter query.

13 On the Data Explorer dialog box, click **Close**.

You can change parameter values at any time by choosing Stored Procedure Parameters from the Database menu.

14 Create your report using the fields in the Parameter query. Only the records that satisfy the parameter values you specified in the Enter Parameter Values dialog box are used in your report.

Note: You cannot use Access Action queries or Update queries in Crystal Reports. However, you can use Access Select queries and Cross-tab queries.

Using ODBC data sources

Setting up an ODBC data source

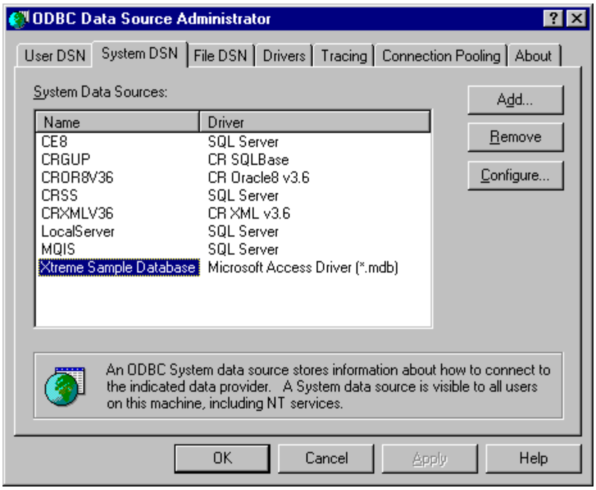
To set up an ODBC data source, you must first have an ODBC driver installed for the type of data you want to use. Many DBMS applications automatically install and set up ODBC drivers. If you are not sure whether ODBC drivers have been installed for your data, refer to the documentation that came with your DBMS application.

Note: Crystal Reports installs version 3.510 of ODBC Administrator. If you have a different version, steps in this procedure will vary slightly.

To set up an ODBC data source

1 From the Program Manager, double-click the ODBC Administrator icon in the appropriate program group.

The ODBC Data Source Administrator dialog box appears.



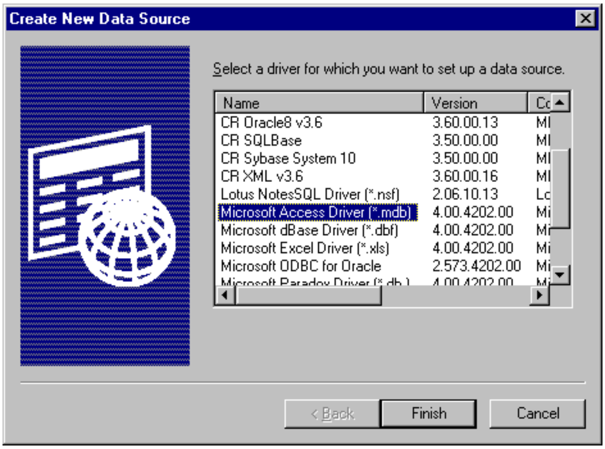
System Data Sources:

Name	Driver
CE8	SQL Server
CRGUP	CR SQLBase
CROR8V36	CR Oracle8 v3.6
CRSS	SQL Server
CRXMLV36	CR XML v3.6
LocalServer	SQL Server
MQIS	SQL Server
Xtreme Sample Database	Microsoft Access Driver (*.mdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

2 Click **Add** to add a new ODBC data source.

The Create New Data Source dialog box appears.



Create New Data Source

Select a driver for which you want to set up a data source.

Name	Version	Cc
CR Oracle8 v3.6	3.60.00.13	MI
CR SQLBase	3.50.00.00	MI
CR Sybase System 10	3.50.00.00	MI
CR XML v3.6	3.60.00.16	MI
Lotus NotesSQL Driver (*.nsf)	2.06.10.13	Lc
Microsoft Access Driver (*.mdb)	4.00.4202.00	Mi
Microsoft dBase Driver (*.dbf)	4.00.4202.00	Mi
Microsoft Excel Driver (*.xls)	4.00.4202.00	Mi
Microsoft ODBC for Oracle	2.573.4202.00	Mi
Microsoft Paradox Driver (*.db)	4.00.4202.00	Mi

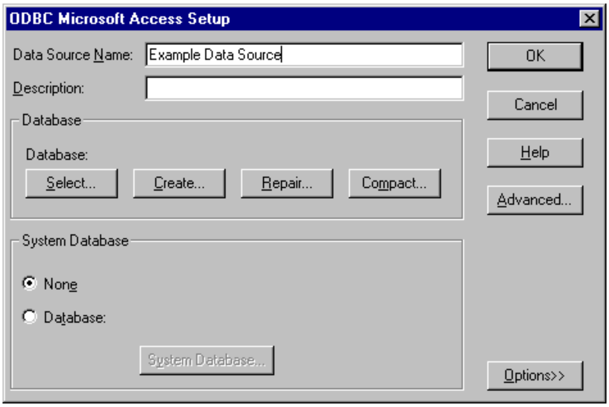
3 Choose the ODBC driver appropriate for your data type from this list.

4 Click **Finish** when finished.

If a driver does not appear for your data type, then the ODBC driver has not been correctly installed. Refer to the documentation for your DBMS application.

An ODBC Data Source Setup dialog box that is specific to the ODBC driver you selected appears.

Note: If an error message appears instead of the Setup dialog box, you may not have the correct ODBC drivers installed on your system for the type of data you selected.



ODBC Microsoft Access Setup

Data Source Name: Example Data Source

Description:

Database

Database:

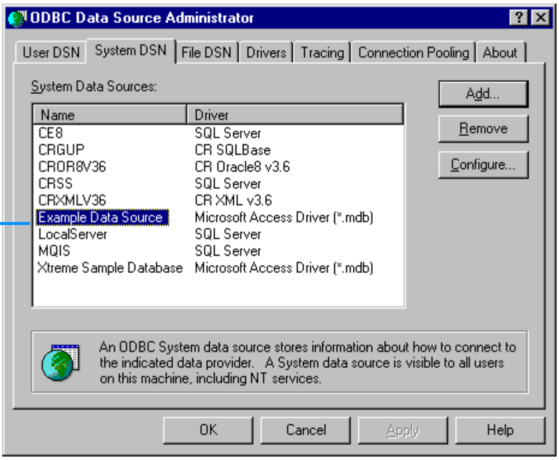
System Database

● None

○ Database:

5 Type the name for your new ODBC data source in the **Data Source Name** box.

Note: The dialog box that appears may look different from the one shown here, depending on the type of data you are using. This dialog box is specific to the Access ODBC driver. For more information about using the dialog box that appears for your data, click Help.



Name	Driver
CE8	SQL Server
CRGUP	CR SQLBase
CROR8V36	CR Oracle8 v3.6
CRSS	SQL Server
CRXMLV36	CR XML v3.6
Example Data Source	Microsoft Access Driver (*.mdb)
LocalServer	SQL Server
MQIS	SQL Server
Xtreme Sample Database	Microsoft Access Driver (*.mdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

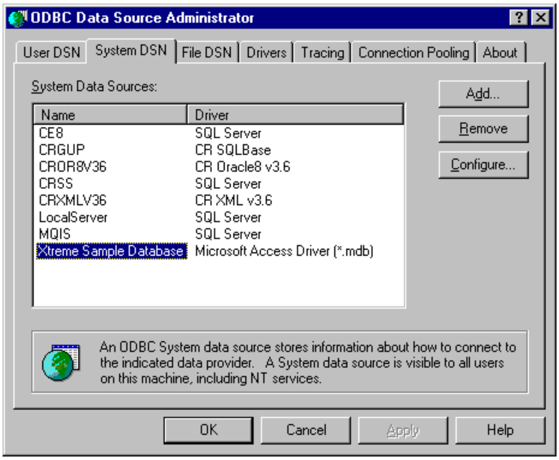
The new ODBC data source now appears in this list.

6 Click **OK** when finished.

Checking settings for an ODBC data source

1 From the appropriate program group or folder, run the ODBC Administrator application.

The ODBC Data Source Administrator dialog box appears.



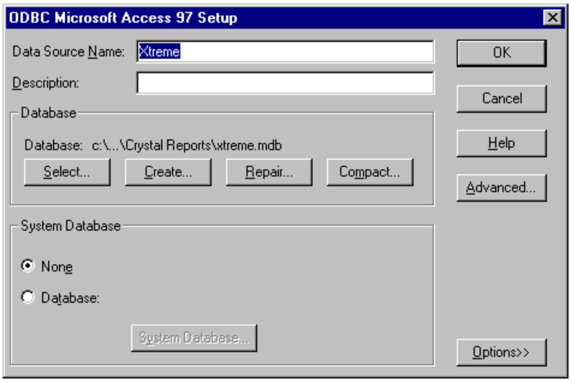
Name	Driver
CE8	SQL Server
CRGUP	CR SQLBase
CROR8V36	CR Oracle8 v3.6
CRSS	SQL Server
CRXMLV36	CR XML v3.6
LocalServer	SQL Server
MQIS	SQL Server
Xtreme Sample Database	Microsoft Access Driver (*.mdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

2 Highlight the appropriate data source from the **User Data Sources** list (on the **User DSN** tab).

3 Click **Configure**.

The ODBC Data Source Setup dialog box appears.



ODBC Microsoft Access 97 Setup

Data Source Name: Xtreme

Description:

Database

Database: c:\...\Crystal Reports\xtreme.mdb

System Database

● None

○ Database:

The ODBC Data Source Setup dialog box that appears is specific to the data source you selected, and contains controls and information for setting up your data source.

4 Check the settings in this dialog box to make sure the information matches your system and database.

5 Make any changes necessary, and then click **OK**.

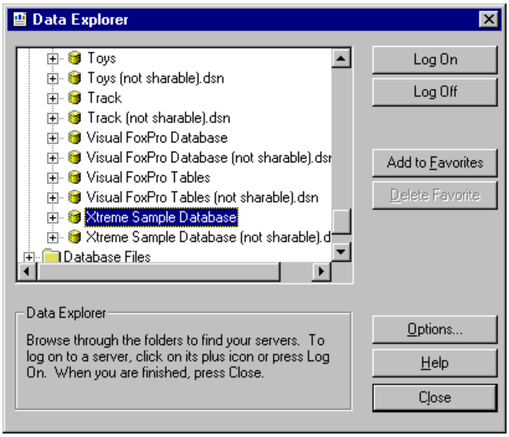
6 Click **OK** to close the ODBC Data Source Administrator dialog box.

Logging on to an ODBC data source

1 Choose **Log On/Off Server** from the **Database** menu.

If no report is open, choose Log On/Off Server from the File menu.

The Data Explorer dialog box appears.



Data Explorer

- Toys
- Toys (not sharable).dsn
- Track
- Track (not sharable).dsn
- Visual FoxPro Database
- Visual FoxPro Database (not sharable).dsr
- Visual FoxPro Tables
- Visual FoxPro Tables (not sharable).dsn
- Xtreme Sample Database
- Xtreme Sample Database (not sharable).d
- Database Files

Data Explorer

Browse through the folders to find your servers. To log on to a server, click on its plus icon or press Log On. When you are finished, press Close.

2 Choose the ODBC database file you want to open and click **Log On**.

If the data source requires a user name and password, or any other log on information, a log on dialog box appears.

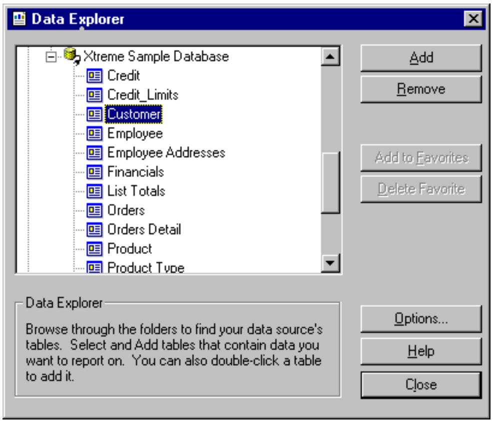
3 Type in the log on information you usually use to access this database, and click **OK**.

If you did not specify a database with the ODBC data source, the Select Database dialog box appears. Select the database file and then click OK.

Adding an ODBC database table to a report

1 Choose **Add Database to Report** from the **Database** menu.

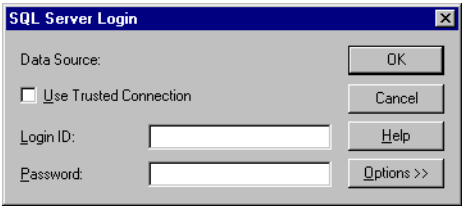
The Data Explorer dialog box appears.



Browse through the folders to find your data source's tables. Select and Add tables that contain data you want to report on. You can also double-click a table to add it.

2 Choose the ODBC database file you want to open and expand it by double-clicking it.

If the data source requires a user name and password, or any other logon information, the Login dialog box appears.



3 Type in the log on information you usually use to access this database, and click **OK**.

If you did not specify a database with the ODBC data source, the Select Database dialog box appears. Select the database file and then click OK.

4 Highlight a database table and then click **Add**.

5 Click **Close** when you are finished with the Data Explorer dialog box.

6 Ensure the tables are linked in the Visual Linking Expert, then click **OK**.

Logging on to MS SQL Server via ODBC

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 any other password-protected data source may require different steps.

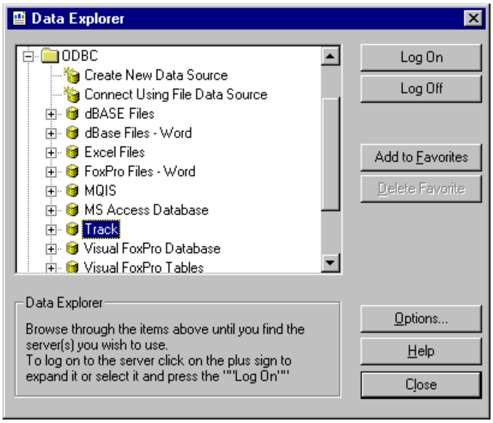
To log on to MS SQL Server via ODBC

1 Verify the settings for the Microsoft SQL Server data source using ODBC Administrator. See "Checking settings for an ODBC data source" on page 556.

2 From Crystal Reports, choose **Log On/Off Server** from the **Database** menu.

If no report is open, choose Log On/Off Server from the File menu.

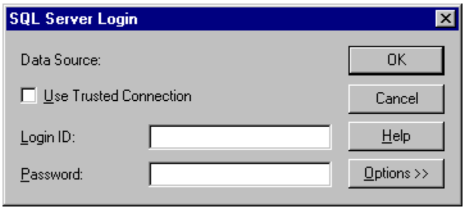
The Data Explorer dialog box appears.



Browse through the items above until you find the server(s) you wish to use.
To log on to the server click on the plus sign to expand it or select it and press the ""Log On""

Choose a SQL Server data source and then click **Log On**. The CRSS data source is automatically created in the ODBC folder during installation. This data source allows you to open MS SQL Server databases.

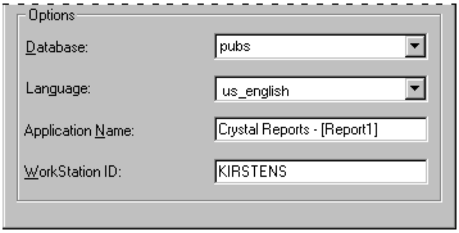
3 The SQL Server Login dialog box appears.



4 Type your MS SQL Server login ID and password.

5 Click **Options**.

The Options section of the SQL Server Login dialog box appears.



Options

Database: pubs

Language: us_english

Application Name: Crystal Reports - [Report1]

WorkStation ID: KIRSTENS

6 Verify the settings for your SQL Server.

7 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 necessary.

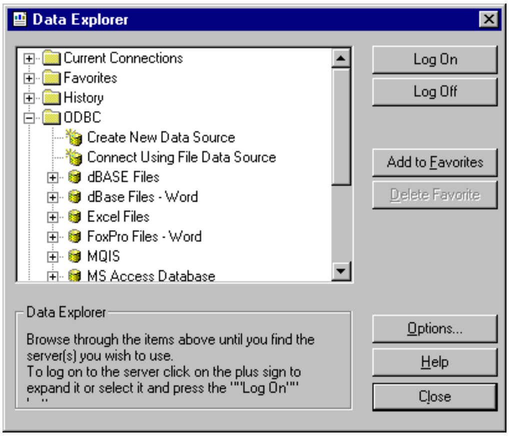
8 Click **OK**.

Logging off an ODBC data source

1 Choose **Log On/Off Server** from the **Database** menu.

If no report is open, choose Log On/Off Server from the File menu.

The Data Explorer dialog box appears.



Data Explorer

Browse through the items above until you find the server(s) you wish to use.
To log on to the server click on the plus sign to expand it or select it and press the ""Log On""

2 Highlight the ODBC data source you want to log off from this list.

3 Click **Log Off**.

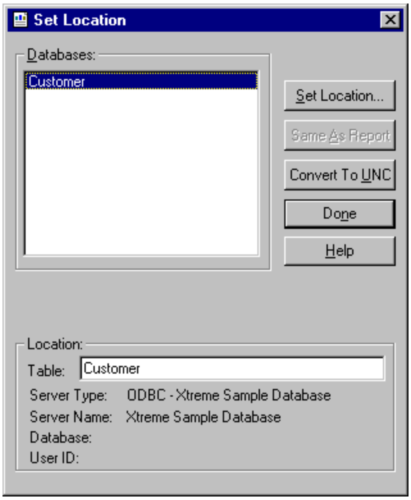
4 Click **Close** when finished with the Data Explorer dialog box.

Changing the ODBC data source accessed by a report

Tip: Another way to change a data source is to use the Convert Database Driver option found on the Database menu.

1 Choose **Set Location** from the **Database** menu.

The Set Location dialog box appears with a list of table aliases for the tables displayed in the report and will display the locations and logon information for the table you select below the list of tables.



Databases:
Customer

Location:
Table: Customer
Server Type: ODBC - Xtreme Sample Database
Server Name: Xtreme Sample Database
Database:
User ID:

2 Highlight the first table in this list box.

3 Click **Set Location**.

4 Choose the original ODBC data source to log on if you are not logged on already. You must log on to the old data source before you can modify 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 you would log on to 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 in the ODBC Data Source Administrator (found in the Control Panel), and make it point to the new server.

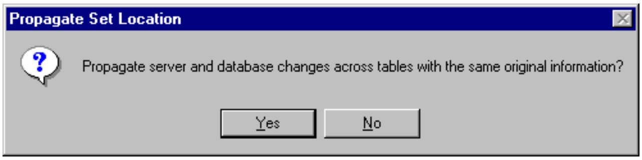
After you log on to the old data source, double-click it so the list of tables appears in the Data Explorer dialog box.

5 Choose the new data source within the Data Explorer dialog box to log on.

You should now be logged on to both the old and the new data sources.

6 Double-click the new data source to display its tables, then select the appropriate table from this list. The table you choose should correspond to the first table you selected in the Set Location dialog box.

7 After clicking **Set** the program prompts you to change the location of all tables in the report to the location you specified for the first table.



Propagate Set Location

Propagate server and database changes across tables with the same original information?

8 Click **Yes** to set the location of all tables in the report to the same database.

9 Close the Set Location dialog box, and choose **Verify Database** from the **Database** menu to refresh the table definitions in the report.

10 Log off the old data source.

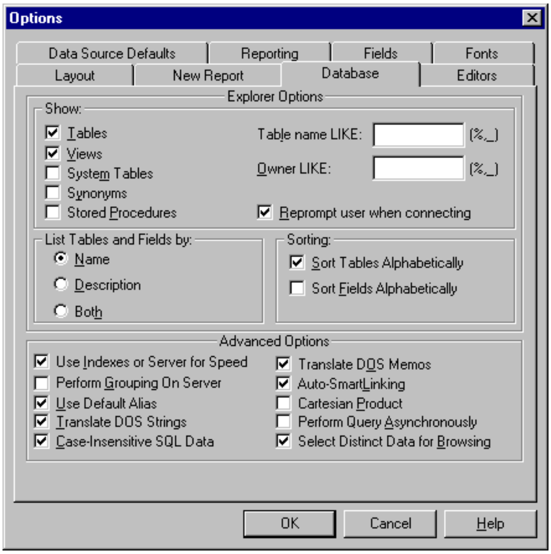
11 Save the report.

SQL stored procedures

Selecting an SQL stored procedure and changing parameters

1 Choose **Options** from the **File** menu.

The Options dialog box appears.



Options

Explorer Options

Show:

☑ Tables
☑ Views
☐ System Tables
☐ Synonyms
☐ Stored Procedures

Table name LIKE: (%,_)
Owner LIKE: (%,_)
☑ Reprompt user when connecting

List Tables and Fields by:
● Name
○ Description
○ Both

Sorting:
☑ Sort Tables Alphabetically
☐ Sort Fields Alphabetically

Advanced Options

☑ Use Indexes or Server for Speed
☐ Perform Grouping On Server
☑ Use Default Alias
☑ Translate DOS Strings
☑ Case-Insensitive SQL Data
☑ Translate DOS Memos
☑ Auto-SmartLinking
☐ Cartesian Product
☐ Perform Query Asynchronously
☑ Select Distinct Data for Browsing

2 Click the **Database** tab and make sure either the **Stored Procedures** or the **Reprompt user when connecting** check box is selected.

- Selecting "Stored Procedures" automatically displays any available stored procedures when you log on to an SQL database.
- Selecting "Reprompt user when connecting" displays the Allow Reporting On dialog box every time you select an ODBC data source. This dialog box provides the same options as the Explorer Options section on the Database tab of the 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. You can 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 have an owner beginning with the letter C.

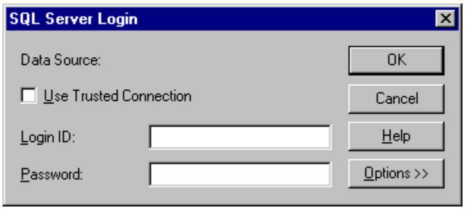
Note: For more information on the SQL LIKE clause, refer to your SQL documentation.

4 Click **OK** to exit the Options dialog box.

5 Create a new report based on the SQL database that contains the stored procedure you want to use.

6 Double-click the ODBC data source for your SQL Server from the Data Explorer dialog box.

The SQL Server Login dialog box appears.



SQL Server Login

Data Source:

☐ Use Trusted Connection

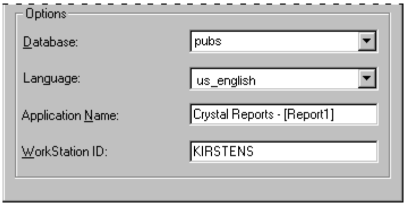
Login ID:

Password:

7 Enter the required information to log in.

8 Click **Options**.

The Options section of the SQL Server Login dialog box appears.



Database: pubs

Language: us_english

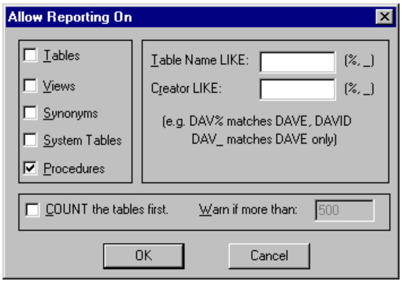
Application Name: Crystal Reports - [Report1]

WorkStation ID: KIRSTENS

9 Verify the settings for your SQL Server.

10 Click **OK**.

If you toggled on the “Reprompt user when connecting” check box in the Options dialog box, the Allow Reporting On dialog box now appears. Otherwise, skip to Step 12.



☐ Tables

☐ Views

☐ Synonyms

☐ System Tables

☑ Procedures

Table Name LIKE: (%, _)

Creator LIKE: (%, _)

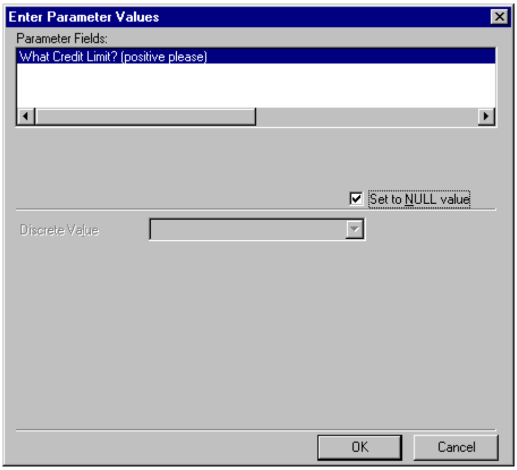
(e.g. DAV% matches DAVE, DAVID
DAV_ matches DAVE only)

☐ COUNT the tables first. Warn if more than: 500

11 Make sure the **Procedures** check box is selected and click **OK** when finished.

12 Highlight an SQL stored procedure and click **Add**.

The Enter Parameter Values dialog box appears.



Parameter Fields:

What Credit Limit? (positive please)

☑ Set to NULL value

Discrete Value

Note: Under the native driver for MS SQL Server, stored procedure input parameters are limited to 64 characters, rather than 255.

13 Highlight a parameter in the **Parameter Fields** list.

Set to NULL Value is selected by default. If you want to assign a value, clear the Set to NULL Value check box.

14 Assign a value by typing into the **Discrete Value** box and then click **OK**.

15 Repeat Steps 13 and 14 for each parameter that appears in the **Parameter Fields** list.

16 On the Data Explorer dialog box, click **Close**.

You can change parameter values at any time by choosing Stored Procedure Parameters from the Database menu.

17 Create your report using the fields from the stored procedure. Only the records that satisfy the parameter values you specified in the Enter Parameter Values dialog box are used in your report.

Advanced database features

One-to-many links

This tutorial demonstrates how to set up an A to B, A to C report using the Customer, Credit, and Orders tables in the Xtreme.mdb sample database. The Xtreme.mdb database is located in the `\Program Files\Seagate Software\Crystal Reports\Samples\En\Databases` directory. Use the instructions here as a guideline for creating A to B, A to C reports with your own database files.

To set up an A to B, A to C link

1 On the Standard toolbar, click **New**.

2 Click **As a Blank Report** and then **OK** on the Crystal Report Gallery.

3 Expand the **Database Files** folder.

4 Double-click the Xtreme.mdb database file to see its tables.

5 Highlight the Credit table and click **Add**.

6 Repeat the previous step for both the Customer and Orders tables.

7 When all three tables have been added, click **Close** on the Data Explorer dialog box.

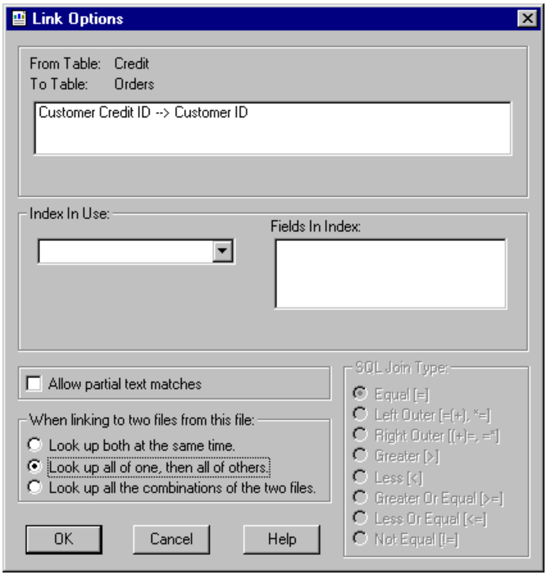
The Visual Linking Expert appears.

8 If the tables are linked by default, click **Clear Links**.

9 Link the **Customer Credit ID** field in the **Credit** table to the **Customer ID** field in each of the other two tables.

10 Select one of the links in the Visual Linking Expert and click **Link Options**.

The Link Options dialog box appears.



From Table: Credit
To Table: Orders

Customer Credit ID --> Customer ID

Index In Use:

Fields In Index:

☐ Allow partial text matches

When linking to two files from this file:

○ Look up both at the same time.
● Look up all of one, then all of others.
○ Look up all the combinations of the two files.

SQL Join Type:

● Equal [=]
○ Left Outer [=(+), *=]
○ Right Outer [(+)=, =*]
○ Greater [>]
○ Less [<]
○ Greater Or Equal [>=]
○ Less Or Equal [<=]
○ Not Equal [!=]

Note: Only the link options on the left of the dialog box apply to native connections.

11 Click **Look up all of one, then all of others** 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.

12 Click **OK** to return to the Visual Linking Expert.

The option you selected affected all links.

13 To make sure the correct link option is in effect, click the link line that you did not select in the last step.

14 Click **Link Options**, and verify the **Look up all of one, then all of others** setting in the Link Options dialog box.

15 Click **OK** in the Link Options dialog box, then click **OK** again back in the Visual Linking Expert.

For more information about this type of linking, see "Look up all of one, then all of others (A to B, A to C)" on page 526.

You have now established an A to B, A to C relationship among the three tables.

SQL queries

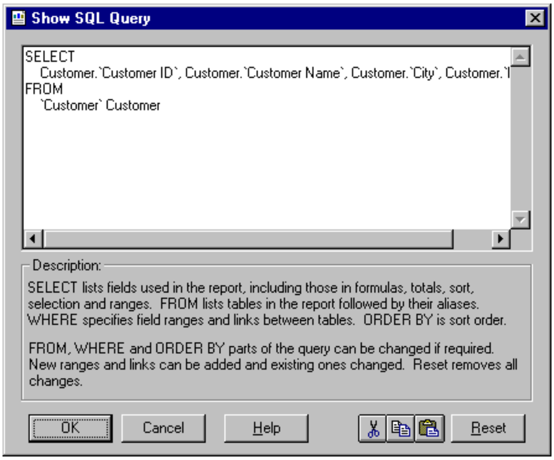
This section is only valid for reports using ODBC data sources. Crystal Reports automatically generates an SQL query when you design a report based on one or more ODBC data source. This query is sent to the ODBC data source(s) as an instruction to gather the data needed by the report.

Note: It is not recommended that you edit SQL queries in the manner described below unless you have experience with the SQL language. To create a customized query, see "The Crystal SQL Designer" on page 621.

To edit an SQL query

Tip: Open a report based on an ODBC data source to follow this procedure.

1 On the **Database** menu, click **Show SQL Query**.
The Show SQL Query dialog box appears.



```
SELECT
   Customer.`Customer ID`, Customer.`Customer Name`, Customer.`City`, Customer.`I
FROM
   `Customer` Customer
```

Description:

SELECT lists fields used in the report, including those in formulas, totals, sort, selection and ranges. FROM lists tables in the report followed by their aliases. WHERE specifies field ranges and links between tables. ORDER BY is sort order.

FROM, WHERE and ORDER BY parts of the query can be changed if required. New ranges and links can be added and existing ones changed. Reset removes all changes.

2 Click anywhere inside the text area to begin making changes.

3 Use the ANSI SQL language to fine-tune the SQL query.

Note: You cannot change the SELECT clause of the SQL statement.

You can cut, copy, or paste any part of the query to or from the Clipboard if needed. For more information, see "The SQL language" on page 539.

4 Click **OK** when finished.

ACT! databases

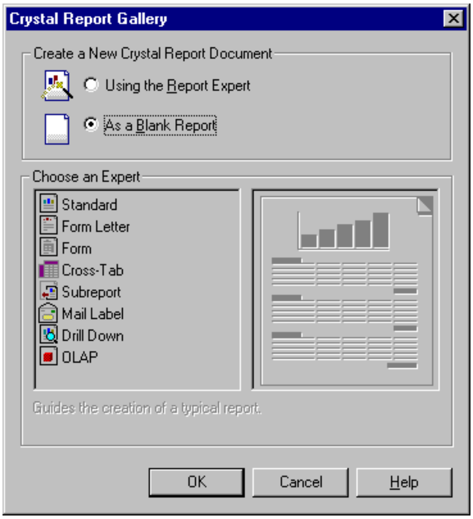
ACT! is a powerful Contact Management application that stores all your contact information in a database format similar to xBASE databases (dBASE, Clipper, and FoxPro).

Note: Versions of ACT! before and including version 2 require the use a file called Crw.act which is not needed in versions 3 and 4. Choose the appropriate procedure based on the version of ACT! you are using.

To use an ACT! 2 database

1 On the Standard toolbar, click **New**.

The Crystal Report Gallery appears.



Crystal Report Gallery

Create a New Crystal Report Document

○ Using the Report Expert

● As a Blank Report

Choose an Expert

- Standard
- Form Letter
- Form
- Cross-Tab
- Subreport
- Mail Label
- Drill Down
- OLAP

Guides the creation of a typical report.

2 Click **As a Blank Report** and then **OK**.

3 In the Data Explorer dialog box, expand the **Database Files** folder and double-click **Find Database File**.

4 Use the Open dialog box to locate and highlight the file Crw.act. By default, this file is installed in the Windows or Winnt System 32 directory. Click **Open** when finished.

The Choose File for ACT dialog box appears.

5 Use this dialog box to locate and highlight your ACT! database.

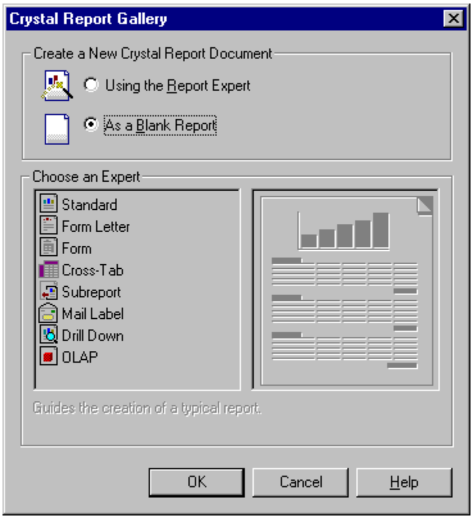
6 Click **OK**.

The Design tab appears in the application window. Create your report using fields from your ACT! database.

To use an ACT! 3 or 4 database

1 On the Standard toolbar, click **New**.

The Crystal Report Gallery appears.



Crystal Report Gallery

Create a New Crystal Report Document

○ Using the Report Expert

● As a Blank Report

Choose an Expert

2 Click **As a Blank Report** and then **OK**.

3 In the Data Explorer dialog box, expand the **Database Files** folder and double-click **Find Database File**.

4 Use the Open dialog box to locate and highlight your ACT! database. Click **Open** when finished.

The database you chose appears in the Data Explorer dialog box.

5 Highlight the table you want to use in your report and click **Add**.

6 Click **Close** on the Data Explorer.

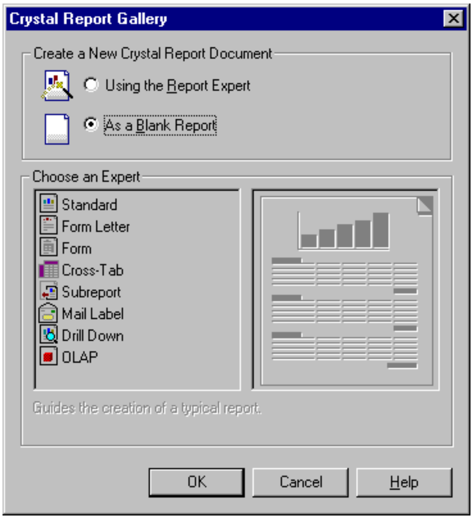
The Design tab appears in the application window. Create your report using fields from your ACT! database.

NT Event Logs

If you are running on a Windows NT machine, Crystal Reports gives you the ability to generate reports based on the NT Event Log.

To report on an NT Event Log

1 On the Standard toolbar, click **New**.
The Crystal Report Gallery appears.



Crystal Report Gallery

Create a New Crystal Report Document

○ Using the Report Expert

● As a Blank Report

Choose an Expert

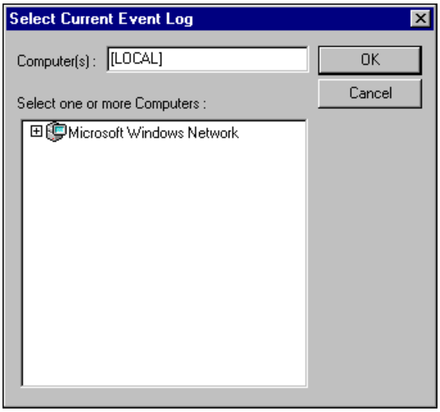
2 Click **As a Blank Report** and then **OK**.

3 In the Data Explorer dialog box, expand the **More Data Sources** tree.

4 Double-click either **NT Archived Event Log** or **NT Current Event Log**.

- Choose NT Archived Event Log to report on a previously archived NT Event Log. If you choose this option the Select Archived NT Event Log dialog box appears, allowing you to navigate to the file you want to report on.
- Choose NT Current Event Log to report on the most recent NT Event Log.

5 If you chose to report on the **NT Current Event Log**, the Select Current Event Log dialog box appears:



6 The computer referenced in the **Computer(s)** box is the computer on which the sample report is based. To change the computer, expand the **Microsoft Windows Network** tree.

The directory structure expands to display the network group to which your computer is connected.

7 Expand the network group to display individual machines connected to the group.

8 Select the NT machine for which you would like to create an NT Event Log Report.

When you select a machine, that name appears in the Computer(s) box.

Three tables become available on the Data Explorer dialog box:

- Application
- Security
- System.

Use these tables to create your report based on the NT Current Event Log.

Note:

- In order to report on the Security table, you need to have the User Right of “Manage auditing and security log” added to your Policy in NT.
- The data that you will see is the same data that is seen from the NT Event Viewer.

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 database topics, refer to the documentation provided with your DBMS application.

Note: 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.

Accessing Data Sources 28

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

Introduction

Crystal Reports can access data stored in almost any common database format, as well as many uncommon formats. In addition, Crystal Reports leverages the full benefits of query (.QRY) files and dictionaries (.DC5). See "The Crystal SQL Designer" on page 621, and "Dictionaries" on page 639.

This section 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 your network administrator.

The principal purpose of Crystal Reports is simply to access data stored in databases and to produce reports on that data. This goal is one of the oldest uses of computers and remains one of the most common and most 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. By understanding how Crystal Reports accesses data, you will gain a better knowledge of the reporting process, as well as a better knowledge of the type of data with which the program can work.

In addition, understanding the data access process will help you troubleshoot problems you may encounter while trying to open a particular database file. This is especially useful for IS managers, or for anyone who provides 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 95 and Windows 98, or Windows NT operating system; and at least one DBMS application is assumed.

Note: Many of the translation files mentioned in this chapter have slightly different names when viewed in the Convert Database Driver dialog box. For example, the translation file normally called P2sodbc.dll appears as Pdsodbc.dll in the Convert Database Driver dialog box. Both of these names represent the same file.

Four types of data

The data that Crystal Reports can access falls into five general categories:

- "Direct access database files" on page 575
- "ODBC data sources" on page 602
- "OLE DB" on page 616
- "Crystal SQL Designer files" on page 617
- "Crystal Dictionary files" on page 618.

Each type of data must be accessed using a specific set of Dynamic Link Libraries (DLLs) and other data access-related files. Once you understand the process the program uses to access each type of data, you will have a better understanding of the report creation process and the elements used to create powerful reports with your data. See "The Crystal SQL Designer" on page 621, and "Dictionaries" on page 639.

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, the program has the built-in capabilities needed 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 the program is installed on your system, you can immediately begin creating reports based on these databases by selecting the appropriate file.

Advantages

Accessing the database directly is the fastest way to read 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 will read all the stored data.

Disadvantages

When you access a database directly through 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.

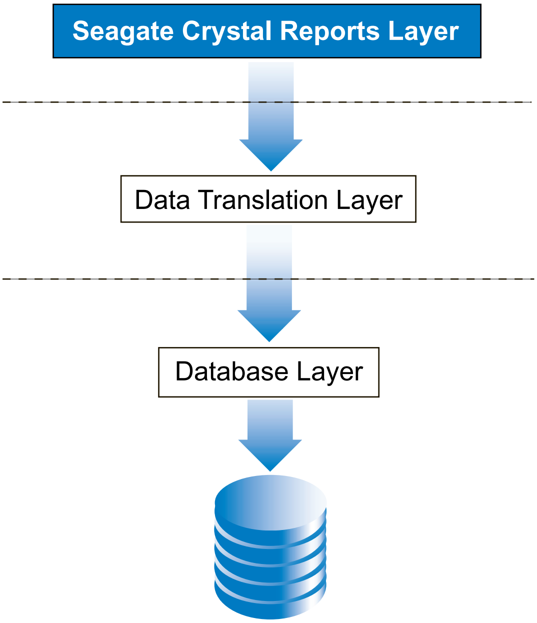
For example, if you design a report based on Btrieve data, you can not change the tables accessed by the report to Access data. Crystal Reports communicates with Btrieve data using Btrieve-specific syntax, a syntax that is not compatible with Access data.

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 data sources" on page 602.

Three layers

Direct access of database files from Crystal Reports requires three layers:

- "Crystal Reports" on page 576
- "Data translation" on page 576
- "Database" on page 577.



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 can 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 in order 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. The program 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 the 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 online help (Runtime.hlp).

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 the program will make them available to you. In other words, when a dBASE file is opened, only 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, as with table fields. See "DAO" on page 579 and "Access" on page 608.

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 DLLs. However, some formats expand the basic three-tiered structure.

The following sections cover the systems used by Crystal Reports to access data from some of the most popular database formats.

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 may depend on how your data is set up. Below is a description of how to open Access data from Crystal Reports through the DAO engine. Another technique uses Microsoft's Open Database Connectivity (ODBC) standard, and is described in Access, page 608.

Note:

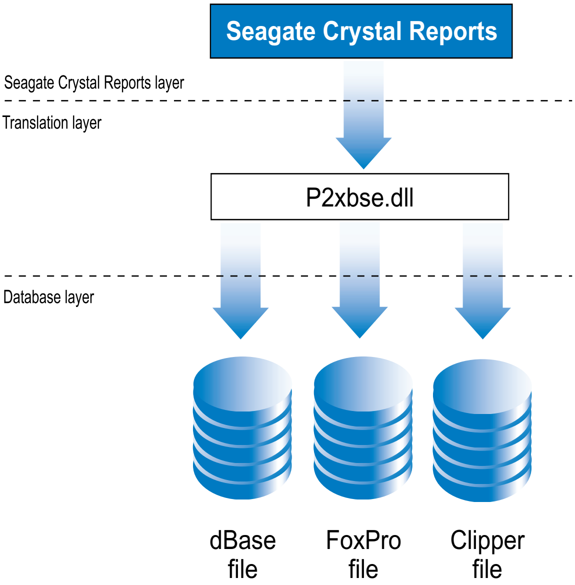
- You can maintain Access security when opening database files with Crystal Reports by running Sysdb32.exe and select the Access Systm.mdw file. In this case, the program will look to Systm.mdw when opening an Access database and the user will be prompted for the Access password (as required) in Crystal Reports.
- When you open an Access database using the 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). This restricts the size of your Access database to about 80 tables, depending on the number of fields in your tables.

dBASE, FoxPro, Clipper

Crystal Reports has been designed to open dBASE data simply and directly through the xBase engine (inside P2bxbse.dll). FoxPro and Clipper are dBASE compatible database formats, and Crystal Reports uses the same DLL to access files created by any of these three DBMS applications.

Note: The P2bxbse.dll translation layer supports FoxPro files up through version 2.6. See "Visual FoxPro™" on page 613 for versions after 2.6.

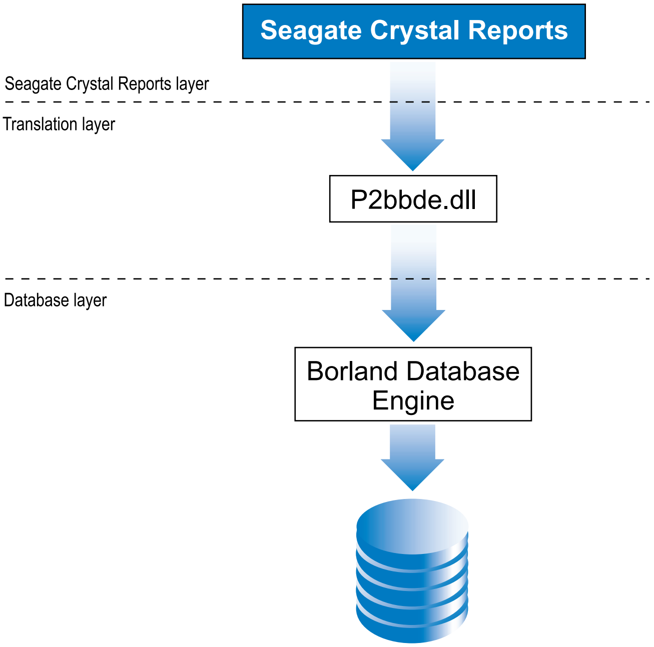


The file P2bxbse.dll handles all translation between Crystal Reports and the dBASE, FoxPro, or Clipper files. Each database file contains only a single database table, but there is no limit to the number of files that can be accessed by a report.

Note: dBASE data can also be accessed through the Borland Database Engine (BDE) using the translation file P2bbde.dll. To see how the BDE communicates with database data, see "Paradox" on page 579. The BDE, however, does not support FoxPro or Clipper data.

Paradox

Files created with Paradox (.DB) are made available to other applications through the Borland Database Engine (BDE). 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 P2bbde.dll translation file.



DAO

Microsoft's Data Access Object™ (DAO) Engine for versions of Access 2.0 and later 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 DAO engine. It acts as a gateway through which Access data is made available to applications. Since the DAO engine is so closely tied to Access data, Crystal Reports considers it a part of the actual database.

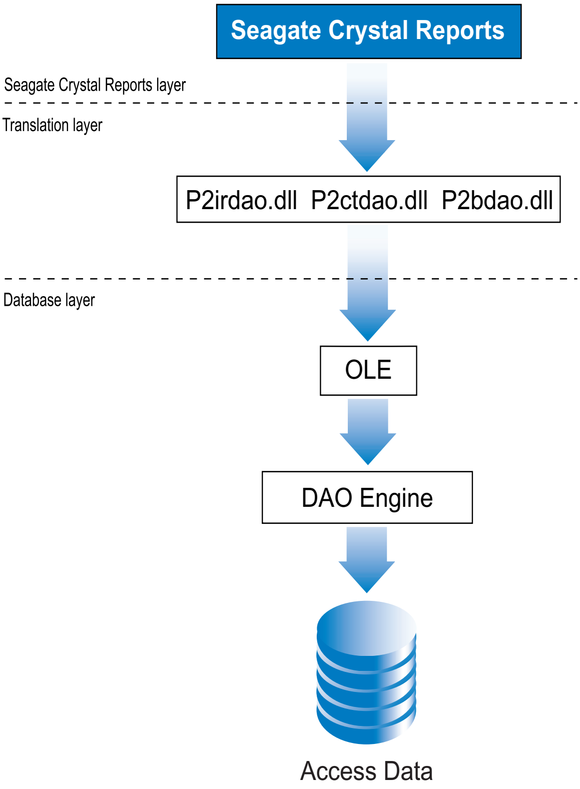
DAO uses Microsoft's Object Linking and Embedding (OLE) technology (installed with Windows 95, Windows 98, and Windows NT) to provide access to Access data through an object-oriented approach.

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 also "Opening Access queries through DAO" on page 547.

Note:

- Access Parameter queries and Cross-tab queries can only be opened through an ODBC connection. See "Opening Access queries through DAO" on page 547. Access Action queries are not supported by Crystal Reports.
- When opening Access queries in a report, make sure the Views and Stored Procedures options are toggled on using the Database tab of the Options dialog box in Crystal Reports. This ensures that the queries will be visible when you open the Access database.

To translate information and data to and from DAO, the Report Designer uses the DAO translation files P2bdao.dll, P2ctdao.dll, and P2irdao.dll.



Secured Microsoft Access Databases

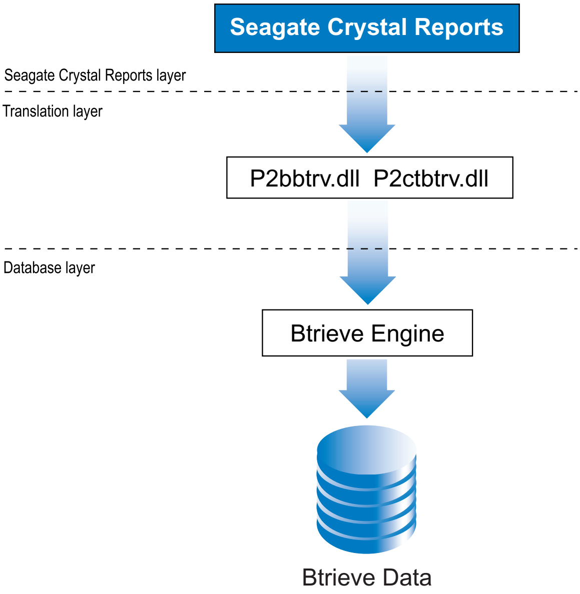
If you will be using secured Access databases, the SystemDB parameter in the Windows Registry database must be set to point at the path where the System.mda or System.mdw (Access 95) file is located. On Windows 95, Windows 98, and Windows NT systems, the SystemDB parameter is located in the following Registry key:

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\Seagate Software\Jet\3.0\ Engines\Jet
```

Crystal Reports includes a utility called Sysdb32.exe to take care of setting this parameter for you. Simply run the appropriate utility and point it at the location of the System.mda or System.mdw file.

Btrieve

Crystal Reports connects to the Pervasive engine through the translation files P2bbtrv.dll and P2ctbtrv.dll.



The Btrieve engine is a complex collection of DLLs and EXEs that are specific to the version of Btrieve you are using. For more information on the different Btrieve engine files, 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 immediately upon installation. Crystal Reports automatically installs the Btrieve files that it requires to read Btrieve data. Refer to your Btrieve documentation to make sure the Btrieve engine is configured correctly.
- When you open a Btrieve database, Crystal Reports opens the entire database and loads information about all the tables in 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). 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 one of the DDFs is selected when creating a new report, Crystal Reports immediately adds all the data files defined in that DDF 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 on configuring the Btrieve engine.

Exchange

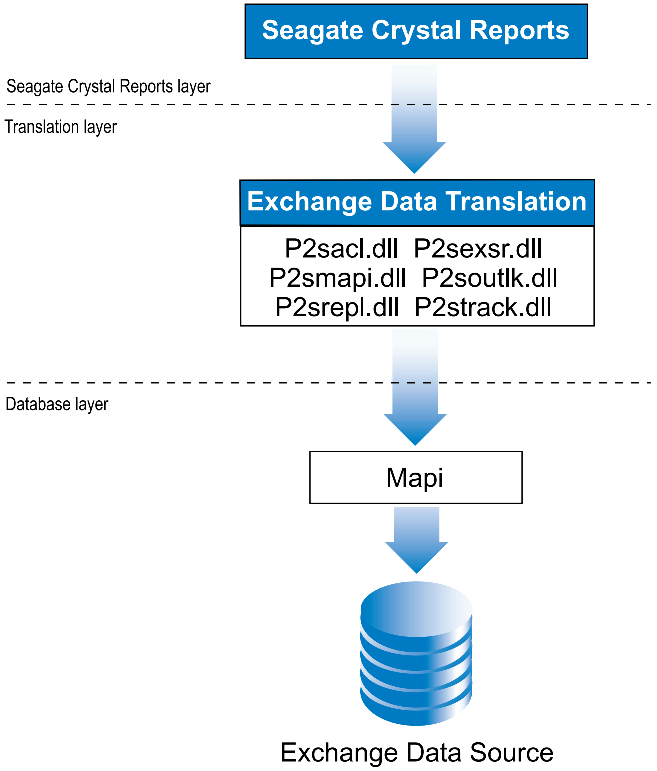
Exchange is Microsoft's successor to MS Mail. Exchange, however, includes not only e-mail, but also management of group scheduling, electronic forms, groupware, and Internet connectivity. An Exchange folder can contain standard notes (mail), files, and instances of Exchange forms. All of this data is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

- Address Lists
  - Personal Address Books
  - Global Address Lists
  - Distribution Lists
- Exchange Folder Contents
  - mail messages
  - Exchange Form applications
  - properties of OLE documents
- Exchange Administrator
  - 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.

You can use each Exchange data source like a database table and you can link each Exchange data source to other data sources. For example, you can join the Message Tracking Log 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 the data source that 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
P2smapi.dll	Exchange Folder Contents and Address Lists
P2sexsr.dll	Properties of Exchange Mailboxes and Public Folders
P2srepl.dll	Replica Lists of Public Folders
P2sacl.dll	Access Control Lists (ACL) of Public Folders
P2soutlk.dll	Microsoft Outlook

Exchange translation files work directly with the Microsoft Messaging API (MAPI). MAPI acts as a database engine for Exchange data.

How the Report Designer treats Exchange data

Crystal Reports treats Exchange Administrator data (Public Folder Admin, Public Folder ACL, Public Folder Replica, and Mailbox Admin server types) as a physical database server. To access this data, you must log on using a SQL/ODBC server and select the appropriate tables from the Choose SQL Table dialog box.

When you log on to any of the Exchange Administrator servers, you must select a profile (or profiles). Each profile represents an Exchange Server.

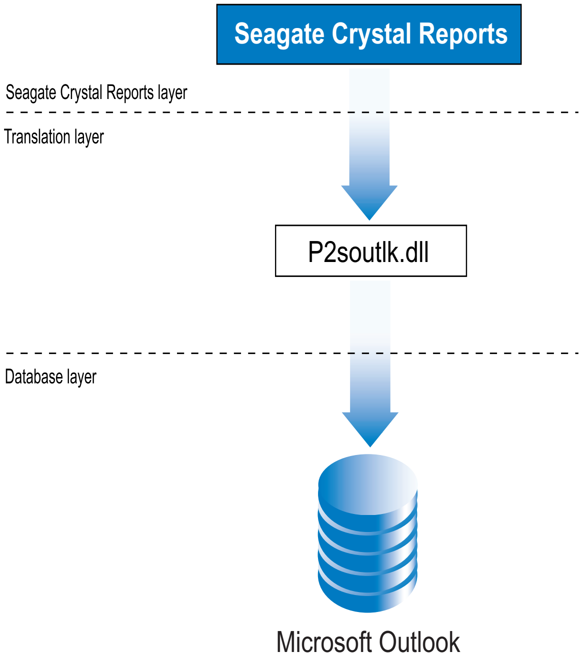
Note: Refer to your Microsoft Exchange documentation for more information about how to set up and create Exchange profiles.

Outlook

Microsoft Outlook provides powerful organization and information management tools in the form of e-mail, calendars, task lists and other popular tools.

Crystal Reports lets you report on your Outlook data by providing the translation file P2soutlk.dll. Because Outlook data is stored in a flat file system, the translation file accesses it directly.

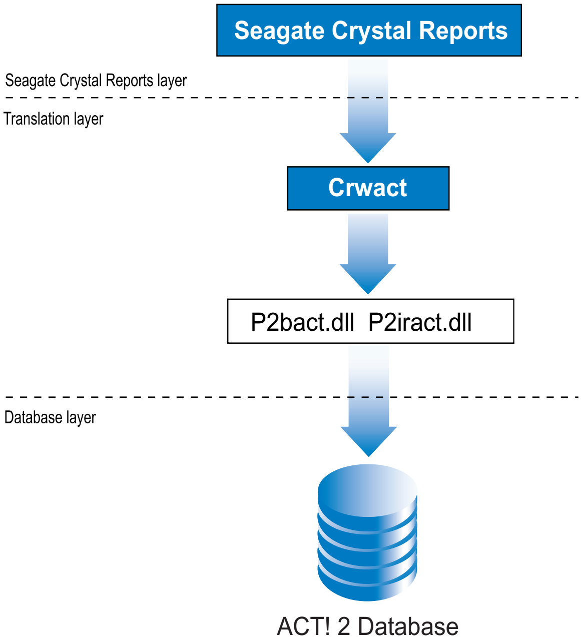
Note: You must have Microsoft Outlook installed, or available through UNC (universal naming convention) read-access. UNC is a standard format for paths that include a local area network file server. The basic syntax is:
`\\server\share\path\filename`



ACT!

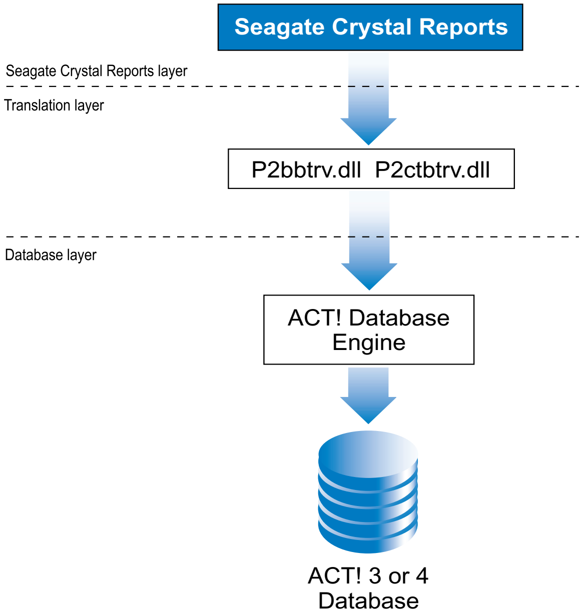
Symantec's ACT! contact management software stores information in a relational database format similar to the xBase format. See "dBASE, FoxPro, Clipper" on page 578. Crystal Reports can read this data and lets you produce reports based on your contact information.

ACT! 2 databases



You open data in ACT! 2 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 the program. This file forces Crystal Reports to load the P2bact.dll and P2iract.dll translation files. This step is important because ACT! 2 data looks like xBase data to Crystal Reports, so it will use the P2bxbse.dll translation file unless instructed otherwise. If this happens, the data will be translated as xBase data rather than ACT! 2 data, and may not appear correctly in your report. See "ACT! databases" on page 568.

ACT! 3 and 4 databases

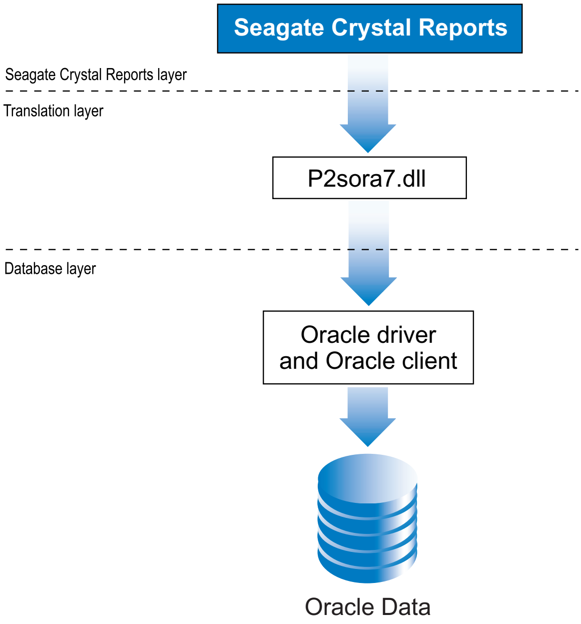


Versions 3 and 4 of ACT! do not use an intermediate step in the translation layer. The translation files used for all ACT! databases version 3 and higher are P2bact3.dll and P2iract.dll. See "ACT! databases" on page 568.

Oracle

Crystal Reports supports direct access to Oracle 7 and Oracle 8 SQL databases. This accessibility is provided by the P2sora7.dll and P2lora7.dll translation files. These files communicate with the Oracle 7 and Oracle 8 database driver, which works directly with Oracle databases and clients, retrieving the data you need for your report.

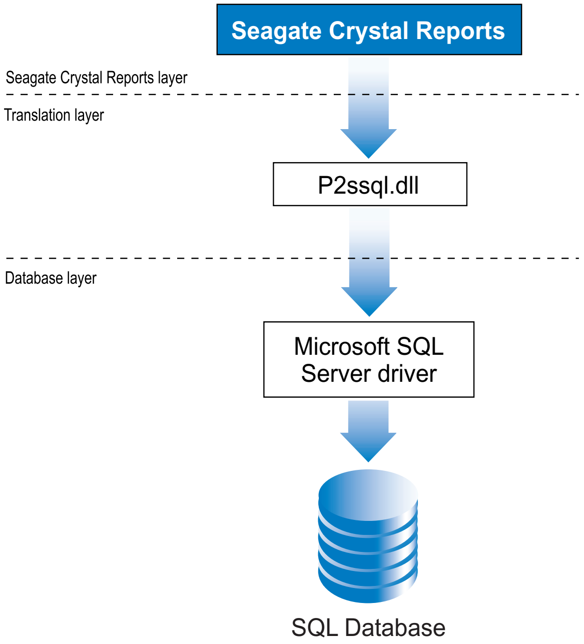
Note: In order for Crystal Reports to use Oracle 7 and Oracle 8 SQL databases the Oracle client software must be installed on your system, and the location of the Oracle driver must be in the PATH statement of your Autoexec.bat file.



Note: When logging on to Oracle using P2sora7.dll or P2lora7.dll, the date format is changed to match the default date/time format of Crystal Reports. For more information about default formats, see "To set standard default formats for Date, Time, and Date/Time fields" on page 228.

Microsoft SQL Server

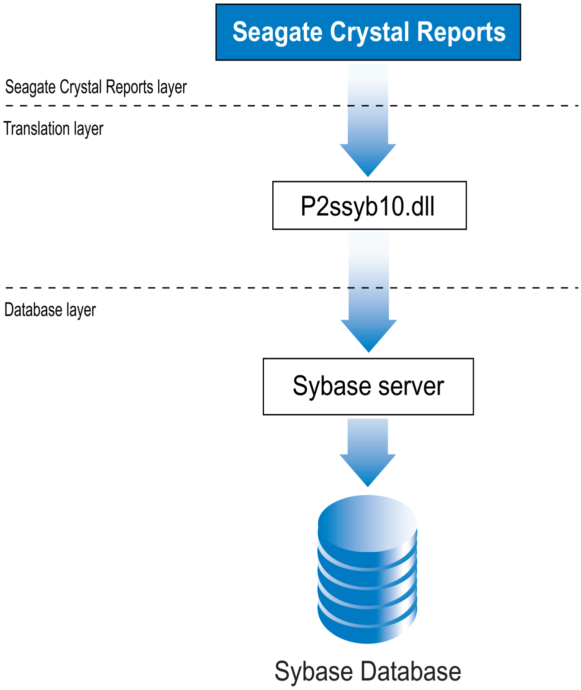
Crystal Reports provides direct access to databases created by Microsoft's SQL Server, versions 6.0, 6.5, and 7.0. The P2ssql.dll file installed with the program translates requests to the SQL server and then returns data from the server. It communicates directly with the Microsoft SQL Server driver.



Note: Network administrators and IS managers who run Microsoft's Systems Management Server for BackOffice can run reports on 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 opens SQL data created by Sybase System 10 or System 11 directly through the P2ssyb10.dll, installed with Crystal Reports. This translation file works with the Sybase database drivers 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.



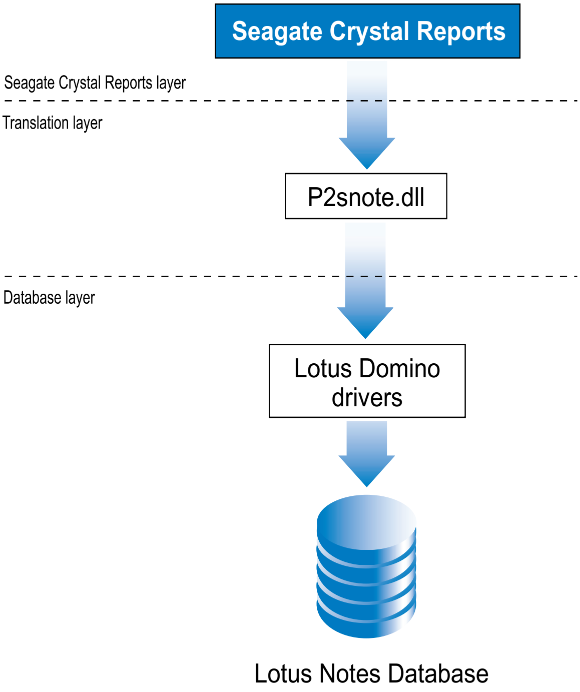
Lotus Domino

Lotus Domino is a powerful groupware application used for communication and information sharing between departments in an organization.

A Lotus Domino database can be read by Crystal Reports using either a native driver or an ODBC driver. For information about the ODBC driver, see "Lotus Domino" on page 614. The following table compares the features of the native driver and the ODBC driver.

Feature	Native driver	ODBC driver
Logon to different servers	Yes	Yes
Use Lotus Domino client extension manager to pass password to Lotus Domino client	Yes	No
Obtain list of available databases (including subfolders)	Yes	Yes
List database filename	Yes	Yes
List database title	Yes	No
Report on Views and Forms	Yes	Yes
Distinguish between Views and Forms with same name	Yes	No
Omit Forms with no fields	Yes	No
Omit Subforms (cannot be used independently)	Yes	No
Text fields shown as strings	Yes	Yes
Text fields shown as memos	Yes	No
Rich Text fields shown as strings	Yes	No
Rich Text fields shown as memos	Yes	Yes
Computed values for composed fields	Yes	No
Multiple value fields shown as first value with proper type	Yes	No
Multiple value fields shown as string and memo	Yes	No
Converting report from other driver	Yes	Partial
Left outer join on linked tables	Yes	Yes
Other joins on tables	No	Yes
Server-side filtering	Partial	Yes

Choosing Lotus Domino during installation adds the translation file P2snote.dll to the Windows or Winnt/Crystal directory.



Note:

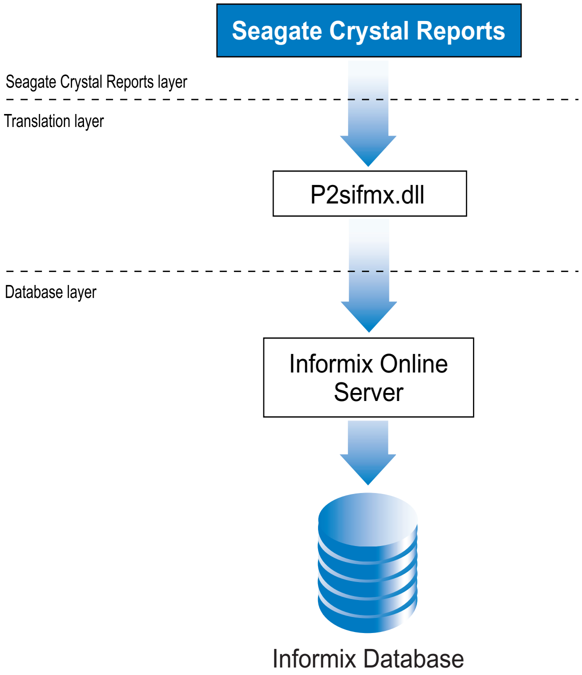
- Nextpwd.dll and the Lotus API DLL (Lcppn201.dll) are installed in the Lotus Domino executable directory.
- The following line is added to the Notes.ini file:

  ```
  EXTMGR_ADDINS=extpwd
  ```
- The Lotus Domino executable and data directories are added to the PATH.

Informix® Online Server™

The Informix Online Server provides client/server capabilities along with SQL compliance.

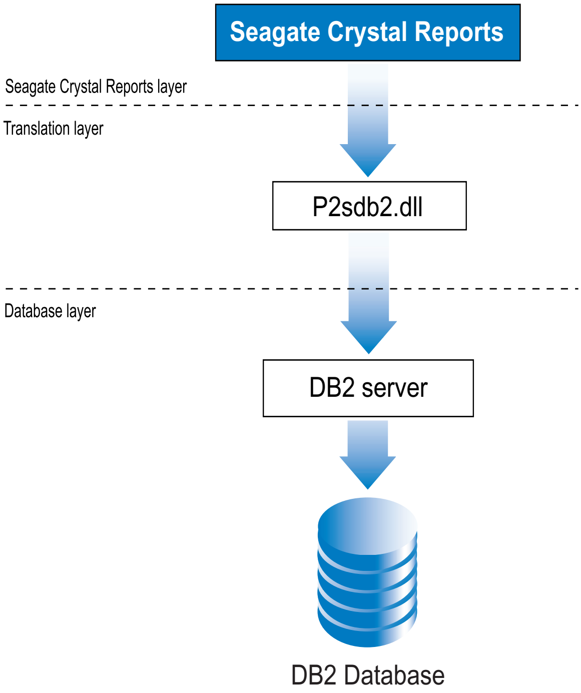
Crystal Reports opens SQL data created by Informix through the P2sifmx.dll, installed with Crystal Reports. This translation file works with the Informix database drivers to read Informix data. You must have the Informix client installed and configured properly before using this translation file.



DB2 Server

IBM’s DB2 Server is an SQL compliant, client/server application that is part of IBM’s popular database management suite.

Crystal Reports opens SQL data created by DB2 through the P2sdb2.dll, installed with Crystal Reports. This translation file works with the DB2 database drivers to read DB2 data. You must have the DB2 client installed and configured properly before using this translation file.

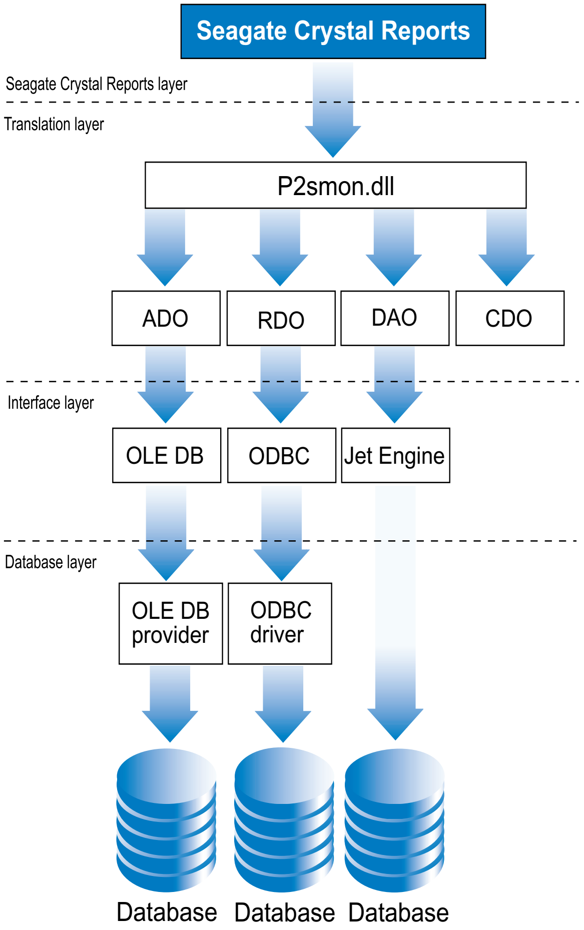


Active Data

Crystal Reports can be used to report off ActiveX data sources, including ActiveX Data Objects (ADO), Remote Data Objects (RDO), and Data Access Objects (DAO). Reports created off these data sources can be used in Visual Basic applications that use these ActiveX data sources. Visual Basic applications, Crystal Reports also supports Crystal Data Objects (CDO), which are sets of relational data created at runtime using Visual Basic arrays.

Crystal Reports also offers the ability to create reports without specifying an actual data source. This is supported through the use of a Data Definition file, which is an ASCII text file with placeholders to represent database fields. By using reports created off a Data Definition File, an application developer has the flexibility of specifying the actual data source at runtime.

The translation file Crystal Reports provides for Active Data reporting is P2smon.dll. This translation file works with a different set of drivers for each data source.

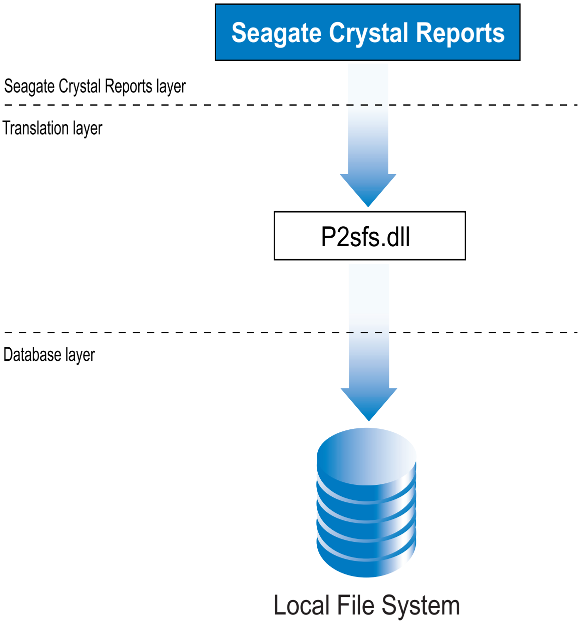


Local file system

You can use Crystal Reports to report on files located on your system or network drives including information such as file name, version number, date, and so on.

The translation file P2sfs.dll is provided by Crystal Reports for reporting on your local file system.

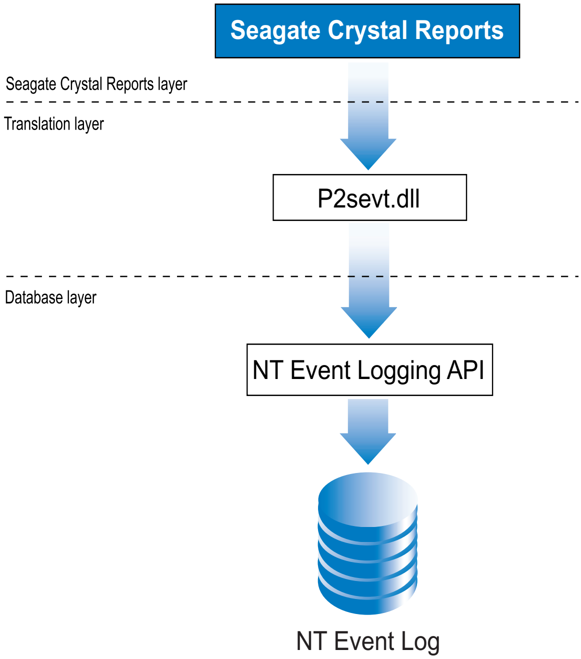
Note: You must have UNC (universal naming convention) read-access to your local file system. UNC is a standard format for paths that include a local area network file server. The basic syntax is: `\\server\share\path\filename`



NT Event Log

If using 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 on 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.

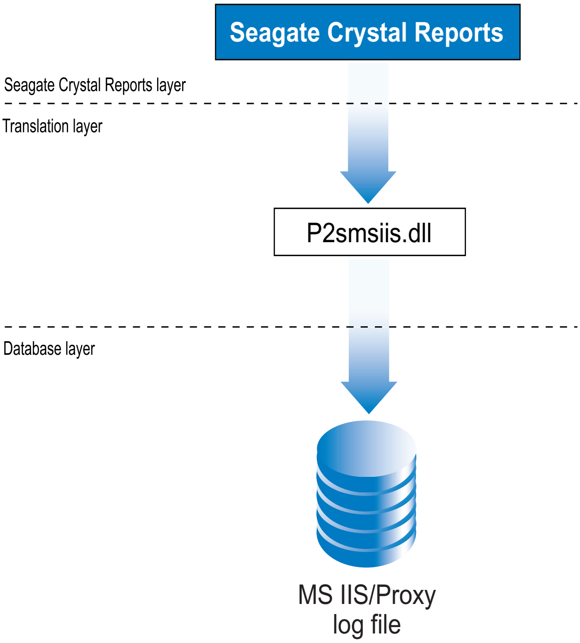


Microsoft IIS™/Proxy log file

If you use Microsoft Internet Information Server (MS IIS), or Microsoft Proxy™, you can use Crystal Reports to report on the log files. These log files keep track of different types of events that occur when using a web server and browser.

The translation file Crystal Reports provides for MS IIS and Proxy log files is P2smsiis.dll. Because log file data is normally stored in a flat file system, the translation file accesses it directly.

Note: You must have UNC (universal naming convention) read-access to your local file system. UNC is a standard format for paths that include a local area network file server. The basic syntax is: `\\server\share\path\filename`

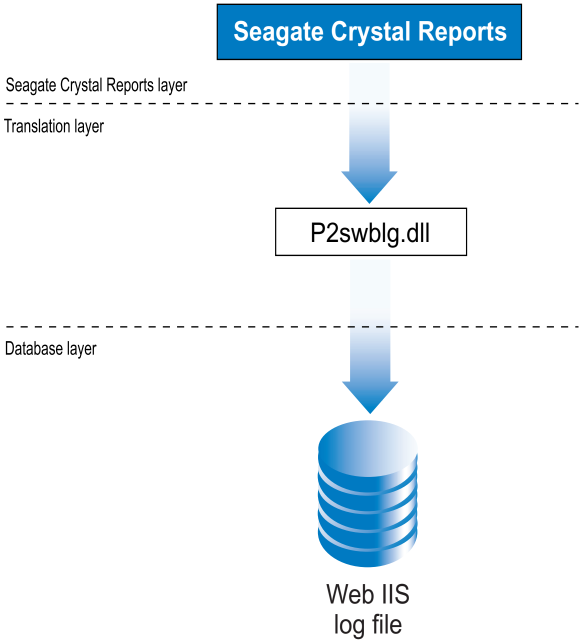


Web/IIS log files

You can use Crystal Reports to report on Web log files. These log files keep track of different types of events that occur when using a web server and browser.

The translation file Crystal Reports provides for Web/IIS log files is P2swblg.dll. Because log file data is normally stored in a flat file system, the translation file accesses it directly.

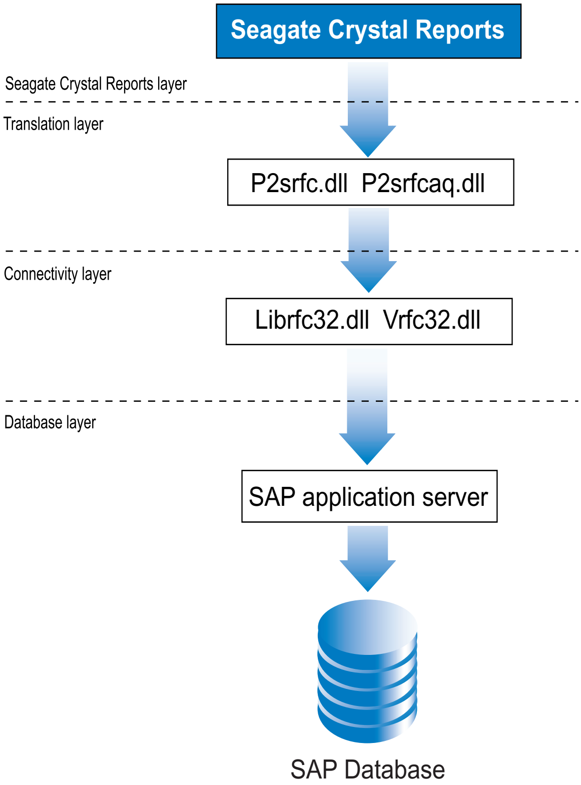
Note: You must have UNC (universal naming convention) read-access to your local file system. UNC is a standard format for paths that include a local area network file server. The basic syntax is: `\\server\share\path\filename`



SAP®

SAP, a vendor of client/server business application solutions, offers an ERP (enterprise resource planning) software package that includes SQL databases you can report on using Crystal Reports.

Crystal Reports provides access to databases created by SAP ERP. The P2srfc.dll and P2srfcaq.dll files installed with Crystal Reports translate requests to the SAP Application Server's network connectivity layer. The SAP Application Server then interprets the request and returns data from the SAP database.

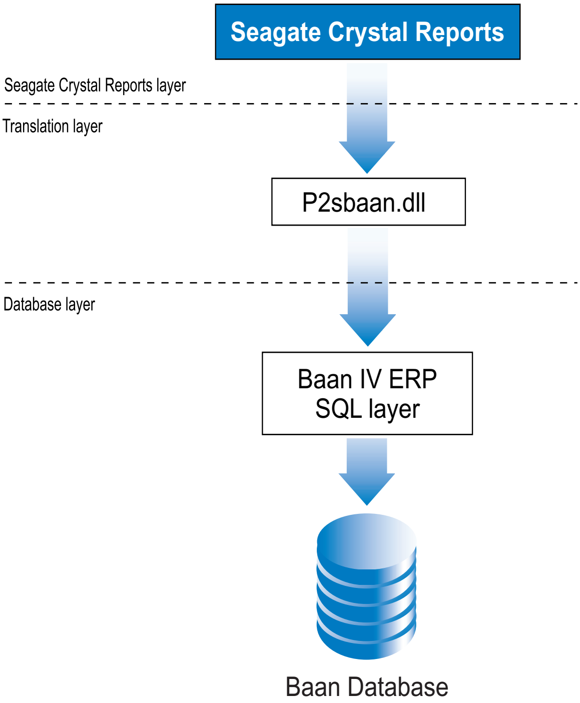


Note: Crystal Reports supports access to the SAP database using the files librfc32.dll and vrf32.dll.

Baan®

Baan, a vendor of client/server business application solutions, offers Baan IV, an ERP (enterprise resource planning) software package. Baan IV includes a custom SQL client/server and database you can report on using Crystal Reports.

Crystal Reports provides access to databases created by Baan IV ERP. The P2sbaan.dll file installed with Crystal Reports translates requests to the Baan SQL server and then returns data from the server.



Note:

- The Baan client does not have to be installed to use the Crystal Reports translation file.
- Crystal Reports supports access to the Baan database using the files Bn4c2nt2.dll and P2sbann.ini.

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 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 that can communicate with ODBC (such as Crystal Reports), will instantly have access to any ODBC data source. With ODBC drivers available for most common DBMS products, the range of data types 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 have.

Because of the extreme flexibility built into ODBC, you can use the same report file with different ODBC data sources. For example, you might design a report using an Oracle data source, and 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 "Changing the ODBC data source accessed by a report" on page 561.

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 statements 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 on to the server system, freeing up local memory and resources for 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 other more important work.

Disadvantages

Because of the many layers involved in passing data through ODBC from a database to an application, ODBC data sources often take more time to return data than natively accessed data sources. First, Crystal Reports must request some data. The request must be translated by the ODBC translation layer into 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. For more information, see "DBMS translation (ODBC data source) layer" on page 605. 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, but it can also fail at any of several possible levels.

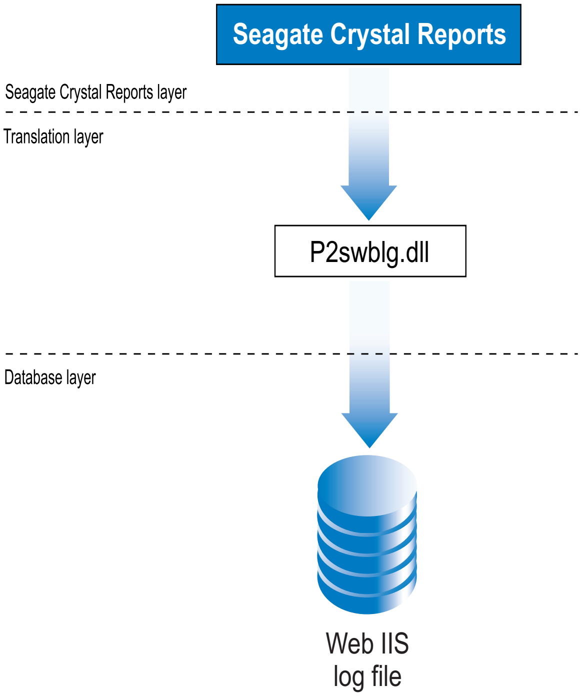
In addition, ODBC data sources must be correctly configured and set up in the Odbc.ini and Odbcinst.ini files before they can be used. If you create a report on one system and try to open it on another system that does not have the same ODBC data source set up, Crystal Reports will not be able to connect 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" on page 539.

Five layers

The process by which Crystal Reports accesses data from an ODBC data source consists of five layers:

- "Crystal Reports layer" on page 604
- "ODBC translation layer" on page 604
- "ODBC layer" on page 605
- "DBMS translation (ODBC data source) layer" on page 605
- "Database layer" on page 606.



By using the Structured Query Language (SQL), all five layers can conveniently pass data from the database to your report.

Crystal Reports layer

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 "SQL queries" on page 567.

ODBC translation layer

Crystal Reports uses the Dynamic Link Library P2sodbc.dll to communicate with ODBC. This file is unique to 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 layer

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 Windows 95, Windows 98, and Windows NT, the ODBC DLLs are:

- Odbc32.dll
- Odcccp32.dll
- Odccint.dll (ODBC 2.5 and later).

Although information regarding data sources is recorded in Odbc.ini and Odbcinst.ini, 32-bit ODBC uses the Windows Registry database to retrieve information about individual data sources.

Note: For more 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 more advanced users often take advantage of the features of the SQL language used by ODBC.

DBMS translation (ODBC data source) layer

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 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 on which it has been installed. 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 with the ODBC Setup application. For more information about 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, Sybase SQL, or MS SQL Server. Some users, however, prefer to actually specify a particular 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 to remember 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 platform your database server exists on. It is the DBMS client that connects and communicates with the server; Crystal Reports and the drivers need only communicate with the DBMS client. For more information, see “Using SQL and SQL databases” on page 535.

Database layer

The database file which the ODBC data source refers to 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 can be located anywhere on a network, as long as ODBC can communicate with it through the ODBC data source.

Installing database clients

Ideally, your database client should be installed and configured before you install Crystal Reports. If, however, you've installed Crystal Reports first, you can ensure the proper configuration of Crystal Reports' ODBC Data Source Name (DSN) by following this procedure.

To ensure correct ODBC DSN configuration

1. On the **Start** menu, select **Settings,** select **Control Panel**, then double-click **Add/Remove Programs**.
2. Select **Crystal Reports** and click **Add/Remove**.
3. In the Crystal Reports Setup dialog box, click **Add/Remove**.
4. Expand the **Data Access** folder and select the appropriate data access option.

   For example, if you have installed the Informix database client, select Informix from the Data Access folder.
5. Click the selected data access option and choose **Entire feature will be unavailable**.

   An X appears next to the option and the Feature Description area includes a remark that the feature will be completely removed.
6. Click **Next** on this dialog box and the following one.

   The data access option you selected is removed.
7. Repeat steps 2 through 4.
8. Click the selected data access option and choose **Will be installed on local hard drive**.

   The X next to the option is removed and the Feature Description area includes a remark that the feature will be installed.
9. Click **Next** on this dialog box and the following one.

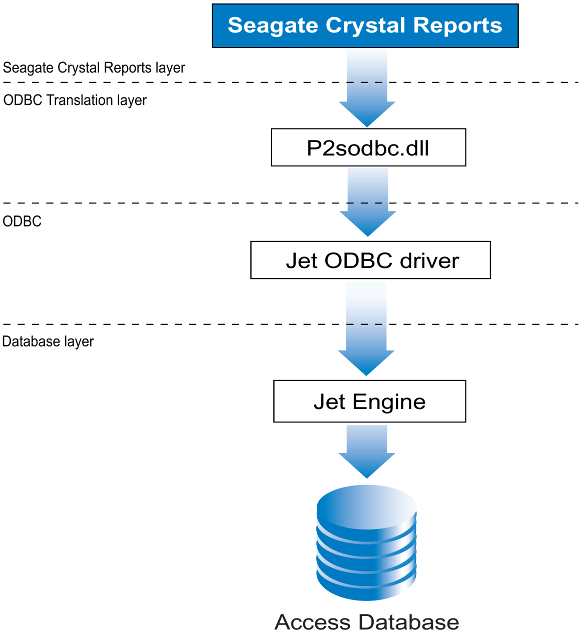
   The data access option you selected is reinstalled and its ODBC DSN is configured properly.

Common ODBC database formats

Access

Although Microsoft Access files can be opened directly by Crystal Reports (see "Microsoft Access" on page 577), you may wish to use ODBC for these files instead. When communicating with Access databases, ODBC uses a translation file that communicates with the Microsoft Jet Database Engine through the DAO (see "DAO" on page 579). The Jet Database Engine is an example of a DBMS-specific ODBC driver.

The Jet engine is the part of the Access DBMS that does all of the actual work with an Access database. Jet is a required component for working with Access data. Since the Jet engine is an integral part of all Access databases, it is shown here as part of the database layer.



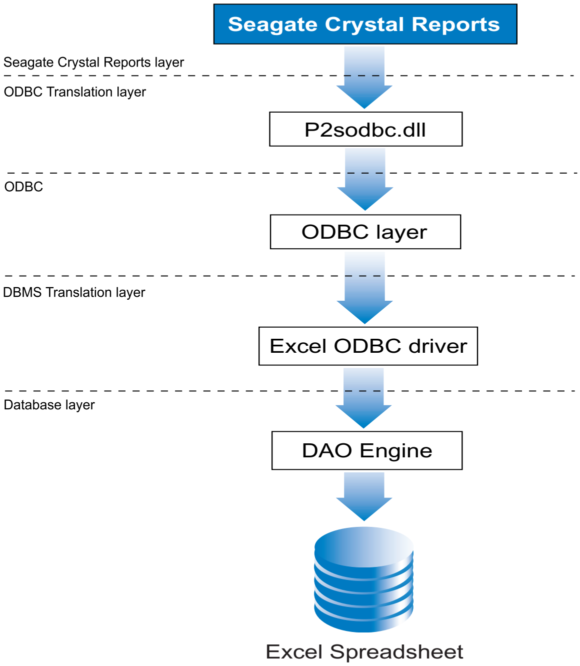
Excel

You can convert Microsoft Excel spreadsheets 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 and later, use Define on the Name submenu of 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, and then select it from Crystal Reports.

Note:

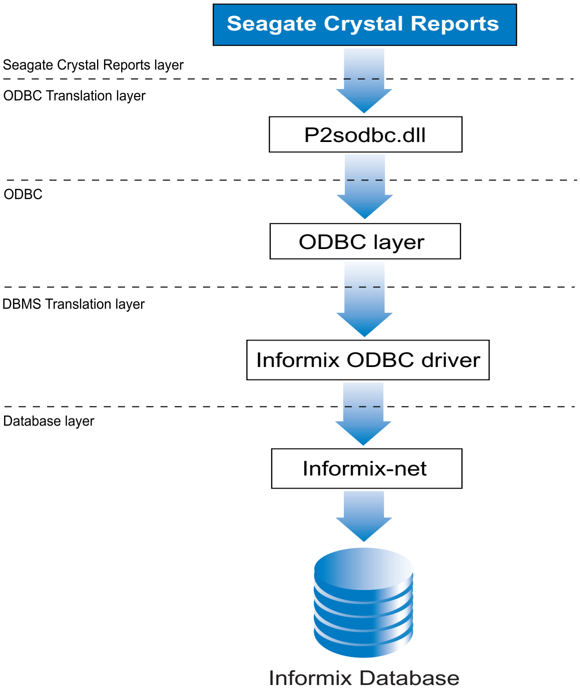
- If you are using Excel 7 or later, you can export your spreadsheets as Access database tables, and read them from Crystal Reports as you would read any other Access tables. For more information, refer to your Excel documentation.
- You can also use the Microsoft Excel add-in to create reports in Crystal Reports.

In Windows 95, Windows 98, and Windows NT, you can set up an ODBC data source for Excel spreadsheets using the DAO engine. The DAO engine is installed on your system when you install Crystal Reports. However, you must set up an ODBC data source manually for your Excel spreadsheet. See "Setting up an ODBC data source" on page 554.



Informix

Crystal Reports accesses Informix databases through ODBC. The Informix client should be installed on your machine. Otherwise, Crystal Reports will provide the driver that ODBC uses to communicate with the Informix database engine.



Note: Crystal Reports provides the ODBC driver Crinf914.dll for reading Informix data.

SQL databases through Merant 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
- DB2/2.

Crystal Reports provides the Merant DataDirect ODBC drivers for these SQL database formats. Not all SQL databases are accessed through these drivers, but as a convenience, the program automatically sets up these drivers for you during installation. Crystal Reports still communicates with ODBC through the P2sodbc.dll translation file, but the DataDirect drivers provide ODBC with easy access to the actual databases.

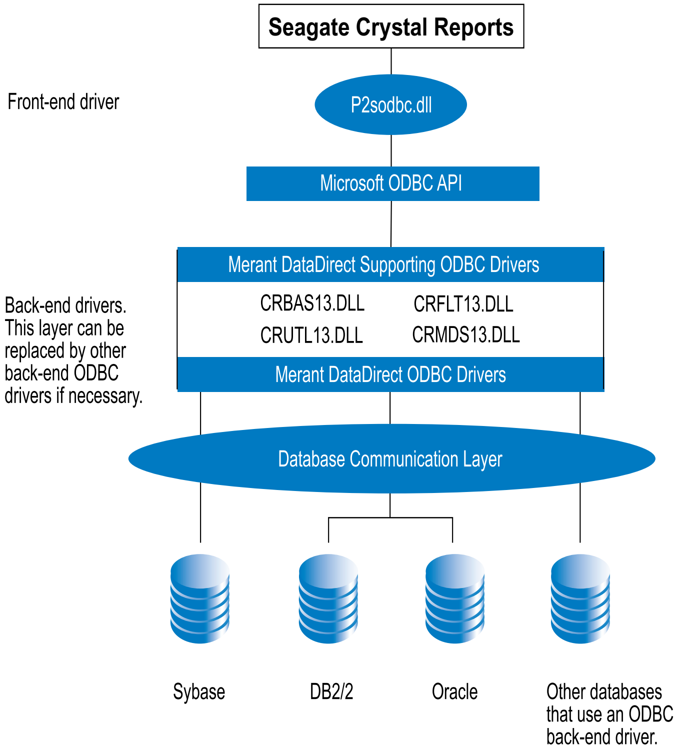
Note: If you are using an SQL database not accessed by the Merant drivers, refer to your DBMS documentation for more information about the required ODBC drivers.

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 databases” on page 535.

SQL databases are any 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 provided for this DBMS translation layer are all Merant DataDirect drivers.

Note: ODBC does not have to use the Merant DataDirect drivers to access the SQL databases shown here. Most SQL DBMS applications provide their own ODBC drivers. However, Crystal Reports sets up the Merant drivers for you automatically.


Seagate Crystal Reports
Front-end driver
P2sodbc.dll
Microsoft ODBC API
Merant DataDirect Supporting ODBC Drivers
CRBAS13.DLL
CRFLT13.DLL
CRUTL13.DLL
CRMDS13.DLL
Merant DataDirect ODBC Drivers
Back-end drivers. This layer can be replaced by other back-end ODBC drivers if necessary.
Database Communication Layer
Sybase
DB2/2
Oracle
Other databases that use an ODBC back-end driver.

Note: Crystal Reports provides Merant drivers for accessing SQL databases. These 32-bit drivers are named Cr*14.dll.

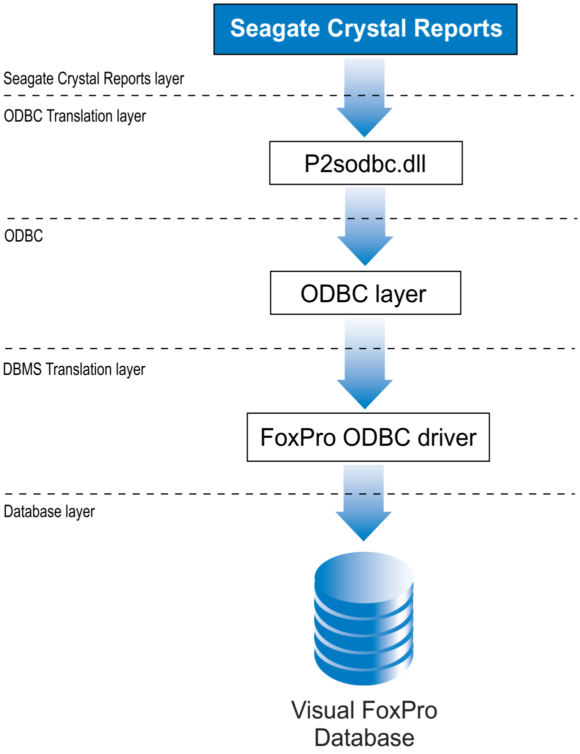
As mentioned previously, Crystal Reports automatically creates several ODBC data sources for the SQL database formats supported by Merant DataDirect. These data sources are automatically created:

- ODBC-CRDB supports DB2/2
- ODBC-CROR7 supports Oracle 7
- ODBC-CROR8 supports Oracle 8
- ODBC-CRSYB supports Sybase System 10.

Visual FoxPro™

Microsoft Visual FoxPro data is accessed through ODBC, while FoxPro data from version 2.6 and earlier is accessed directly through the xBase engine. If you are using FoxPro version 2.6 or earlier, see "dBASE, FoxPro, Clipper" on page 578.

Crystal Reports provides the ODBC driver Vfpodbc.dll, allowing ODBC to work with Visual FoxPro data.

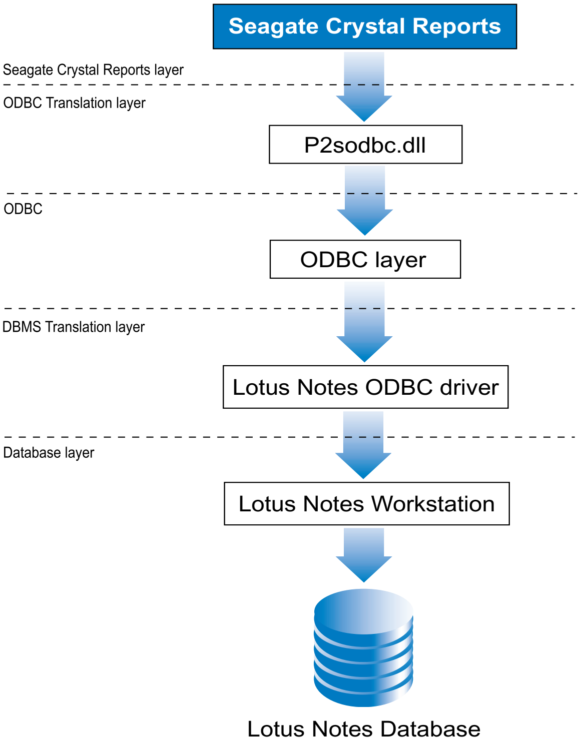


Lotus Domino

A Lotus Domino database can be read by Crystal Reports through ODBC. For Windows 95, Windows 98, and Windows NT, the Lotus Domino DBMS translation layer consists of three files:

- Nsql32.dll
- Nsqlv32.dll
- Nsqlc32.dll

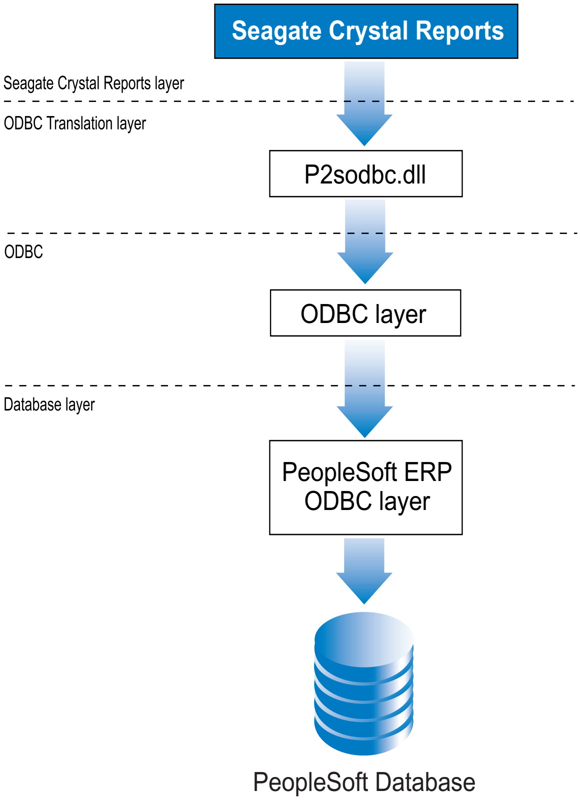
These files use the drivers installed by the Lotus Notes client to work with the Lotus Domino database. The client component of Lotus Domino must be installed on the local machine.



Note: The Domino Server can be substituted for the Lotus Notes client in the database layer of this diagram.

PeopleSoft®

PeopleSoft, a vendor of enterprise application solutions, offers the PeopleSoft ERP (enterprise resource planning) system. Crystal Reports works with PeopleSoft ERP data using ODBC.



Note: The PeopleSoft client must be installed and configured properly to use the Crystal Reports ODBC translation file (P2sodbc.dll).

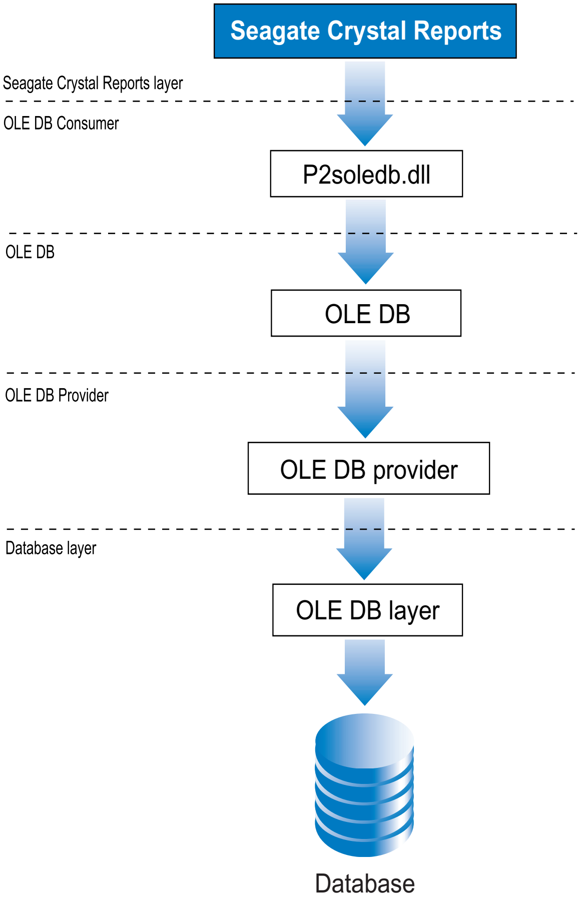
OLE DB

OLE DB is a database connectivity entity from Microsoft similar in conception to ODBC.

Note:

- OLE DB is a connectivity methodology and not a database. Like ODBC, it facilitates the exchange of data created in other database management systems by providing an interface layer.
- OLE DB requires the use of an OLE DB provider, which is the OLE DB equivalent of an ODBC driver.

Crystal Reports provides the OLE DB consumer P2soledb.dll, allowing OLE DB to communicate with an OLE DB provider.

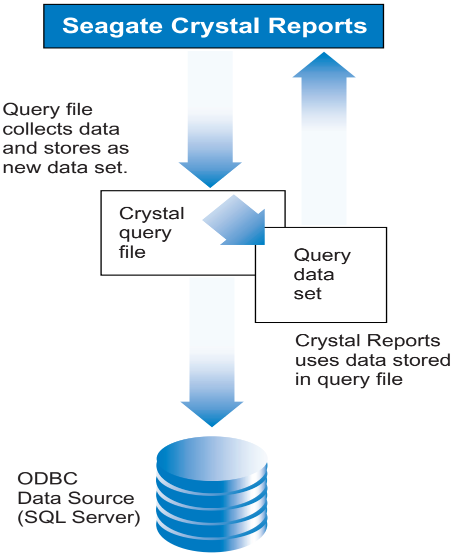


Crystal SQL Designer files

The Crystal SQL Designer is a query which 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 SQL Designer file (.QRY). The Crystal SQL Designer 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 SQL Designer file gathers data from ODBC data sources (such as SQL servers) by means of SQL (Structured Query Language). The SQL Designer sends 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 SQL Designer and stored as a query file.

Note: If you are not familiar with SQL, see "SQL join types (ODBC data sources)" on page 527.


Seagate Crystal Reports
Query file collects data and stores as new data set.
Crystal query file
Query data set
Crystal Reports uses data stored in query file
ODBC Data Source (SQL Server)

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 on 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, the Report Designer only needs to work with a small set of data, saving you time and trouble. For more information, see "Creating a new query using the Crystal SQL Designer" on page 623.

Advantages

While there are many reasons for using the Crystal SQL Designer, the primary one 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. By using this SQL pass-through technique, where data retrieval tasks are passed through to the server, you retrieve your data faster and more efficiently. Any time you are working with SQL data, you should consider building a query file before designing your report.

In addition, the SQL Expert provides for more control over your data with the SQL language. The Report Designer provides some SQL language features when accessing SQL or ODBC data, but the 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" on page 539.

Disadvantages

Since a query represents a complete data set, any records, fields, or tables that were not included when the query was created will not be available when using the query to build a report. A report cannot 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 include an additional data source, such as a database table or another query file, to use in the report.

Crystal Dictionary files

A dictionary (.DC5) file is a structured and simplified view of organizational data that you can create for any of the individuals in your organization that are using Crystal Reports. With a dictionary, end users only see the subset of tables and fields they need.

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 the user can still access data directly from the database. Dictionaries simply provide all the convenience without the headaches.

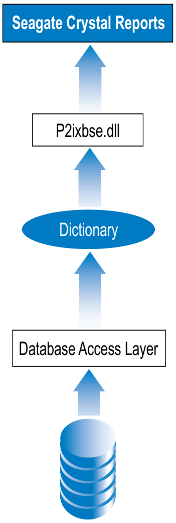
If you choose to design a report by using a dictionary, you cannot use any other data source in that report. This includes other dictionaries, Crystal reports, ODBC data sources, direct access databases, and all other data sources. The purpose of a dictionary is to provide fast, easy to understand access to a particular set of data. Connecting to other sources of data within the same report would defeat this purpose. See "Creating a new dictionary" on page 641.

Dictionaries allow you to:

- 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
- 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. The complexity of your data will not stop end users from creating powerful reports.



Crystal Reports reads the dictionary file by using the P2ixbse.dll file. This translation file is based on the xBase engine used to access dBASE, FoxPro and Clipper databases, but it provides all the flexibility needed to read dictionaries. See "dBASE, FoxPro, Clipper" on page 578.

Note: If you upgraded from an earlier version of Crystal Reports that allowed dictionary files, the older dictionary files (.DCT) were based on the Btrieve engine. If you convert those older files to new dictionary files, 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 dictionary 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 using the same routes that Crystal Reports uses to read the data. The sections on direct access database layers and ODBC data sources describe the files dictionaries need for reading data.

The benefit of dictionaries 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 that is easily accessible and convenient for users.

Finally, dictionaries provide an easy means of changing the originally accessed data without changing the view of data seen by users. For example, your original database may contain Price, Quantity, and Cost fields, and 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 is also useful when the entire underlying database format is changed. Once again, just update the dictionary.

Dictionary files are often created by an IS manager who controls a company database. The IS manager can work with data that is organized and classified to create simple, easy-to-use dictionaries for company employees. In each dictionary file, the employees will find only the data they need, since the original data has been manipulated or customized by the IS manager for varying usage.

The Crystal SQL Designer 29

A query is a selection statement used to obtain specific information from the database. This chapter describes how to use the Crystal SQL Designer to create, modify and optimize queries.

The Crystal SQL Designer

The Crystal SQL Designer creates a query, a request for specific information from a database. If you are requesting information from an SQL database (or from a database accessed 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 with SQL can take quite a while. To assist you, Crystal Reports provides you with the Crystal SQL Designer, a component that eliminates the need to fully understand SQL so you can start building effective queries right away.

- If you are new to querying, you will appreciate the way the Crystal SQL Designer helps you create queries. By answering a few questions on a set of sequential tabs, you'll give the program all the information needed to generate a query that fits your needs.
- If you are an SQL professional, you will appreciate the ability to fine-tune queries that the Crystal SQL Designer generates. If you are more comfortable writing your own SQL queries, you will find it easy to enter the queries directly or even copy and paste them in from another source.

Note: The Crystal SQL Designer can only access data stored in an ODBC data source. Any database (including SQL) mentioned in this chapter must be accessed through ODBC.

Why use a query?

Query files provide a means of off-loading much of the data-generating tasks normally performed by Crystal Reports to the database server. The set of data returned represents a subset of the actual data in the database - only the data you specifically need.

If you are an experienced SQL programmer, the Crystal SQL Designer enables you to reuse your existing queries, quickly and easily. All of your existing queries become convenient query files that you can use to design reports with Crystal Reports.

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 an SQL statement. By using the Crystal SQL Designer, you are able to use all of the power of SQL, including complex joins, sorts, and aggregate functions. In addition, a query file provides full ANSI SQL compatibility.

Using the Crystal SQL Designer

The Crystal SQL Designer has two primary uses:

- designing and developing data sets for building reports in Crystal Reports
- retrieving and analyzing information on an "as needed" basis to facilitate informed decision-making.

Crystal Reports and query data sets

Crystal Reports enables you to design a report based on a query data set rather than on tables and fields. Since the query contains a predefined set of data, the tables and fields necessary for the report are already included.

When designing your report, you can use the set of data produced by the SQL query. The name of the query, along with fields it accesses, appear 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 make projections, etc.). The Crystal SQL Designer makes it easy 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, with minimal effort.

Note: The Crystal SQL Designer cannot retrieve any number value longer than 20 characters or calculate a formula with a value longer than 20 characters. Values longer than 20 characters will be truncated.

Creating a new query using the Crystal SQL Designer

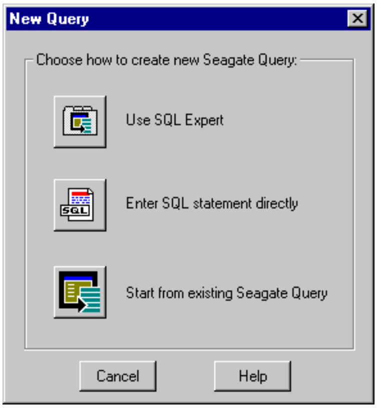
Accessing the Crystal SQL Designer

1. On the **Start** menu, select **Programs,** select **Crystal Reports Tools**, then click **Crystal SQL Designer**.

   Note: The location of the Crystal SQL Designer may vary depending on the location you specified during installation.

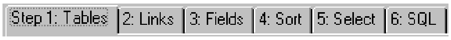
2 On the **File** menu, click **New**.

The New Query dialog box appears.


New Query
Choose how to create new Seagate Query:
Use SQL Expert
Enter SQL statement directly
Start from existing Seagate Query
Cancel
Help

3 Click **Use SQL Expert**.

The Create SQL Expert appears. This expert contains six tabs that will lead you step-by-step through the query creation process.



Note: The Links tab only appears when two or more database tables are selected on the Tables tab.

Previewing data

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 Finish at the bottom of the dialog box. After you are done viewing the query data, click Edit and choose Query to return to the Create SQL Expert.

Specifying a table

1 Click **SQL/ODBC** or **Dictionary**.

If you click SQL/ODBC:

- Select a server type from the Log On Server dialog box.
- Click OK.
- Choose the tables you would like to add to the query. Click each table, then click Add. The tables appear in the list area of the Choose SQL Table dialog box. After you have added all of the tables, click Done.

If you click Dictionary:

- Select the Dictionary (.DC5), then click OK.
- All ODBC database tables accessible from the dictionary are added to your SQL query.

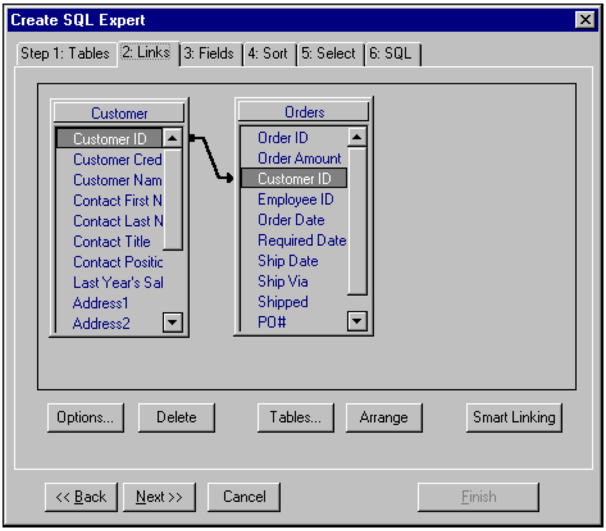
Note: Dictionary files must be based on an ODBC data source.

2 Click **Next**.

The Links tab appears (if two or more database tables have been selected).

Linking tables and specifying a join type

When you click "Smart Linking," the Create SQL Expert makes links between the matching fields of each table. The link is represented by a line with an arrow. This is referred to as a link line.



Link information

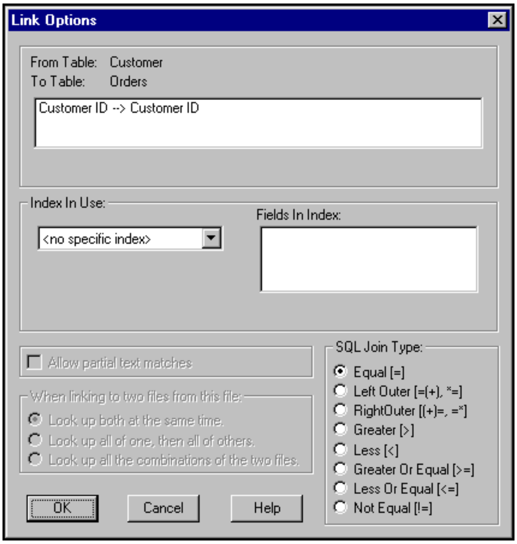
To find out more about a link

1 Click the link line.

The line becomes highlighted, along with the fields on either end of the line.

2 Click **Options**.

The Link Options dialog box opens, displaying the link details.



Link Options

From Table: Customer
To Table: Orders

Customer ID --> Customer ID

Index In Use: <no specific index>

Fields In Index:

□ Allow partial text matches

When linking to two files from this file:
● Look up both at the same time.
○ Look up all of one, then all of others.
○ Look up all the combinations of the two files.

SQL Join Type:
● Equal [=]
○ Left Outer [=(+), *=]
○ RightOuter [(+)=, =*]
○ Greater [>]
○ Less [<]
○ Greater Or Equal [>=]
○ Less Or Equal [<=]
○ Not Equal [!=]

3 Click **Cancel** to return to the Create SQL Expert.

4 Click **Next**.

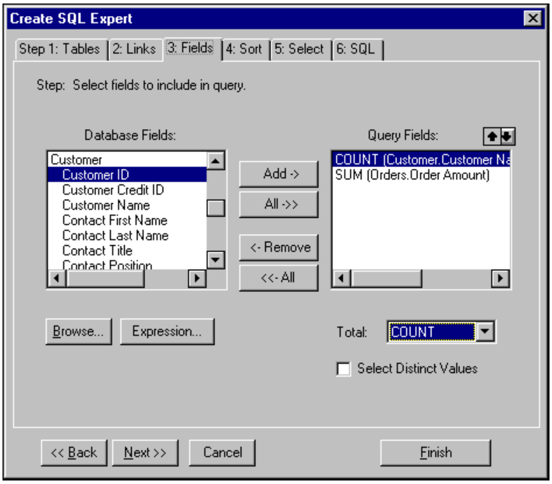
The Fields tab appears.

Adding fields to a query

1 Click the field from the Database Fields area you would like to have appear in your query.

2 Click **Add** to move it to the Query Fields area.

3 Click **Remove** to remove a field from the Query Fields area.



Step: Select fields to include in query.

Database Fields: Customer, Customer ID, Customer Credit ID, Customer Name, Contact First Name, Contact Last Name, Contact Title, Contact Position

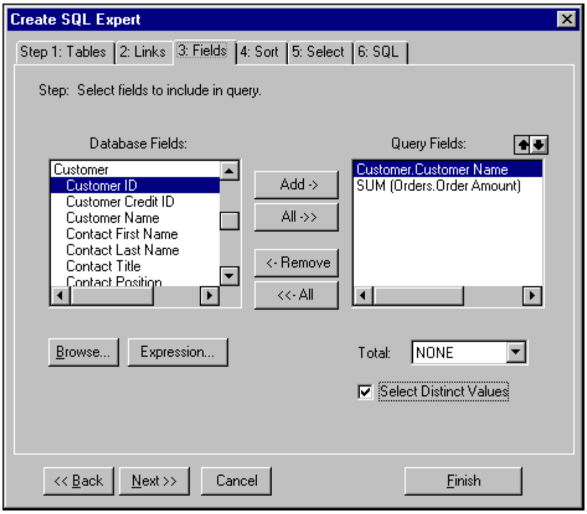
Query Fields: COUNT (Customer.Customer Na, SUM (Orders.Order Amount)

Total: COUNT

☐ Select Distinct Values

Identifying unique values in a query

Select the Select Distinct Values check box to activate it.



Step: Select fields to include in query.

Database Fields: Customer, Customer ID, Customer Credit ID, Customer Name, Contact First Name, Contact Last Name, Contact Title, Contact Position

Query Fields: Customer.Customer Name, SUM (Orders.Order Amount)

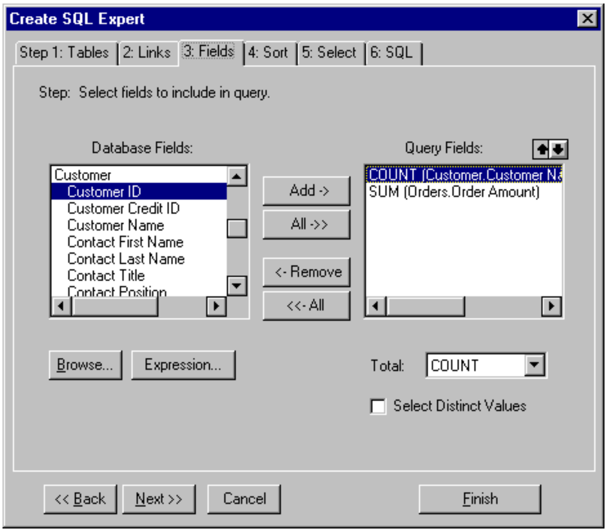
Total: NONE

☑ Select Distinct Values

Adding a SQL expression

Note: You must have some familiarity with the SQL language and SQL expressions before trying to add an expression to your query file.

1 Click **Expression**.



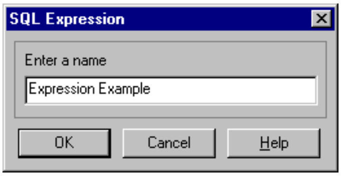
Step: Select fields to include in query.

Query Fields:
COUNT (Customer.Customer Na
SUM (Orders.Order Amount)

Total: COUNT

□ Select Distinct Values

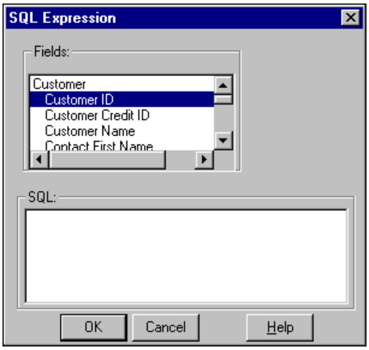
The SQL Expression dialog box appears.



Enter a name: Expression Example

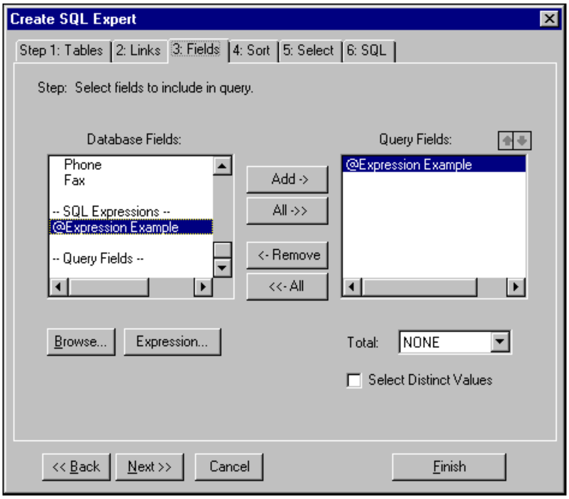
2 Type a name for the new expression, then click **OK**.

The SQL Expression dialog box appears.



3 Double-click on a field in the **Fields** area to add it to the expression. The field moves into the SQL area.

4 Edit the expression if needed.



5 Click **OK** when you have finished.

6 Select the new SQL expression.
It will be indicated by an @ sign.

7 Click **Add**.
The field is added to the Query Fields list.

Summarizing data with aggregate functions

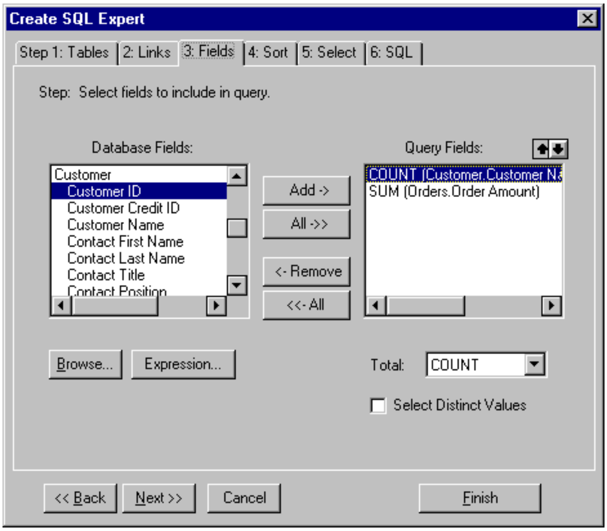
Aggregate functions

Use aggregate functions to obtain summary information on all records or on specific groups of records. Aggregate functions are most useful when you do not need the detailed information and you only want to examine totals.

For example, you might need the total number of orders made and the average amount of each order. For this type 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.

To summarize data with an aggregate function

1 Select the field you want to summarize from the **Query Fields** area.



Step: Select fields to include in query.

Database Fields:

Customer
Customer ID
Customer Credit ID
Customer Name
Contact First Name
Contact Last Name
Contact Title
Contact Position

Query Fields:

COUNT (Customer.Customer Na
SUM (Orders.Order Amount)

Total: COUNT

☐ Select Distinct Values

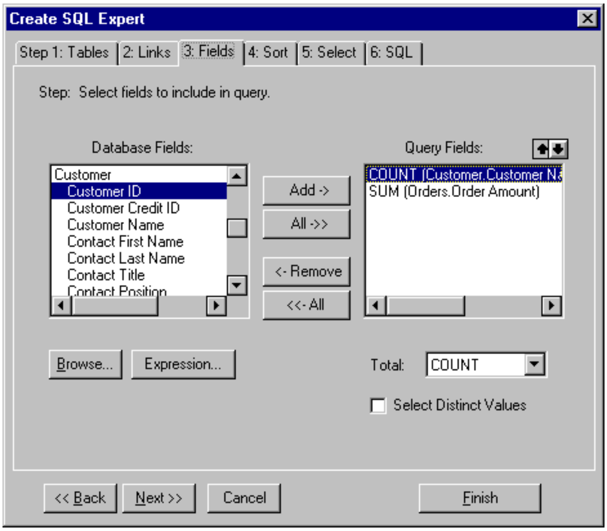
2 Choose an aggregate function to apply to the selected field from the **Total** drop-down list:

- COUNT( ) counts the number of values within a group
- SUM( ) adds the values within a group, to provide a total
- AVG( ) finds the average of all values within a group
- MIN( ) finds the minimum value within a group
- MAX( ) finds the maximum value within a group.

Any other fields that appear in the Query Fields area can be used for sorting.

Selecting groups to be included in a query

1 Select the field from the Database Fields you want to summarize.



Create SQL Expert

Step: Select fields to include in query.

Database Fields:

Customer
Customer ID
Customer Credit ID
Customer Name
Contact First Name
Contact Last Name
Contact Title
Contact Position

Query Fields:

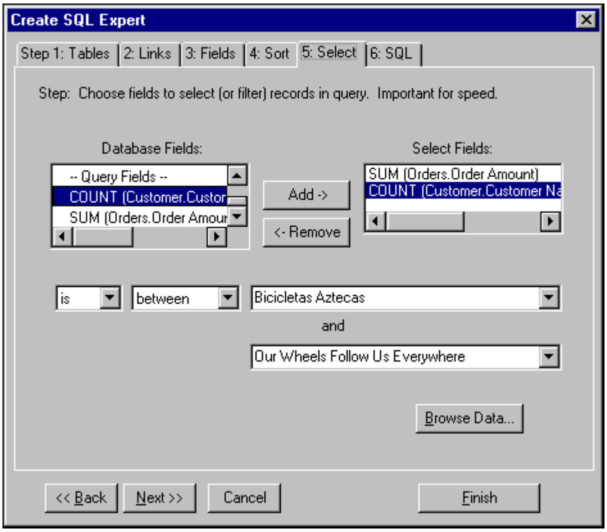
COUNT (Customer.Customer Na
SUM (Orders.Order Amount)

Total: COUNT

□ Select Distinct Values

2 From the **Total** drop-down list select the aggregate function to apply to the field.

3 Click the **Select** tab.



Create SQL Expert

Step: Choose fields to select (or filter) records in query. Important for speed.

Database Fields:

-- Query Fields --
COUNT (Customer.Custor
SUM (Orders.Order Amour

Select Fields:

SUM (Orders.Order Amount)
COUNT (Customer.Customer Na

is between Bicicletas Aztecas

and

Our Wheels Follow Us Everywhere

4 Select the aggregate function from the **Database Fields** area.

5 Click **Add**.

The field is added to the Select Fields list.

6 Use the selection criteria drop-down lists to specify which group summary values based on the aggregate function should appear in the result.

Sorting and grouping information

You can group 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.

To group by sorting data

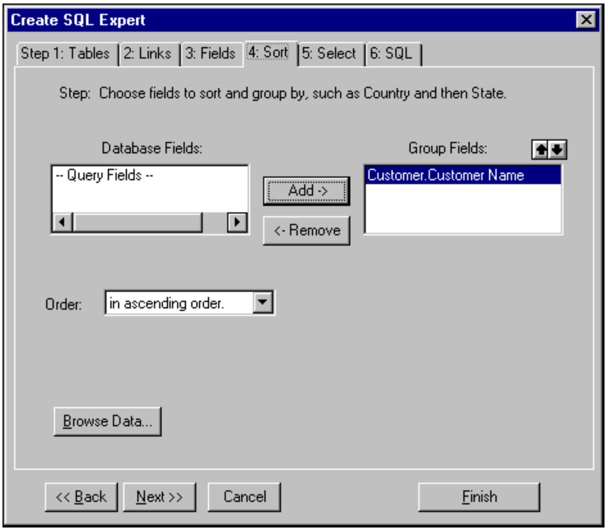
1 From the **Database Fields** area of the Sort tab, select the field you want to sort.

2 Click **Add**.

The field is added to the Group Fields area.

3 From the **Order** drop-down list, choose the order in which you want the data sorted.

- In ascending order: A-Z.
- In descending order: Z-A.
- In original order: the order that Crystal Reports receives the records. In most cases this is the order of the data from your database.


Create SQL Expert

Step: Choose fields to sort and group by, such as Country and then State.

Database Fields: -- Query Fields --

Group Fields: Customer.Customer Name

Order: in ascending order.

4 Repeat these steps for each field as required.

5 Click **Next**. The Select tab appears.

Specifying records to be included in a query

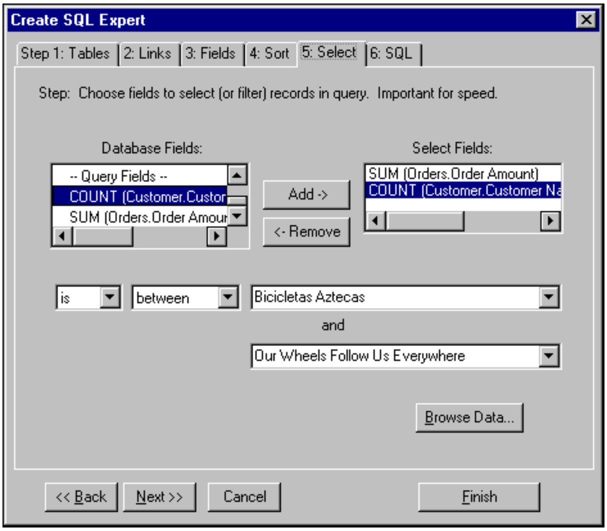
1 From the **Database Fields** area, click the field that contains the data you want to filter or specify.

2 Click **Add**.

The field is added to the **Select Fields** area.

3 Use the record selection drop-down lists to specify the filter/selection criteria to be applied to the field.

4 Repeat Steps 2 and 3 for every field in the **Select Fields** area. This will increase your report performance as only the records matching the criteria you have specified are included.



Create SQL Expert

Step: Choose fields to select (or filter) records in query. Important for speed.

Database Fields:
-- Query Fields --
COUNT (Customer.Custor
SUM (Orders.Order Amour

Select Fields:
SUM (Orders.Order Amount)
COUNT (Customer.Customer Na

is | between | Bicicletas Aztecas

and

Our Wheels Follow Us Everywhere

5 Click **Next**.

The SQL tab appears.

Editing the SQL Statement

1 Make changes to the SQL statement as required.

2 Click **Finish**.

3 Click **Yes** to process the query.

The Crystal SQL Designer window appears with the results of your query.

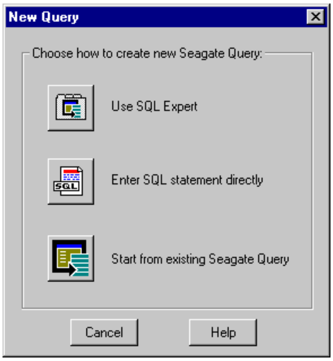
Creating a query from another Crystal Query

Accessing the Crystal SQL Designer

1 From the **Start** menu, select **Programs,** select **Crystal Reports Tools**, then click **Crystal SQL Designer**.

2 On the **File** menu, click **New**.
The New Query dialog box appears.

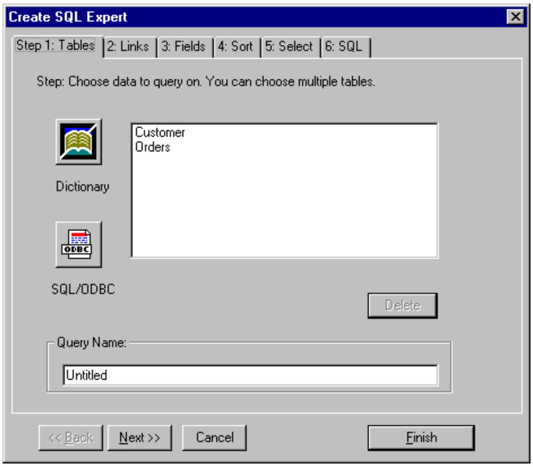
3 Click **Start from existing Seagate Query**.


New Query
Choose how to create new Seagate Query:
Use SQL Expert
Enter SQL statement directly
Start from existing Seagate Query
Cancel
Help

4 Select the query (*.QRY) file that you want to use as the base for your new query.

5 Click **OK**.
The program runs the existing SQL query and displays the data set.

6 On the **Edit** menu, click **Query**.
The Create SQL Expert appears with the specifications for the selected query already in place.



Step: Choose data to query on. You can choose multiple tables.

Customer
Orders

Query Name: Untitled

7. Use the Create Query Expert to make any changes necessary to the query.
8. Click **Finish** after completing all of the changes.
9. Click **Yes** to process the query.

   The Crystal SQL Designer window appears with the results of your query.
10. On the **File** menu, click **Save As** to specify a name for the new query.

    The new query is saved in a separate file; your source query remains unchanged.

Selecting a query for a report

1. From the **Start** menu, select **Programs,** then click **Crystal Reports**.
2. On the **File** menu, click **New**.

   The Crystal Report Gallery appears.
3. Confirm **Standard** is selected, then click **OK**.

   The Standard Report Expert appears.
4. Click **Crystal SQL Query**, from the **Data** tab.

   The Data Explorer dialog box appears.
5. Expand the Crystal SQL Query Files folder to locate the queries (.qry) file you want to use to create the new report.

   Note: Only one query file can be used with each report.
6. Click **Open**.

   The query file for your report is added to the Data tab.
7. Continue working through each of the tabs in the Standard Report Expert to finish designing the report.

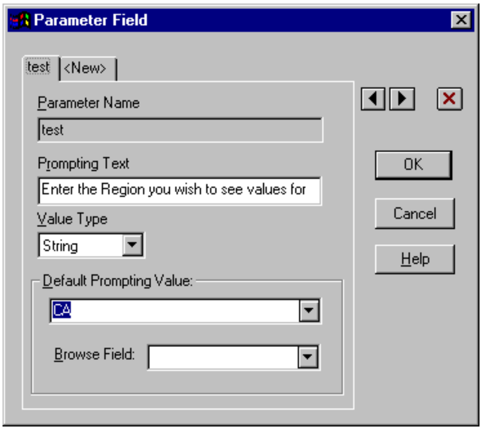
Creating a parameter field for use in a query

A parameter field can be created while previewing a query in the Crystal SQL Designer.

1. On the Edit menu in the Crystal SQL Designer, click **Parameter Field**.

   The New dialog box appears.
2. Enter the name you want to use to identify the parameter, then click **OK**.

The Parameter Field dialog box appears.



Parameter Name: test

Prompting Text: Enter the Region you wish to see values for

Value Type: String

Default Prompting Value: CA

Browse Field:

3 Enter the following information in the fields provided:

- **Prompting Text**
  Enter the text you want to appear when the program prompts you.
- **Value Type**
  Enter the data type of the parameter field.
- **Default Prompting Value**
  Enter the value you want the program to use if you do not supply a new value.
- **Browse Field**
  Use this drop-down list to specify a default field.

4 Click **OK**.

The parameter field can now be used in your query.

Using an SQL query that was designed elsewhere

1 Copy the SQL statement to the Windows Clipboard.

Most SQL editors allow the SQL statement to be copied to the Clipboard by using the Ctrl-C key combination.

2 From the **Start** menu, select **Programs,** select **Crystal Reports Tools**, then click **Crystal SQL Designer**.

3 On the **File** menu, click **New**.

The New Query dialog box appears.

4 Click **Enter SQL statement directly**.

The Log On Server dialog box appears.

5 Select the Server Type, then click **OK**.

The SQL Server Login dialog box appears.

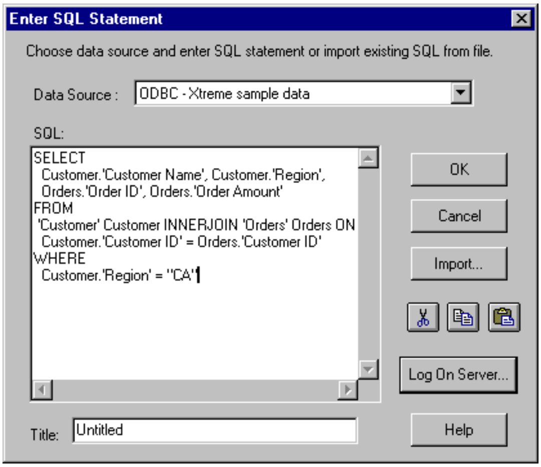
6 Enter your user ID and password to log on to the SQL server, and click **OK**.

A message appears indicating the success of your log on.

7 Click **OK**.

The Enter SQL Statement dialog box appears.

8 Enter the SQL statement in the **SQL** area by clicking anywhere inside the box and pressing Ctrl-V, or click **Paste**, to paste the SQL statement into the SQL area.


Enter SQL Statement
Choose data source and enter SQL statement or import existing SQL from file.
OK
Cancel
Import...
Log On Server...
Help

Data Source : ODBC - Xtreme sample data

SQL:

```
SELECT
  Customer.'Customer Name', Customer.'Region',
  Orders.'Order ID', Orders.'Order Amount'
FROM
 'Customer' Customer INNERJOIN 'Orders' Orders ON
  Customer.'Customer ID' = Orders.'Customer ID'
WHERE
  Customer.'Region' = ''CA'
```

Title: Untitled

From within the SQL area, you can make changes to the SQL statement. If needed, you can even delete the statement, and create a new one.

9 Click **OK**.

10 Click **Yes** to process the query.

11 On the **File** menu, click **Save** to save the new query.

The 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 a SQL statement saved in an ASCII text file. To do so, 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.

Dictionaries 30

Dictionaries are structured and simplified views of data that can be created for some or all of the individuals in an organization. When working with Dictionaries, users see only the information they need and that you want them to see. They reduce support costs and time, and reduce data misuse, loss, and damage. This chapter shows how to set up and use Dictionaries to improve organizational efficiency and security.

Dictionary 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 optional components. Data can still be accessed directly by the user. Dictionaries simply provide all of the convenience without the restrictions.

Dictionaries allow you to:

- design a single, dynamic view of all the data that is necessary to create 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
- create complex data-manipulation formulas that users can access without the need to understand formula concepts.

Note: The Formula Editor used in Crystal Dictionaries is not as complete as the one used in Crystal Reports.

Dictionaries reduce data misuse, loss, and damage. When a Dictionary is used to create a report, the only data used in the report is the data accessed through the Dictionary. You can not use a Dictionary and some other data source in the same report. Because Dictionaries are often used to impose data security, it would breach that security to allow unrestricted data access in a Dictionary report.

Note: You can include a subreport based on a different data source in a primary report based on a Dictionary.

Finally, Dictionaries provide an easy means of changing the underlying data set without changing the view of the data seen by users. You can change field and table names in the underlying data, for example. Then you remap the Dictionary to the new field and table names without changing the aliases assigned to the data. The users create their reports using the same data interface they've been using, never knowing what has changed under the surface.

Once created, the Dictionary acts as a filter, providing a view of complex data that is clear and easy for any user to understand.

Using a dictionary

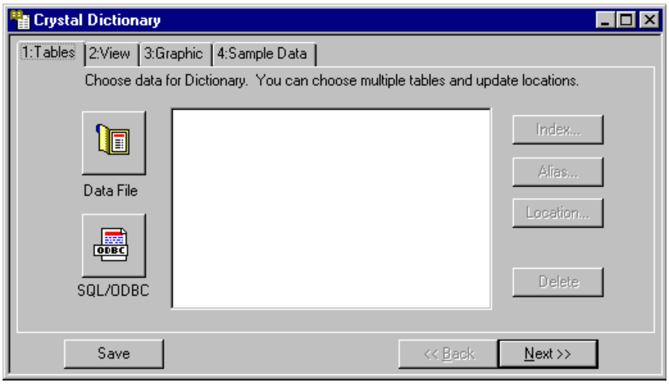
Dictionaries are often designed and distributed by Information System (IS) Managers or Network Administrators who control and manage a company's databases. These databases are often complex collections of data spread throughout several tables with hundreds or even thousands of fields. A user, trying 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 need. For example, the Accounting department's dictionary can be different from 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.

Creating a new dictionary

1. From the **Start** menu, select **Programs**, select **Crystal Reports Tools**, then click **Crystal Dictionaries**.
2. Click the **New** button on the toolbar.

   The Crystal Dictionary Expert appears.



Choose data for Dictionary. You can choose multiple tables and update locations.

This Expert contains several tabs. Each tab is numbered to lead you through the development process step-by-step.

Note: The Links Tab appears only when you have added more than one table to your dictionary.

3 To access the options on each tab, 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) tabs.

Note: Certain steps must be performed before others in the creation process. For example, you must select tables 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.

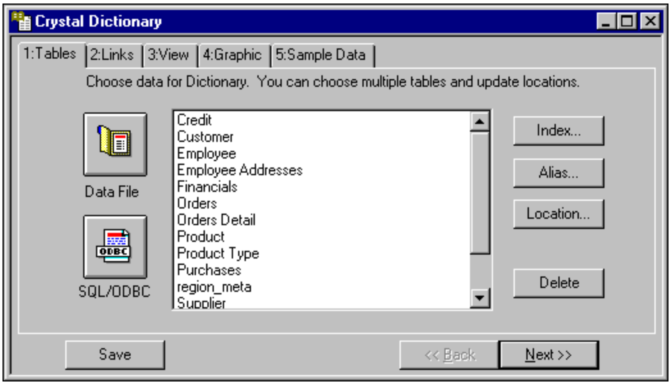
4 Click **Save** at the bottom of the Expert to save at any time.

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.

Adding tables to a dictionary

Adding a data file

1 While in the Crystal Dictionary Expert, click the **Tables** tab.



Choose data for Dictionary. You can choose multiple tables and update locations.

Credit
Customer
Employee
Employee Addresses
Financials
Orders
Orders Detail
Product
Product Type
Purchases
region_meta
Supplier

2 Click the **Data File** button.

The Choose Database File dialog box appears.

3 Use the options in this dialog box to highlight a database file.

4 Click **OK** to add the file. If the selected database file contains only one table, that table will appear in the list box on the Tables Tab.

Note:

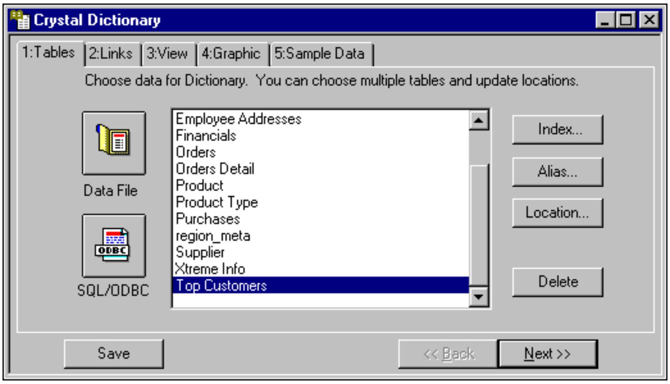
- When there is more than one table to add from the selected database file, the Select Tables dialog box appears. Highlight the tables you wish to include from the list and click OK. Only the tables you selected will appear in the list box on the Tables Tab. See "Designing the view of the data" on page 646.
- You cannot create a dictionary from two different data sources.

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 a database file is added, the tables and fields from that file will not necessarily appear to the user who opens the dictionary from the Report Designer. The tables that appear on the Tables Tab are only the tables available to the creator of the dictionary. To add specific fields, see "Designing the view of the data" on page 646.

You may also need to choose an index file to be used by one of the database tables. This is done on the Tables tab:



Choose data for Dictionary. You can choose multiple tables and update locations.

Employee Addresses
Financials
Orders
Orders Detail
Product
Product Type
Purchases
region_meta
Supplier
Xtreme Info
Top Customers

7 Highlight the database table you want to pick a specific index for.

8 Click the **Index** button.

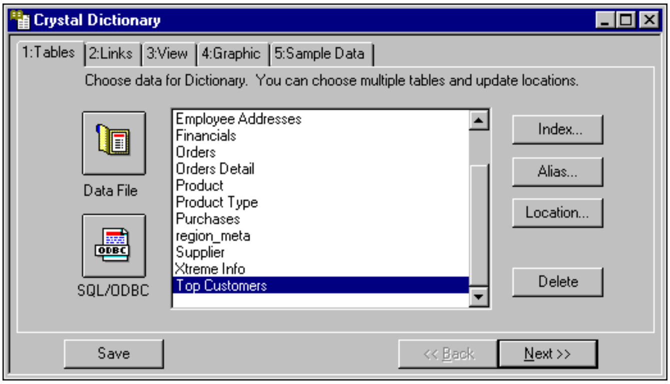
The Choose New Location dialog box appears.

9 Use the controls in this dialog box to highlight 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.

Adding an SQL or ODBC data source

1 While in the Crystal Dictionary Expert, click the **Tables** tab.



Crystal Dictionary

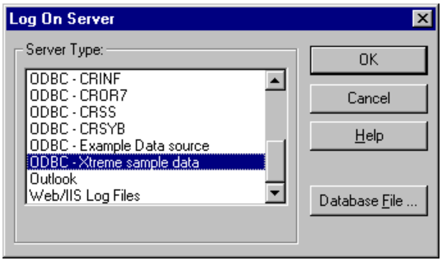
1:Tables | 2:Links | 3:View | 4:Graphic | 5:Sample Data

Choose data for Dictionary. You can choose multiple tables and update locations.

Employee Addresses
Financials
Orders
Orders Detail
Product
Product Type
Purchases
region_meta
Supplier
Xtreme Info
Top Customers

2 Click the SQL/ODBC button.

The Log On Server dialog box appears.



Log On Server

Server Type:

ODBC - CRINF
ODBC - CROR7
ODBC - CRSS
ODBC - CRSYB
ODBC - Example Data source
ODBC - Xtreme sample data
Outlook
Web/IIS Log Files

3 Select an SQL or ODBC data source and click **OK** when finished.

If the data source requires any log on information, such as user name and password, the SQL Server Login dialog box appears.

4 Log on to the ODBC data source just as you normally do from your Database Management System application.

5 Click **OK**.

The Choose SQL Table dialog box appears.

6 Highlight a database in the SQL Databases list box.

The SQL Tables list box displays all of the tables for the database you selected.

7 Highlight the table you would like to have included and click **Add** to add it to your dictionary.

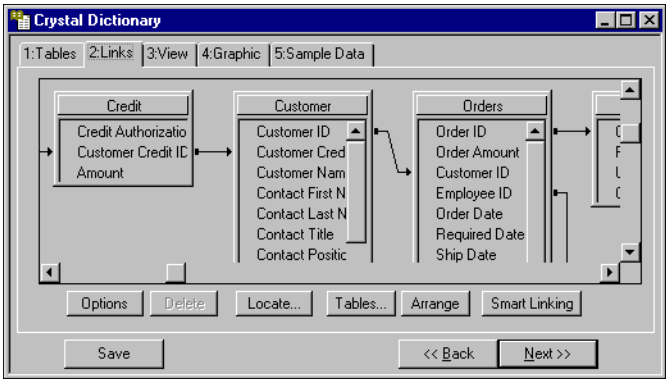
8 Repeat Step 7 for each table you want to add to the dictionary.

9 Click **Done** when finished.

Note: When an SQL or ODBC data source is added from the Tables Tab, the tables and fields from that data source will not necessarily appear to the user who opens the dictionary. The tables that appear on the Tables Tab are only the tables available to the creator of the dictionary, for adding to the dictionary. See "Designing the view of the data" on page 646.

Linking multiple tables

If you have added more than one database table to your dictionary click the Links tab in the Crystal Dictionary Expert.



Creating a link

- 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. The application will draw a new link arrow between the tables.
- If you click the Smart Linking button, the application will create logical links between tables in your Dictionary.

Changing a link

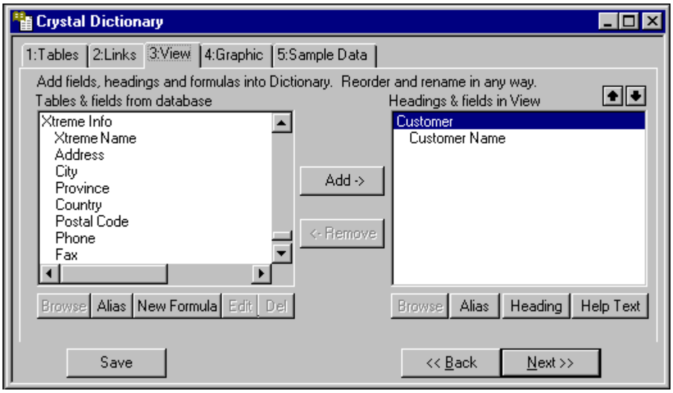
- If you select a link and click the Options button, the Link Options dialog box appears. You can use the controls in this dialog box to make any necessary changes to the highlighted link.

Designing the view of the data

Selecting tables and fields for users

After using the Tables tab to add tables to the dictionary, those tables, and the fields in them, are not automatically available to users. The View tab must be used to expose these fields. The View tab lets you design the actual view of the data that the users will see.

1 While in the Crystal Dictionary Expert, click the **View** tab.



2 Highlight the field you want to make available to users from the **Tables & fields from database** list.

3 Click the **Add** button.

The field is added to the Headings & fields in View list.

To make a table and all its fields available to users

Highlight the table name in the Tables & fields from database list box, and click Add. The table name becomes a field heading in the Headings & fields in View list box. All fields from the table appear as fields under the new field heading.

Key points

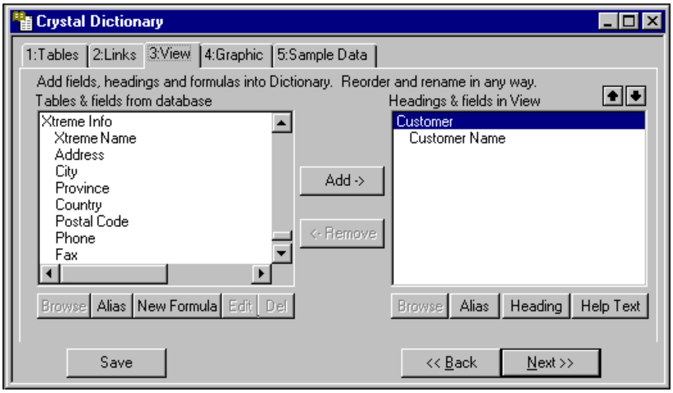
When adding and organizing tables and fields in the View Tab, keep in mind the following points:

- Tables are not displayed to the dictionary user 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 Headings & fields in View list box to provide clearer organization of data for your users. See "Adding a new field heading" on page 648.
- 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 Headings & fields in View list box and no field heading appears in the list box (the list box is empty), the application 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 Headings & fields in View list box can be renamed to anything you want. See "Renaming fields, field headings, and formulas" on page 649.

Adding formulas to the view

1 While in the Crystal Dictionary Expert, click the **View** tab.



Add fields, headings and formulas into Dictionary. Reorder and rename in any way.

Tables & fields from database:

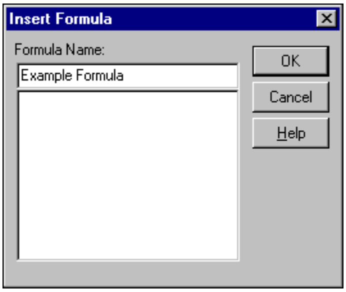
Xtreme Info
- Xtreme Name
- Address
- City
- Province
- Country
- Postal Code
- Phone
- Fax

Headings & fields in View:

Customer
- Customer Name

2 Click the **New Formula** button.

The Insert Formula dialog box appears.



Formula Name: Example Formula

3 Type a name for the formula and click **OK** when finished.

The Formula Editor appears.

4 Use the Formula Editor to create a formula for the Dictionary, just as you would create a formula for a report.

Note: The Crystal Dictionary supports formulas created in Crystal syntax only. See "Creating Formulas with Crystal Syntax" on page 437.

Not all Crystal functions are available in the formula editor in the dictionary designer.

5 Click **Accept** when finished.

The new formula is added to the Tables & fields from database list box.

6 To add the formula to the view, highlight the formula you just created in the **Tables & fields from database** list.

7 Click the **Add** button. The formula field is added to the Headings & fields in View list.

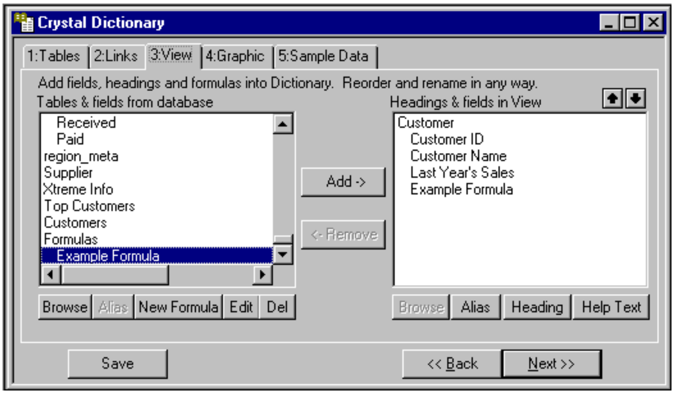
8 To modify a formula, highlight it and click the **Edit** button.

Note: Normally, formulas are represented with an @ symbol. The formula name appears in the Headings & fields in View list box without the @ symbol to hide the fact that this is a formula.

Organizing the view of the data

Adding a new field heading

1 While in the Crystal Dictionary Expert, click the **View** tab.



Add fields, headings and formulas into Dictionary. Reorder and rename in any way.

Tables & fields from database

Received
Paid
region_meta
Supplier
Xtreme Info
Top Customers
Customers
Formulas
Example Formula

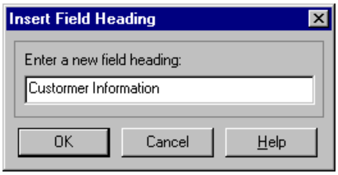
Headings & fields in View

Customer
Customer ID
Customer Name
Last Year's Sales
Example Formula

2 Highlight the field that will be the first to appear under the new field heading from the **Tables & fields from database** list box.

3 Click the Heading button.

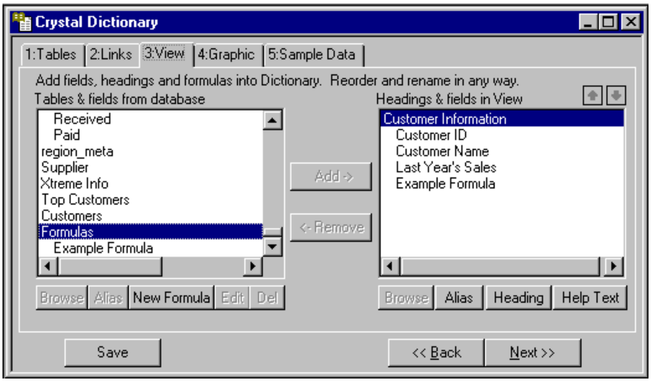
The Insert Field Heading dialog box appears.



Enter a new field heading:
Custormer Information

4 Type the new name into the text box and click **OK** when finished.

The new field heading appears in this list, as specified.



Add fields, headings and formulas into Dictionary. Reorder and rename in any way.

Tables & fields from database

Received
Paid
region_meta
Supplier
Xtreme Info
Top Customers
Customers
Formulas
Example Formula

Headings & fields in View

Customer Information
Customer ID
Customer Name
Last Year's Sales
Example Formula

Moving fields/field headings

1 While in the Crystal Dictionary Expert, click the **View** Tab.

2 Highlight the heading or field you want to move from the **Headings & fields in View** 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.

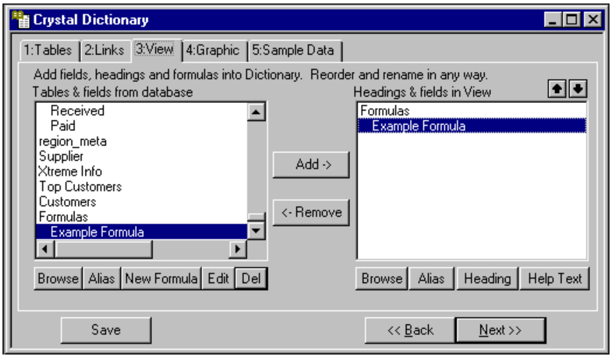
Note: When organizing fields and field headings, remember that field headings take the place of database tables. Because of this organization, the first item in the list box must be a field heading.

Renaming fields, field headings, and formulas

1 While in the Crystal Dictionary Expert, click the **View** Tab.

2 Highlight the heading or field you want to rename.

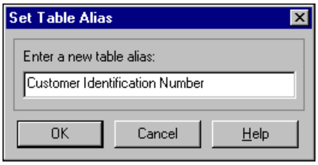
3 Click **Alias**.



Add fields, headings and formulas into Dictionary. Reorder and rename in any way.

If you chose a heading, the Set Table Alias dialog box appears.

If you chose a field, the Set Field Alias dialog box appears.



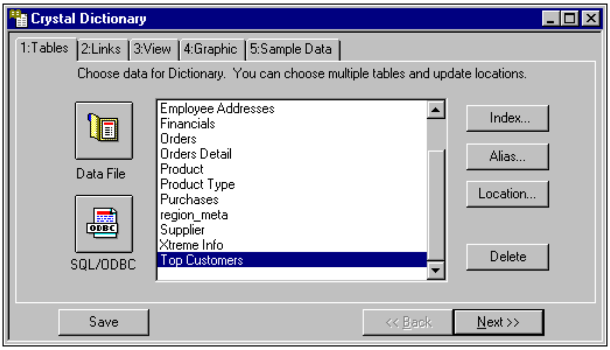
Enter a new table alias: Customer Identification Number

4 Type the new name into the 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 is changed. The original database file is not affected.

Updating the location of a database table

1 While in the Crystal Dictionary Expert, click the **Tables** tab.



Choose data for Dictionary. You can choose multiple tables and update locations.

2 Highlight the table or field that has changed name or location from the list box.

3 Click the **Location** button.

Note: Remember, the table names that appear here are aliases rather than the actual database table names.

If the table is from a data file, the Choose New Location dialog box appears.

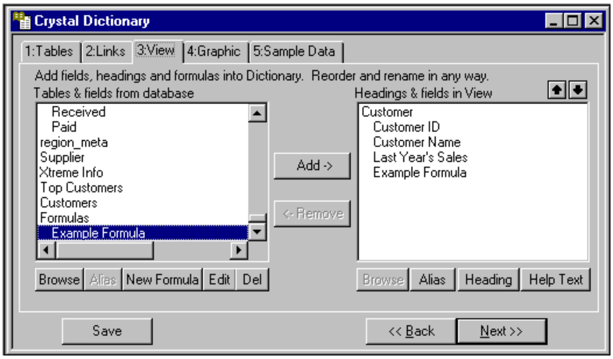
If the table is from an SQL server or other ODBC data source, the Choose SQL Table dialog box appears.

4 Highlight the new name or location of the database table from the dialog box that appears, and click **OK**.

Adding 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 Crystal Dictionary Expert, click the **View** tab.



Add fields, headings and formulas into Dictionary. Reorder and rename in any way.

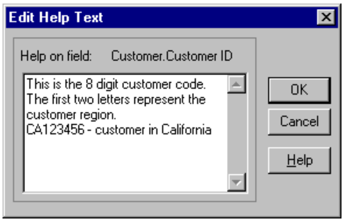
Tables & fields from database: Received, Paid, region_meta, Supplier, Xtreme Info, Top Customers, Customers, Formulas, Example Formula

Headings & fields in View: Customer, Customer ID, Customer Name, Last Year's Sales, Example Formula

2 Highlight the field that you want to add Help text to from the **Tables & fields from database** list box.

3 Click the **Help Text** button.

The Edit Help Text dialog box appears.



Help on field: Customer.Customer ID

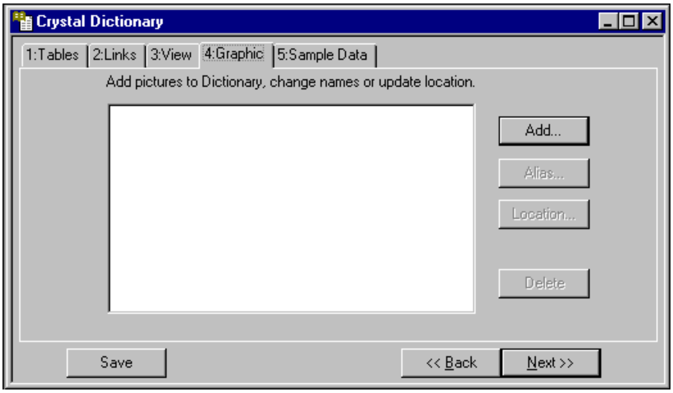
This is the 8 digit customer code.
The first two letters represent the customer region.
CA123456 - customer in California

4 Enter the Help Text and click **OK** when finished.

When the user selects a field or field heading and holds the cursor over the selected item, the Help text appears in a pop-up window.

Adding graphics to the dictionary

1 While in the Crystal Dictionary Expert, click the **Graphic** tab.



2 Click the **Add** button.

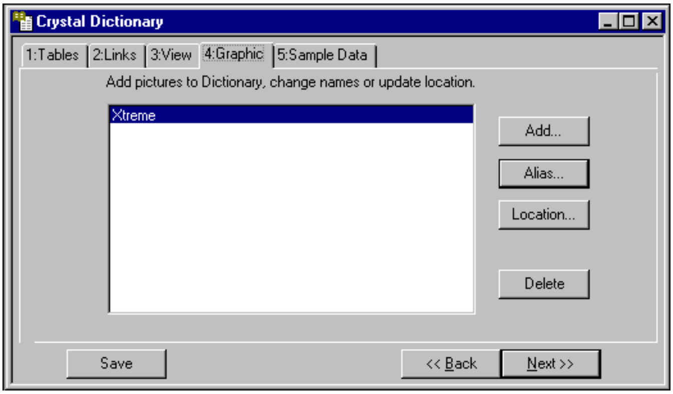
The File Open dialog box appears.

3 Use the controls in the File Open dialog box to highlight a graphic image file, and click **OK** to add it to the dictionary.

Dictionaries support the Windows bitmap (.bmp), PC Paintbrush (.PCX), Tiff (.TIF), JPEG (.jpg, .jpeg) and TARGA (.TGA) graphic formats. The image file appears in the list box of the Graphic Tab.

Organizing graphics

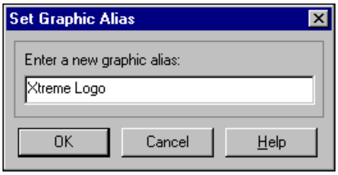
You may now want to change the name, update the location, or delete a graphic. Highlight the desired graphic and click the appropriate button for your needs.



1 Highlight the desired graphic in the Add pictures to Dictionary, change names or update location list box.

2 Click the Alias button to change the name,

- click **Delete** to delete the graphic from the list, or
- click the **Location** button to change the location.
- If you clicked the **Alias** button, the Set Graphic Alias dialog box appears.



Xtreme Logo

3 Type a new name for the graphic in the **Enter a new graphic alias** text box and click **OK** when finished.

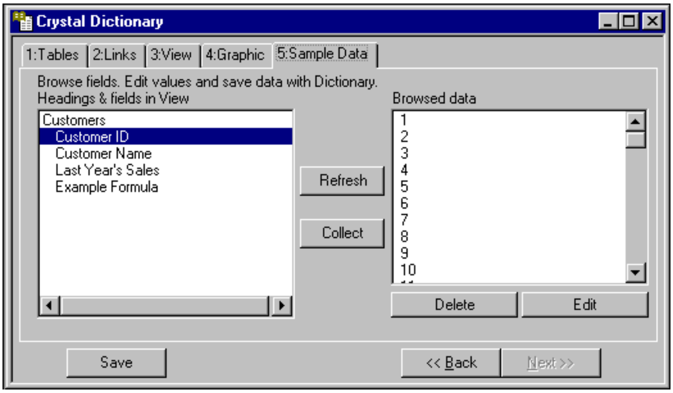
If you clicked the Location button to set the new location of the image, the Choose New Location dialog box appears. Use this dialog box to find the new name and/or location of the graphic file.

Creating 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 Crystal Dictionary Expert, click the **Sample Data** tab.

2 If the data has changed, click the **Refresh** button to retrieve a list of the new data.



Browse fields. Edit values and save data with Dictionary.

Headings & fields in View

Customers
Customer ID
Customer Name
Last Year's Sales
Example Formula

Browsed data

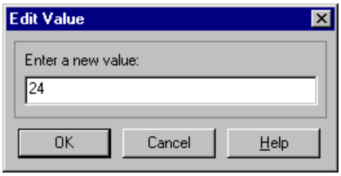
1
2
3
4
5
6
7
8
9
10

3 Highlight a field that you want to make browseable from the Headings & fields in View list.

4 Click the Collect button. The selected values will be added to the **Browsed data** list.

5 To modify any of the browseable data, highlight a value, then click either **Delete** to delete the value or **Edit** to change it.

If you clicked the **Edit button**, the Edit Value dialog box appears.



Edit Value

Enter a new value:

24

6 Enter the new value in the text box and click **OK** when finished.

The new value appears 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. This allows you to customize the look of the data when users browse data, without actually changing the data that is reported on.

Working with an existing dictionary

Accessing a dictionary for your report

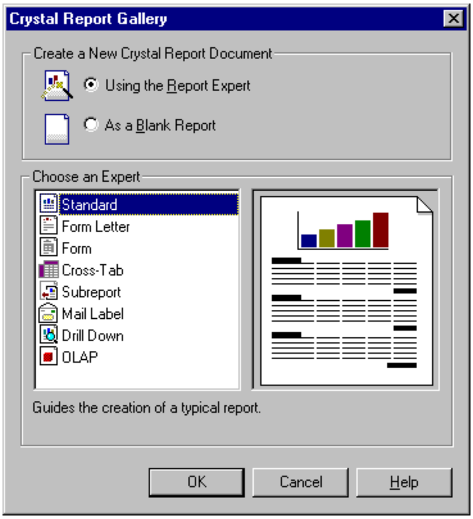
When creating a new report in Crystal Reports, you can specify a dictionary as your data source. By using a dictionary for your report, administrators can monitor company information being disclosed and users are not burdened by extra data they will not be using in their reports.

Note: You cannot 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.

To access a dictionary

1. Click the **New** button on the Standard toolbar in Crystal Reports.

   The Crystal Report Gallery appears.



Crystal Report Gallery

Create a New Crystal Report Document

● Using the Report Expert

○ As a Blank Report

Choose an Expert

Guides the creation of a typical report.

2. Click **Using the Report Expert** button.
3. Select the type of report you want to create from the **Choose an Expert** list.
4. Click **OK**.
5. In the Report Expert dialog box, click **Crystal Dictionary**.
6. Select the dictionary you want to use.

   You return to the Report Expert. The field headings of the data in the dictionary are listed on the right.
7. Click the **Fields** tab.

   The database fields from the dictionary are listed.
8. Select the database fields you want to include in your report.
9. Click **Add**.
10. Click **Next** to move to the next folder to have the report expert help you sort the data.
11. Continue to click **Next** if you want the Report Expert to help you design the rest of your report.
12. Click **Design Report** to access your report.

    The Crystal Report Expert closes and creates a report based on your specifications.

Editing an existing dictionary

1 Click the **Open** button on the Crystal Dictionaries toolbar.

The File Open dialog box appears.

2 Choose the existing dictionary file and click **OK**.

The Crystal Dictionary Expert appears.

3 Use the Expert to make changes to the dictionary file.

Converting a 4.x or 5.x dictionary file

1 Click the **Open** button on the toolbar.

The File Open dialog box appears.

2 Select the old dictionary (*.DC5) file and click **OK**.

The Select View dialog box appears, listing the names of all views from the old dictionary file.

3 Highlight the view you want to convert to a new dictionary file, and click **OK**.

The Crystal Dictionary Expert appears with the data from the view you selected.

Note: This version of Crystal Dictionaries provides an easier and more powerful method for controlling data access than earlier versions. However, each view from an older dictionary file is handled as a separate dictionary in this version. An older dictionary file, on the other hand, held several views. To convert an entire dictionary 4.x or 5.x file, you must open each view in the file separately and save it as a new dictionary file.

4 Use the Expert to make changes to the new file.

5 When the file is saved, it will be saved in the new dictionary format (*.DC5).

Report Processing Model A

This appendix provides you with in-depth information about the Report Processing model. This model determines the order in which data is accessed and manipulated during report generation.

Overview

Crystal Reports uses a three-pass reporting method to generate reports. The sections below describe what happens during each step of this process. To see a visual representation, refer to the flow-chart at the end of this appendix.

What is a “pass”?

A pass is a process that Crystal Reports uses each time the data is read and manipulated. Depending on the complexity of the report Crystal Reports may make 1, 2, or 3 passes over the data. This feature allows for complex reporting and formula manipulation.

Pre-pass 1

When previewing a report, the first elements to be evaluated are “constant” formulas. Constant formulas are those that have a constant value for the entire report. They do not change from record to record. For example, 100* 30 would be a constant formula. Constant formulas are evaluated at the beginning of the print generation process and are never evaluated again. This process is known as “BeforeReadingRecords.” If you were to place a constant formula field (i.e., 100*30) in the Details section, the result would be 3000 for each record displayed.

Pass 1

After the “BeforeReadingRecords” process has taken place, Crystal Reports begins reading the database records. During the record reading process, the following will occur:

- Record retrieval. Where possible record selection and sorting are pushed down to the database in this step.
- Evaluation of recurring formulas. These formulas are those that contain database fields but do not contain references to subtotals or summary information. This evaluation time is known as “WhileReadingRecords.”Formulas that contain references to subtotals or summary information are processed in the second pass.
- Application of the record selection locally. If the record selection is too complex to be pushed down to the database, it is applied by Crystal Reports in this step.
- Sorting, grouping, and totaling. In this step, Crystal Reports sorts the records, separates them into groups, and then calculates the subtotals and summaries needed for each group.

- Cross-Tab generation.
- Storage of saved data. After the totaling process is complete, all of the records and totals are stored in memory and to temporary files. Crystal Reports does not read the database again, but instead uses this saved data during all subsequent processing.

Pre-pass 2

During Pre-Pass 2, Crystal Reports orders the groups in the report for Top/Bottom N or Hierarchical Grouping. The records are not read in this process, instead Crystal Reports only looks at group instances from Pass 1, and takes the Top N as appropriate, or orders the groups based on the Hierarchical Grouping settings specified.

Pass 2

Crystal Reports enters the second pass through the data to format pages. The pages are formatted on demand. This means that Crystal Reports will not format a page until it is requested by the user, or it is required for the total page count in Pass 3.

During page formatting, Crystal Reports does the following:

- Group selection formula
- Running totals
- Calculation of formulas marked “WhilePrinting Records”.

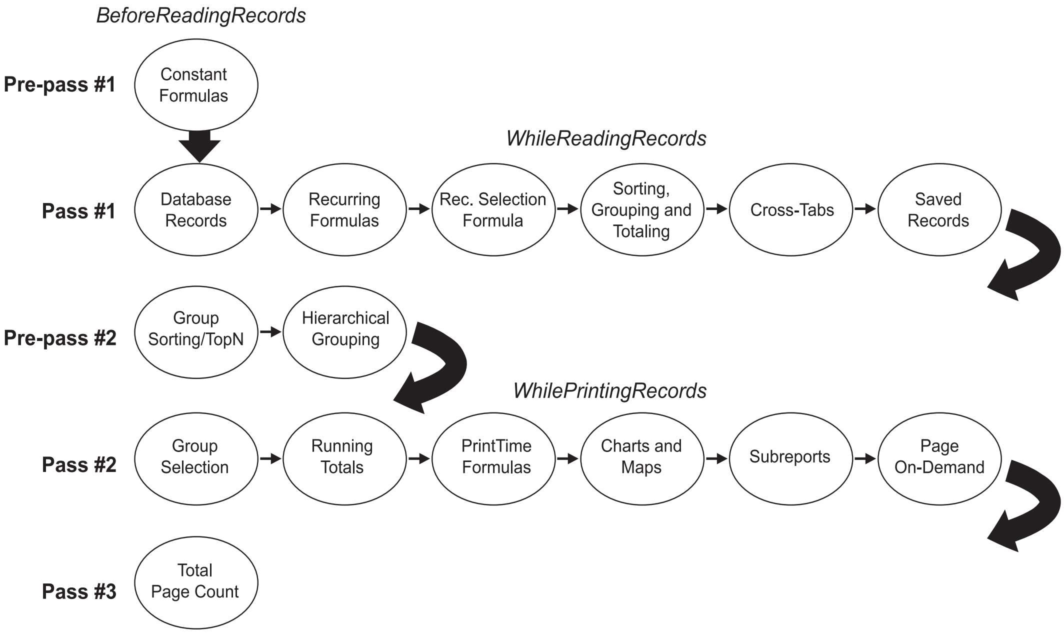
  These are formulas that contain references to subtotals or summary information, also known as “PrintTime” formulas. This evaluation time is known as “WhilePrinting Records.”
- Charts and maps
- Subreports
- Generate Pages on Demand.

Note: Subtotals, grand totals, and summaries may be incorrect if the report has a group selection formula. This occurs because the grand totals and summaries are calculated during Pass 1, but the group selection formula filters the data again in Pass 2. Running total fields can be used instead of summaries to total data in reports with a group selection formula.

Pass 3

In the third, and final pass, the total page count is determined. This applies to reports that use the total page count, or Page N of M, special variable fields.

Multi-pass reporting flow chart



Note: Although subreports appear in Pass 2 in the flow chart, you can use on-demand subreports to ensure that your main report remains a single-pass report. With on-demand subreports, Crystal Reports must still make a second pass through the data; however, this second pass does not begin until you drill down on the subreport. Consequently, you can increase the performance of reports that contain subreports by using on-demand subreports.

International Office Directory

B

This appendix provides contact information by region. For technical support information, see the Technical Support Guide.

International Office Directory

The following section lists the contact information for Sales and Product Support at each Seagate Software office, worldwide.

For more information, visit our web site at `http://www.seagatesoftware.com`

North and South American Offices

Canada and USA - North & South American Head Office

Seagate Software
840 Cambie Street
Vancouver, BC V6B 4J2
Canada

Sales and General Inquiries

Telephone: 604 681 3435

Toll-Free: 1 800 877 2340

Fax: 604 681 2934

Latin America

Sales and General Inquiries

Telephone: 604 681 3435

Fax: 604 681 2934

Asia/Pacific Offices

Australia

Seagate Software Pty Ltd.
Level 9, 52 Alfred Street
Milsons Point, NSW 2061
Australia

Sales and General Inquiries

Telephone: +61 2 9955 4088

Fax: +61 2 9955 7682

Hong Kong

Seagate Software (HK)
Suite 2603, 26th Floor
Universal Trade Center
3 Arbuthnot Road, Central
Hong Kong

Sales and General Inquiries

Telephone: +852 2575 2576

Fax: +852 2893 2727

Japan

Seagate Software
3F, Bridgestone Bldg
2-13-12 Hirakawa-cho,
Chiyoda-ku
Tokyo 102-0093

Sales and General Inquiries

Telephone: +81 3 5226 3601

Fax:+81 3 5226 3606

Email: sales_jp@img.seagatesoftware.com

Malaysia

Seagate Software
(a subsidiary of Seagate Technology)
Level 36 Menara Citibank
165 Jalan Ampang
Kuala Lumpur 50450
Malaysia

Sales and General Inquiries

Telephone: 603 2169-6307 or 603-2169-6308

Fax: 603-2169-6168

Email : southasia@seagatesoftware.com

Singapore

Seagate Software
14 Science Park Drive
#03-02 The Maxwell
Singapore Science Park
118226 Singapore

Sales and General Inquiries

Telephone: +65 777 0533

Fax: +65 777 8786

Europe

United Kingdom - EMEA and Northern European Head Office

Seagate Software
The Broadwalk
54 The Broadway
Ealing,
London, W5 5JN
United Kingdom

Seagate Software
St. Matthews Court
4 Civic Drive
Ipswich, Suffolk, 1P1-2QA
United Kingdom

Sales and General Inquiries

Telephone: +44 (0) 208 566 2020

Fax: +44 (0) 208 231 0600

Austria

Seagate Software
Frankfurter-Str. 21-25
D-65760 Eschborn
Germany

Sales and General Inquiries

Telephone: +49 (0) 69 9509 6310

Fax: +49 (0) 69 9509 6314

Belgium

Seagate Software
62 bis, Avenue Andre Morizet
Boulogne-Billancourt Cedex
Paris, F-92643
France

Sales and General Inquiries

Telephone: +33 (0) 155 174 082

Fax: +33 (0) 155 174 086

France - SEMEA. Southern Europe and Middle East Head Office

Seagate Software
62 bis, Avenue Andre Morizet
Boulogne-Billancourt Cedex
Paris, 92643
France

Sales and General Inquiries

Telephone: +33 (0) 155 174 082
or: +33 (0) 1 41 10 16 00

Fax: +33 (0) 155 174 086
or: +33 (0) 1 41 10 16 57

Germany - Central European Head Office

Seagate Software GmbH
Frankfurter-Str. 21-25
D-65760 Eschborn
Germany

Sales and General Inquiries

Telephone: +49 (0) 69 95 09 63 10

Fax: +49 (0) 69 95 09 63 14

Ireland

The Broadwalk
54 The Broadway
Ealing, London
W5 5JN England

Sales and General Inquiries

Telephone: +44 (0) 208 566 2020

Fax: +44 (020) 208 231 0600

Italy

Seagate Software
Via Conservatorio 22
20122 Milano
Italy

Sales and General Inquiries

Telephone: +39 027 729 310

Fax: +39 02 772 940

Netherlands

Seagate Software
Hojel City Centre
GebouwD, 4e etage
Graadt Van Roggenweg 328
Postbox 19127

3501 DC Utrecht
Netherlands

Sales and General Inquiries

Telephone: +44 (0) 208 566 2020

Fax: +44 (0) 208 231 0600

Spain

Seagate Software
Paseo de la Castellana 93-4a
28046, Madrid
Spain

Sales and General Inquiries

Telephone: +34 91 555 5198

Fax: +34 915 559 957

Sweden

Seagate Software
Kanalvagen 10C
Upplands Vasby
194 61 Sweden

Sales and General Inquiries

Telephone: +46 (0) 858 771 171

Fax: + 46 (0) 858 771 172

Switzerland

Seagate Software
World Trade Centre
PO Box 112
Leutschenbachstrasse 95
CH-8050 Zurich
Switzerland

Sales and General Inquiries

Telephone: +41 1 308 3922

Fax: +41 1 308 3500

Africa and Middle East

South Africa

Seagate Software
Sandown Village Office Park
East Court
1 Gwen Lane cnr Maude Street
Sandton 2146
South Africa

Postal Address

Seagate Software
PO Box 78720
Sandton 2146
Republic of South Africa

Sales and General Inquiries

Telephone: +27 11 3059700

Fax: +27 11 3059702

Glossary

absolute formatting

Formatting that is always applied to an object. See also *conditional formatting*.

access

To access data means to retrieve data.

Acrobat Reader

An application used to review and print online manuals.

active database

An active database is a database that has been selected for use in a report. You activate databases with the New command on the File Menu and the Add Database to Report commandon the Database Menu.

Active Server Page

Active Server Pages are web pages that run under Microsoft's Internet Information Server (IIS) version 3.0 and later. Active Server Pages combine HTML, VBScript or JScript, and ActiveX controls to create dynamic web pages that can be viewed from any of the most popular web browsers. The Quik Reports for Windows Engine Automation Server, the Crystal Design-Time ActiveX Control, and the Crystal Active Data Driver can be combined to create active web sites in Microsoft Visual InterDev that display reports from an Internet or intranet site.

ActiveX Control

A Custom Control for Visual Basic 4.0 and above that incorporates the Object Linking and Embedding (OLE) technology. Formerly known as an OLE Control (OCX).

aggregate functions

An operation that summarizes data (sums, calculates an average, identifies a maximum value, etc.). The term "aggregate functions" is often associated with SQL data sources.

alias

An alias is an alternative name assigned to a database. If a database is called CUSTOMER.DB, for example, you can 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 related 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, can not operate by itself. It needs an argument that identifies the item to be truncated. Thus, in the formula in Crystal syntax:

```
Truncate ({orders.ORDER AMOUNT})
```

«where Truncate is the function and {orders.ORDER AMOUNT} is the argument, the value of the {orders.ORDER AMOUNT} field is the item that needs 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 used to show whether or not fields are indexed fields.

attribute

An attribute is a quality applied to an object (i.e., font size, color, etc.).

auto arrange

When you place a field on your report, the program 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. To create neat-looking reports, it is often necessary to resize the fields so that 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 Analyzer menu.

bitmap

A graphic file that can be added to a report.

BLOB fields

A BLOB field is a database field whose data consists of Binary Large Objects—such as bitmap graphics, images, OLE objects, metafiles, and so on. Inserting a BLOB field into your report allows you to access these binary objects 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
    “”
```

This Crystal syntax formula 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

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 the program returns a number, the value of {orders detail.QUANTITY} multiplied by 6. All selection formula and group selection formulas must be Boolean.

browser

A browser is an application that enables viewing of documents in HTML format.

calculated data field

A calculated data field is a field that holds a value that comes from a calculation rather than 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 by using a calculated data field. To create a calculated data field, you create a formula that subtracts {file.COST} from {file.SALES}. The formula calculates a Gross Profit value for each row and prints that value wherever you place the formula field.

case-sensitive

Case-sensitive means that a program differentiates between uppercase and lowercase letters when evaluating a text string. Thus, 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," and similar mixed-case strings. Crystal Reports operators (Equal, In string, etc.) are case-sensitive.

column

A column is the display of data from a single field or formula. Columns run up and down the page. Compare with row.

comments

Comments are blocks of descriptive text that accompany formulas. Crystal Reports ignores comments when it runs the formula.

concatenate

To concatenate is 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 *If x<5 Then x Else 5*, the expression *x<5* is the condition.

conditional formatting

Conditional formatting is formatting that applies only if certain situations occur. For example, you can conditionally format numeric database fields to display in red when negative.

conditional formatting formula

A conditional formatting formula is an expression that applies specific attributes to objects or sections only if certain criteria are met.

conditional properties

Conditional properties are properties that are performed on an object only if a comparison statement returns a value of True.

confidence threshold

A percentage (*C*) used when the Document Import Tool converts a report. If fewer than *C* percent of the fields are located correctly, the conversion will fail.

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 If *x<5 Then x Else 5*, the expression Then *x* is the consequence.

constant

A constant is a value that is fixed and unchanging as opposed to a variable value, which can take on various values depending on the circumstances.

The value 5 is a constant; the value of the Quantity field (which may be 5 sometimes, but may be a different number at other times) is a variable value. For example, in the formula for converting pounds to ounces (Ounces = Pounds * 16), *16* is a constant while *Ounces* 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 (i.e., 14-9).

container document

A file that contains an embedded or linked OLE object.

conversion interface file

A conversion interface file (.cif extension) is the file in which the program saves the formatting and highlighting from the Document Import Tool. When you highlight something in the Document Import Tool and set properties for database fields, these settings are saved in the .cif file. This file can be used to quickly format the same report later, without highlighting.

cross-tab

A cross-tab is a report that summarizes and presents data in a compact row and column format that makes it easy to compare data and identify trends.

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).

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 report.

data source

A data source is a database, table, query, dictionary, Info View or stored procedure result set that provides the data for a report.

data type

A data type is a classification of the data that appears in a field or formula. Each piece of data used in a report or formula has one of the following data types: string, currency, number, date, date/time, time, 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), the program uses a different set of calculation rules for one type of data than it uses for another.

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 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 prevent a system error, the program refuses to print a report which contains a formula that divides a value by zero.

drag

Drag can mean different things, depending on the context in which the word is used:

- When referring to moving a field, drag means to click the field frame and, while keeping the mouse button pressed, move the frame to a new position. You release the mouse button when the field is in the desired position.
- When referring to resizing a field, drag means to click one of the field frame handles and, while keeping the mouse button pressed, make the field bigger or smaller. You release the button when the field is the desired size.
- When referring to formatting text, drag means to highlight text by moving the I-beam cursor across it while the mouse 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 an application 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 Report Engine can be called as a DLL by developers for use with applications they are developing.

element

The word element is used in the documentation to describe individual report components such as database fields, formulas, and group fields. The Design Tab uses rectangular frames to represent fields.

embed, embedded object

An embedded object contains a presentation of the object, all of the data pertaining 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 neither month, day, or year, and thus does not print. 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 a non-printing (empty) date 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 typically printed when a value does not meet a specific condition in a numeric If-Then-Else formula. 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 this event, nothing gets printed in the case of a zero value.

empty string

An empty string (designated as ““) is a string that contains no characters. 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 stage in the reporting process when a formula gets evaluated. The three evaluation time functions are:

- BeforeReadingRecords
- WhileReadingRecords
- WhilePrintingRecords.

Expert

Crystal Reports offers you several Experts. Experts are tools designed to 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 through the steps. When you have completed the final step, the Expert will do the rest of the work.

Note: The term "Expert" in Crystal Reports is equivalent to the term "wizard" in Microsoft Access.

export

To export is to distribute your report to a disk file or through email. Crystal Reports allows 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 it can contain a value. Field data is typically displayed/printed in columns in the Details section of a report.

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 Trim functions for removing excess white space from field values that do not fill their respective fields.

file

A file is a collection of related data stored under one 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 user's attention. For example, in an accounts receivable report, the words "past due" might be printed as a flag beside every past due account.

footer

The footer is the text that appears at the bottom of a report page. Footer text frequently includes page numbers, and may include other information that describes or identifies the report. Crystal Reports gives you the option of printing the footer on all pages or on only selected pages of your report.

formatting toolbar

This toolbar displays buttons that you can click to perform many common formatting tasks.

form letter

In Crystal Reports, a form letter is a letter that can be reproduced, personalized, and customized using the program's 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 modifications you want performed on certain data before it is printed on your report.

For example, if your report is to contain a {file.SALES} field and a {file.COST} field, you may want to create a GrossProfit field and designate its text strings as {file.SALES} - {file.COST}. This is a simple formula that tells the program 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 value into a single string, and to perform a multitude of other operations. Creating a formula in Crystal Reports is much like creating a formula in your favorite spreadsheet.

Note: The term "formula" in Crystal Reports is equivalent to the term "expression" in Microsoft Access.

Formula Editor

The Formula Editor is 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 syntax

Formula syntax is the set of grammar rules you are required to follow when creating formulas using the formula language.

free form

Free form implies that placement of objects is not limited to grids (vertical or horizontal).

function

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 operators built into the function without needing each operator specified separately. In this way, a function is a kind of shorthand that makes it easier and less time consuming for you to create reports.

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.

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 the *Snap to Grid* option toggled on in the 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 might consist of all those customers living in the same Zip Code, or in the same Region. In a sales report, a group might consist of all the orders placed by the same customer, or all of the orders generated by a particular sales representative. Crystal Reports offers you a great deal of flexibility in the way you group the data on a 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.

guidelines

Guidelines are non-printing lines that you can use to align, move, and resize objects with precision. Guidelines allow you to work in a free form environment (without a grid), while retaining 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 items 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 on only selected pages of a report.

Highlighting Expert

The Highlighting Expert is most commonly used for highlighting number or currency field values that are in some way distinguished from the other values in the report. The Highlighting Expert offers a wide range of conditional formatting, including font color, background color, and border style.

You can think of the Highlighting Expert as an advanced formula editor that runs the following equation: If Condition is True, Then Apply These Formatting Specifications.

HTML

The language used by the World Wide Web to publish linked web pages on the Internet.

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, the program 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 Section Expert.

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 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, the program 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.

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 equivalent to the term “relationship” in Microsoft Access.

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.

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.

Map Expert

The Map Expert is a powerful tool for better organizing your data by placing geographic maps on a report. You can drill-down on these maps to see the underlying data.

Microsoft Foundation Class

The Microsoft Foundation Class (MFC) is 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 employee’s 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 arithmetical calculations. 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. The program uses the following kinds of objects:

- field
- text
- cross-tab
- graph
- subreport
- picture
- OLE.

Each of these objects can be individually formatted, moved, resized, duplicated, and so on.

object frame

An object 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 object frame appears. When you move the frame to the place in the report you want the field or formula to appear and click the button, the program inserts the item at the point specified.

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). 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.

on-demand subreport

On-demand subreports appear only as hyperlinks in the main report.

The actual data is not read from the database until the user drills down on the hyperlink. Only data for the on-demand subreport that is actually viewed will be retrieved from the database. On-demand subreports may also be called "real time" subreports in some cases.

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.

operator

An operator is a special symbol that describes an operation or an action to take place between two or more values.

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, among others.

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 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 equivalent to the term “parameter queries” in Microsoft Access.

paste

Paste means to retrieve and place data from the Clipboard into a report or formula. The data may have been cut from the same report or formula or from a different one.

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 20 houses in one year. The population of houses sold by that agent in that year is 20.

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 the program.

population variance

Population variance is the square of the 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 the program.

properties

Properties are qualities that define the appearance or action of an object or a section. There are two kinds of properties:

- **On/Off**
  A property that can only be toggled on or off.
- **Attribute**
  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

The Report Footer section is the last section of a report in the Design Tab. You can place a summary in this section that you want to appear only on the last page of a report.

Report Gallery

The Report Gallery is a special dialog box that appears when you click the **New** button on the standard toolbar or choose the New command from the File menu. The Report Gallery serves as a gateway to all the report creation experts and to the graphical interface for selecting the report and data type for creating custom reports.

Report Header

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

A request is a set of criteria that specifies 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 can create a record selection request that retrieves only California records for your report. You create record and group selection requests using the Section Expert.

returns

The word “returns” refers to the result of a function, an operator, or a formula. For example:

- When using 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 1.2345.
- When using an operator, the result of the operation using that operator 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 using 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 following 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

A row is the display of data from a single record. Rows run horizontally across the page. The words row and record are sometimes used interchangeably in this manual. Contrast with column.

ruler

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.

The Ruler is visible in both the Design Tab and Preview Tab when their respective check boxes are toggled on in the Options dialog box.

running total

A running total is a total that is 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 a report.

select

- With regard to a report element (data field, formula, etc.), select means to point to the element and then click 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 server document is a file that stores the original OLE object.

server-side processing

Server-side processing is a feature that allows you to set up reports that perform the majority of their processing on the server. These reports push only relevant details to your computer, thus saving you time and memory.

shortcut menu

A dynamic menu available in the Design and Preview Tabs. Access the shortcut menu by highlighting an object 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-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.

For example, you may have a customer report that contains the {customer.CUSTOMER ID} and {orders.ORDER AMOUNT} fields. If you want to subtotal by customer (total the orders for each customer), 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 can also use sort-and-group-by fields to trigger summaries.

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 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} 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. The report could also be sorted in ascending alphabetic 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.

sort order

Sort order is an indicator of the direction in which you want your data to be presented once it is 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 the program.

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 can not 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.

subreport

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 "B" is a substring of the string "President Bill Clinton".

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).

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.

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.

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.

suppress

If you suppress something it does not appear. In Crystal Reports you can suppress report sections (headers, details, and footers), rows and columns that don't contain a value, as well as specific values.

syntax

Syntax is a set of rules used to make a correct formula. When creating formulas, you have the option of using Crystal or Basic syntax.

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 a template is used, your original report remains unchanged.

text object

A text object is a specialized 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.

toolbar

A bar at the top of Crystal Reports application window which contains a number of buttons that you can click to activate the most frequently used commands.

total

A total is a sum of values. See also grand total, running total, and subtotal.

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, etc.) 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. In a field called {customer.CONTACT FIRST NAME}, for example, John or Mary might be the value. In 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 the program. For a more thorough discussion on the use of variance, consult any reliable statistics text.

verify

In Crystal Reports 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.

Index