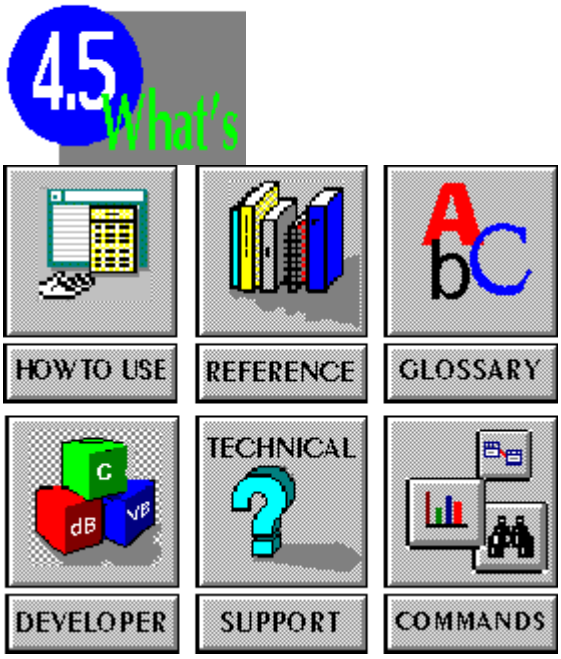
Seagate Software Crystal Reports for Visual Basic Help



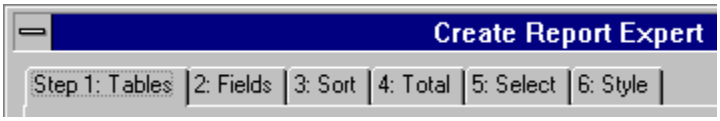
Seagate Software Information Management Group

Crystal Reports Experts

Crystal Reports includes a number of Experts that help you:

- create several kinds of reports,
- prepare graphs,
- set up your overall report format,
- link related databases,
- select records and groups for inclusion in your report, and
- gather the files you need to include when you distribute your report.

In most cases Experts have a series of tabs numbered Step 1, Step 2, etc., like in the picture below:



To use this kind of expert, simply begin at step one and proceed to the last step. When you have completed the last step, the Crystal Reports Expert will do the rest of the work. Its that easy!

Some of the other experts don't use the tab system. In these cases, simply follow the instructions in the dialog boxes that appear and the Expert will lead you through otherwise complex procedures with a minimum of effort.

How to use Crystal Reports

Click on a topic below for more information:

Help topic index (alpha)

Creating a report

Creating specific reports

Working with formulas

Working with Pictures

How to use ODBC (Open Database Connectivity)

SQL Features (topical index)

SQL Features (topical index)

Click on a topic below for more information:

Setting SQL Defaults

SQL Options

Log Off Server command (Database menu)

Log On Server command (Database menu)

Show SQL Query command (Database menu)

Logging onto a SQL server

See Also

Crystal Reports Features List

ODBC – Open Database Connectivity

Click on a topic below for more information:

ODBC – An Overview

Crystal Reports and ODBC

Logging on to a SQL Server or other ODBC data source

Using Crystal Reports with Microsoft Access

Reference

Click on a topic below for more information:

Report Editing Tools

Menu commands

Operator index (alpha)

Operator index (by type)

Function index (alpha)

Function index (by type)

Error messages and formula compiler warnings

Runtime File Requirements

Foreign Language Runtime File Requirements

Technical Support

Click on a topic below for more information:

BEFORE CALLING TECHNICAL SUPPORT....

Areas To Seek Assistance

Register/Change Address command (Help menu)

Technical Support Request dialog box

Other forms of technical support

Where to install CRPE files, Report files, and Database files

CRPE files

CRPE.DLL, C01C40EN.DLL, PG.DLL, and CRRUN.EXE should be in your Windows\System directory. Database DLLs and Export DLLs should be in your Windows\Crystal directory.

NOTE: 32-bit users should note that the 32

-bit version of CRPE is called CRPE32, and CRRUN is CRRUN32, and that files normally located in the SYSTEM directory will be located in the SYSTEM32 directory.

Report Files

This depends on what you have specified in your application. It is recommended that you use the report name with no path. If you have no path specified in your call, you can install your runtime application in whatever directory you like on the user's machine.

When the call is made, the print engine will look in the current directory for the report file. If CRPE doesn't find the report file in the current directory, it will look for it in the Windows directory, in the Windows System directory, and then in the path.

This means that you can place the report files in the same directory as your application and your application can be installed in any directory you like. As long as the application keeps this directory as the current directory, it will find the report files.

If you have specified a path in your call then CRPE will look in that directory for the report.

Database Files

We recommend that you set the database location to be "Same as Report" in the report file itself. This will tell the CRPE print engine to look in the same directory as the report definition (.rpt) file for the databases. Again, if it does not find the databases in this directory, CRPE will look for the databases in the Windows, then Windows\Crystal directory, in the Windows\System directory, and then in the path.

Before calling Seagate Software technical support, the following should be readily available

1. Exact version number of Crystal Reports. (This can be found by selecting the Help|About Crystal Reports menu option.)
2. The operating system you are using, i.e., Windows 95 or Windows 3.x, and whether you have a 16 or 32–bit version of Crystal Reports
3. Serial number from registration. (This can be found by selecting the Help|About Crystal Reports menu option.)
4. The company name to which the product is registered.
5. The telephone and fax number of the company/person to which the product is registered.
6. Which database type and database version is being used and what type of database access is being used.
7. Crystal Reports open, should the call be referring to Crystal Reports itself and not the runtime environment. Should the call refer to a specific development language environment, please state the exact computer language and language version to which the support call pertains.
8. Know exactly what you are trying to accomplish and explain such to the technical support person.

Areas to seek assistance before calling technical support

1. Look under the Help menu for documentation on topics and calling conventions within the development environment.
2. Look at the **readme.hlp** file for further very important documentation.
3. Dial into the Faxback system for an index on technical documentation.

NOTE: There are many helpful hints and technical documents available. These documents could hold the solution to many difficulties experienced.

Other Forms Of Technical Support

CompuServe

You can receive support on Crystal Reports via CompuServe. Once in CompuServe, issue the Go Reports command and you'll enter a forum for Crystal Reports users that is monitored by Crystal staff.

Telephone Support

Registered users of Crystal Reports are entitled to free telephone support (subject to availability) for 60 days from the time of their purchase. Crystal telephone support is available from 8 to 5, P.S.T.

Please have available the following information:

- Product name and version number
- Registration number
- Technical Support Request form from product
- Contents of autoexec.bat and config.sys files
- List of steps necessary to recreate the problem

Telephone Number (604) 669-8379

Mail-in Support - FREE

If you would prefer, you can also contact us by mail with written questions or comments, and we will respond by return mail.

If you believe that your questions are such that we should review all of your related files (such as databases, etc.) to help solve your problem, then copy all of the related files and the crystal Reports Report file (FILENAME.RPT) onto a diskette of any PC compatible format and send it to us. We will investigate the problem and mail back a response to you as soon as possible with your original diskette.

Our mailing address is:

Seagate Software
1095 West Pender Street
4th Floor
Vancouver, BC, Canada V6E 2M6

Online Bulletin Board Support - FREE!

Crystal has been supporting Crystal Reports and its predecessor products for many years. In the process, it has developed a number of support resources that are available on a free bulletin board system (BBS).

There is no assigned staff monitoring the BBS. The BBS is designed for users to converse with one-another, and to upload and download information. If you are sending information to this area for technical support, you should make sure that the technical support representative is aware of this action.

The BBS offers solid support of Crystal Reports for Windows. In addition, the BBS provides easy access to other related information and a direct line to Crystal for various requests you might have.

The power of BBS support is in its ability to accept, via the phone lines, the very files you are seeking support for. You can upload your report definition files (*.RPT) onto the BBS along with the description of your problems. One of our Customer Support Reps will address the problem and send back the solution through the BBS system. your message will be dealt with as quickly as possible. All you need is a modem to get started with this valuable service.

Crystal Bulletin Board (604)681-9516

Facts On Demand

The Facts On Demand service offers listings of technical support documents that are available.

Facts On Demand (604)681–3450

– ALPHABETICAL INDEX TO INSTRUCTION TOPICS

The following major topics are listed in alphabetical order. Choose a letter from the button bar below to jump to a spot in the index, and then choose the topic of interest from the list.

A

Adjusting the number of Browse values

Adding, copying, deleting, and editing text

Auto Arrange feature

B

Bit–mapped pictures concepts

Button Bar

C

Changing default settings

Changing fonts

Changing the format for an entire report section

Changing the format for individual fields

Charting/Graphing with Crystal Reports

Concatenated text strings

Configuration Options

Copying text

Creating A to B, A to C reports

Creating a report

Creating formulas

Creating labels with borders

Creating mailing labels and label–type items

Creating multi–column "telephone book" reports

Creating reports that identify "top" groups

Creating reports using outer joins

Creating reports using self–joins

Creating round labels

Creating subtotals within group reports

Creating summary reports

Creating tables

CrossTab reports

Crystal Reports data types
The Crystal Reports Window
Cursors

D

Data Dictionaries
Databases that work with Crystal Reports
Dates stored in text or number fields
Deleting
- Blank lines
- Text
- Report sections

Design Window
Drilling Down on your summary reports

E

Editing
- Text
- Formulas

Editing, deleting, moving, and sorting data
Experts Overview

F

Format Bar
Formatting report data
Formula Editor
Formulas – an overview
Formulas in action

G

Grouping and Sorting Topics

H

How data types are sorted
How to identify the "Top" groups

I

If–then
–else formulas
Including comments with formulas
Index to formula topics
Inserting fields

L

Logging on to a SQL server

M

Memo fields, using with Crystal Reports
Menu Bar
Moving
    Text
    Fields
Multi –condition if
–then
–else formulas
Multiple field sorts

O

ODBC (Open Database Connectivity)
Order of precedence
Outer joins

P

Paradox secondary indexes
Printing your report
Preview Window

R

Report Style Expert
Report Templates
Right mouse–button capabilities

S

Secondary indexes, Paradox
Selecting a database
Selecting additional databases
Selecting an alias
Self–joins
Setting SQL defaults
Sorting by Record or Group
SQL features (topical index)
Status Bar
Stored Procedures
Support Questions and Answers

T

Technical Support Topics
Templates, Report
The Button Bar
The Crystal Reports Window
The Cursors
The Design Window
The Format Bar
The Formula Editor
The Menu Bar
The Preview Window
The Status Bar

U

Understanding the invisible grid
Using memo fields with Crystal Reports
Using variables in formulas

V

Variables, using in formulas
Visual Linking Expert

– Creating A Report

Crystal Reports is quick and easy to learn. It requires very little time before you will be designing interesting and informative reports for your needs. The following is a list of topics covering step–by–step instructions for creating any report in Crystal Reports.

Click on a topic below for more information:

Selecting a database

Selecting an alias

Selecting additional databases

Understanding the grid

Inserting fields

Deleting blank lines

Grouping data

Identifying the "top" groups

Creating formulas

- Formulas – an overview
- Index of formula topics

Manipulating text and data

Formatting the report

Printing your report

Report Templates

Data Dictionaries

Creating a CrossTab report

Crystal Reports Report Editing Tools

Click on a topic below for more information:

Button Bar

The Status Bar

The Menu Bar

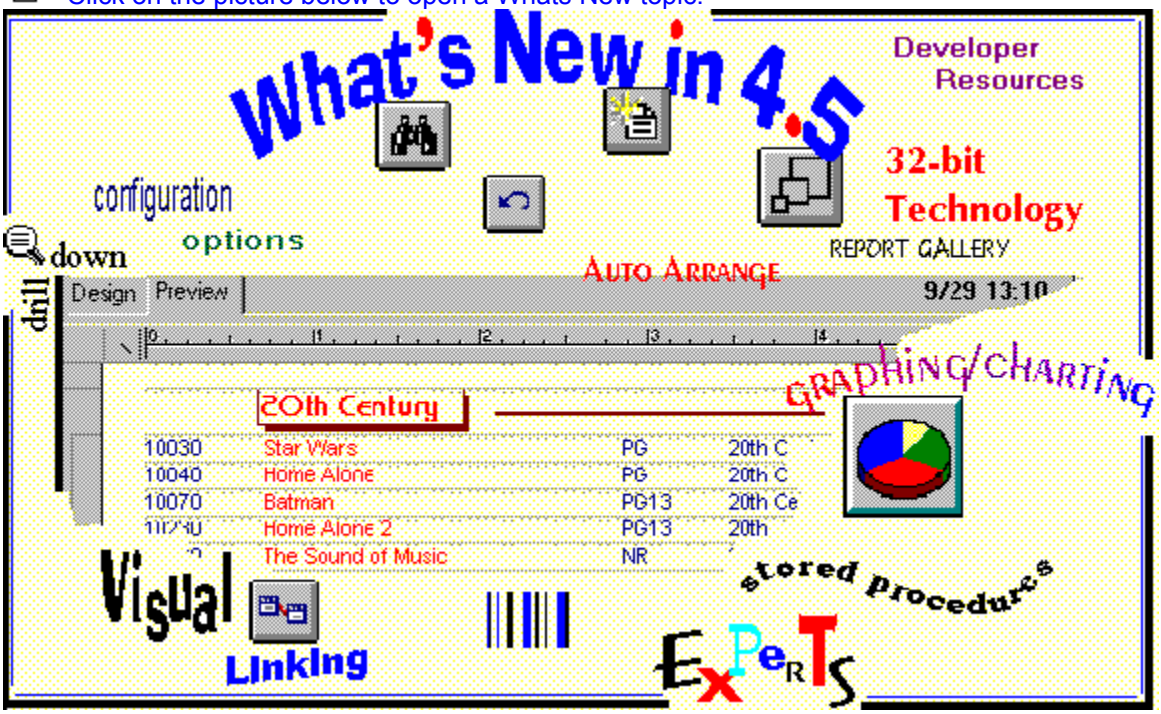
The Format Bar

The Cursors

The Ruler

What's New?

Click on the picture below to open a Whats New topic.



See Also

The What's New List

A Quick Look At New Features In Crystal Reports...

Click on a topic below for more information:

New Developer Resources

Improved MS Access Support

Creating Custom Groups

Configuration Options

Undo and Redo

Visual Linking Expert

New Sorting and Grouping Topics

Drilling Down on your summary reports

Auto Arrange feature

Stored Procedures

Report Creation Experts

Search Feature

Top N Expert

Preview Sample

Report Style Expert

Stored Procedures supported

Faster Report Processing

Advanced Graphing

Report Options saved with Report

Reporting with Lotus Notes

Export in Excel 5.0

Export in Excel 5.0

In Crystal Report, you can now export information directly to Excel 5.0 format.

Reporting with Lotus Notes

Crystal Reports now gives you the ability to create reports importing Lotus Notes (via ODBC) and the ability to export to the Lotus Notes format. (16 bit versions only)

Report Options saved with Report

The new File|Report options allows saving of choices with the report. This can make it easier to deploy reports across machines and throughout an organization.

Advanced Graphing

Take advantage of 12 basic styles, or totally advanced graphing with **over 80 styles and editing control** in the 16 bit version. Control every aspect of graphs in a report. It's a snap to rotate a graph, changing its perspective, or adjust font size or colors. Graphs are easy to insert, and get updated whenever data is refreshed. Graphing options can be controlled at runtime, and you can now drill down directly on graphs to view detailed data.

NOTE: Graph work created in 16 bit can be read and printed in 32 bit.

32–bit Technology and Crystal Reports

You can now run a 32–bit version of Crystal Reports under Windows NT, and Windows 95. There is full compatibility between the 16–bit and 32–bit versions of Crystal Reports, so your applications will be truly cross platform.

New Developer Resources

Report Engine DLL in 32 bit

Now access the full power of Crystal Reports in 32 or 16−bit technology so you can develop and report in any Windows environment. There is full compatibility between both technologies, allowing you to create applications that are truly cross

−platform.

Full Featured OLE Control

Crystal now has a new OCX with over 80 properties for full access to the Crystal Report Engine DLL. The new tabbed interface makes it really easy to navigate around the properties page. You now have numerous report integration options including the OCX and Crystal ActiveX control. Development has never been easier.

Reporting from the NT event log

Use Crystal Reports to report on the Windows NT event log. Monitor and analyze workstation or system related activity. An IS favorite!

Improved MS Access Support

Crystal now offers better reporting with support for Access OLE objects. Report on the information you want, including Access Pictures fields and OLE fields. Save time and effort by using Access queries to generate reports.

Faster Report Processing

New Smart Engine Technology has added speed to Crystal Reports. Now, access data and process reports 2 to 10 times faster than before. Crystal Reports saves time and lets you concentrate on report design.

Crystal Reports Experts

Crystal Reports Offers you several Experts. Experts are tools that take you step by step through various aspects of report Creation. In most cases Experts have a series of numbered tabs. Simply begin at step one and proceed to the last step. When you have completed the last step, the Crystal Reports Expert will do the rest of the work. Its that easy!

Click on the picture below for more information:

See Also

Experts Overview topic.

- # The Undo and Redo Features

Undo command

The Undo command is a dynamic command. The text changes on the menu based on the last action you performed in Crystal Reports. The command supports multiple levels of Undo.

Undo is also available as a button on the button bar.

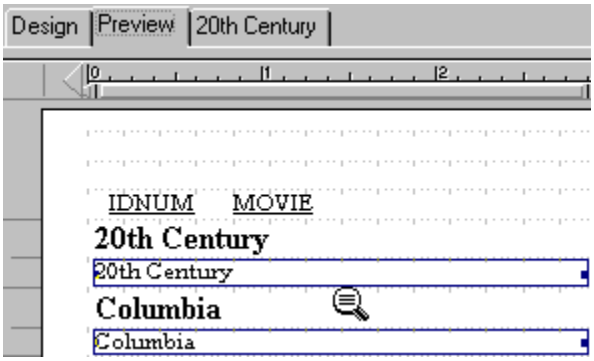
Redo command

The Redo command is a dynamic command. The text changes on the menu based on the last Undo you performed in Crystal Reports. The command supports multiple levels of Redo.

Drilling Down on your summary reports

The program now has data analysis tools allowing you to drill down to find the details behind summarized data.

To drill down on a summary report



1. Open your report. The report must contain at least one summary field.
2. In the Design Window, make sure the Details section is hidden.
3. Print the report to the Preview Windowfinetunr. A summary report will appear showing summarized values for each group and a single Report tab appears at the top of the window. Since the details are hidden, a drill-down cursor replaces the normal arrow cursor. You use this new cursor for drilling down to get the details behind any group summary.
4. To drill down on a particular group, double-click any of the fields in the group of interest using the drill down cursor. The program displays the details behind the group summary (a group details report) and it creates a new tab at the top of the window for the group you're working with.
5. To drill down on another group, Click the Report tab to return to the Summary report and repeat Step 4 for the new group.

NOTE: Each time you drill down on a group, the program creates a new tab for that group. You can move to any detail report or return to the summary report simply by clicking the tab for the report of interest.

If the program creates too many tabs to fit in the Preview window, it activates the two scroll buttons in the top right-hand corner of the window. You can use these buttons to scroll through the tabs to find the report that you want.

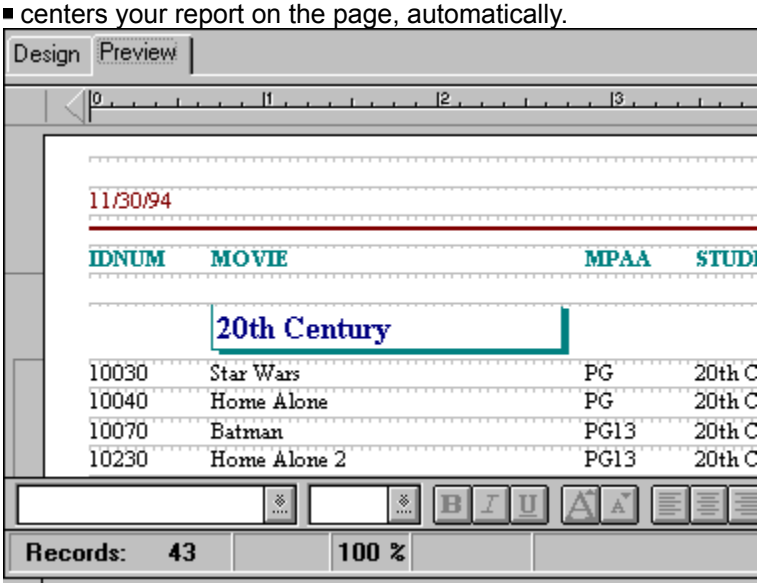
NOTE: When you close the summary report, your Drill Down tabs are not saved.

- ## Auto Arrange

When you place a field on your report, Crystal Reports allocates a space equal to the field width as specified in your database. Often that field width is far larger than the values that actually appear in each of the fields. For neat looking reports, it is often necessary to resize the fields so the space allotted more closely matches the size of the field values. Once you've resized the fields you often need to reposition them for proper balance.

To save you time, Crystal Reports now includes an "Auto Arrange" capability that

- adjusts the spacing between fields,
- repositions the fields, and then
- centers your report on the page, automatically.



The Ruler

The program now makes a ruler and a positioning wand available in both the Design Window and the Preview Window.

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.

You can perform the following actions using the ruler and positioning wand:

Change page margins

Resize objects

Move Objects

To Change Page Margins using the ruler

To reset the left page margin using the ruler

To reset the left page margin, position your cursor at the left end of the ruler, in the gap between the ruler and the arrowhead. The cursor changes to a resizing cursor.

Drag the cursor to the right or left to the new margin position. Negative numbers appear to the left of the zero mark to show you how far the left margin is from the left edge of the paper. Use the negative numbers and/or the Status bar as guides for setting the left margin. When you release the mouse button, the negative numbers disappear and the left end of the ruler (the zero mark) defines your new left margin.

To reset the right page margin using the ruler

To reset the right page margin, position your cursor at the right end of the ruler, in the gap between the ruler and the arrowhead. The cursor changes to a resizing cursor.

Drag the cursor to the right or to the left to the new margin position. As you drag the cursor, a vertical line appears to define the right margin and the ruler numbers remain visible on screen. Use those numbers and/or the Status bar to position the right margin where you want it. When you release the mouse button, the ruler resizes so the right end of the ruler defines the right page margin.

NOTE: You can also set the margins by using the File|Page Margins command .

To Resize Objects using the positioning wand

1. Click on the object you would like to resize. A wand with arrows on either side appears directly below the ruler.
2. Place your cursor on the line between the left arrow and the positioning wand (to resize to the left) or between the positioning wand and the right arrow (to resize to the right). The arrow cursor turns into a resizing cursor when the mouse is in the correct position.
3. Drag the cursor to the right or left until the object is the desired size. Use the increment marks on the ruler and/or the Status Bar coordinates at the bottom of the screen as guides for resizing the object.

To Move An Object Using the positioning wand

1. Click on the object(s) you would like to move. A wand with arrows on either side appears directly below the ruler.
2. Move the cursor directly over the wand until it becomes a tiny hand cursor.
3. Drag the positioning wand to the desired position. Use the increment marks on the ruler and/or the Status Bar coordinates at the bottom of the screen as guides for making the exact placement.

Formatting report data

Click on a topic below to format fonts and styles for fields and text.

Changing the format for an entire report section

Changing the format for individual fields

Changing fonts

Manipulating text and data

How to edit, delete, sort, and change text and fields.

Copying

- Text

Editing

- Text
- Formulas

Deleting

- Text
- Report sections

Moving

- Text
- Fields

Sorting data by Group or Record

- ## Creating specific kinds of reports

Click on a topic below for more information:

Creating CrossTab reports

Creating summary reports

Creating A to B, A to C reports

Creating tables

Creating mailing labels and label-type items

Creating labels with borders

Creating round labels

Creating multi-column "telephone book" reports

Creating reports that identify "top" groups

Creating subtotals within group reports

Creating reports using self-joins

Creating reports using outer joins

Working with pictures

Click on a topic below for more information:

Bitmapped pictures concepts

Creating tables using lines and boxes

Glossary

A

Access

Active database

Alias

Argument

Array

Auto Arrange

B

BLOB Fields

Boilerplate text

Boolean formulas

Boolean expression

C

Calculated data field

Case sensitive

Character attributes

Column

Concatenate

Condition

Consequence

Constant

Cursor keys

Cut

D

Data field

Database

Data types

Default

Definition

Details section

Divide by zero protection

Drag

Dynamic Link Library (DLL)

E

Element (report)
Empty date
Empty string
Expert

F

Field
Features List
Field placement box
Flag
Footer
Form letter
Formula
Function

G

Grand total
Grid
Group
Group value

H

Header
Hourglass cursor

I

I-beam cursor
Icon
Index
Insertion point
Integer

L

Link

M

Maximize button
Menu
Minimize button

N

Nesting

Null string
Numeric

O

ODBC
Operators

P

Paste
Point
Population
Population standard deviation
Population variance
Precedence, order of precedence

R

Range
Record
Report
Report definition
Returns
Row
Ruler

S

Sample
Select
Selection formula
Sort and group by field
Sort field
Sort order
Sorting
S.Q.L
Standard deviation
String
Substring
Subtotal
Summary field
Summary section
Syntax

T

Tabs

Text string

Title bar

Title section

Total

Truncate

Two pass formula/function

V

Value

Variance

access data

To access data means **to retrieve data**.

alias

In Crystal Reports, an alias is **an alternative name assigned to a database**. If a database is called customer.db, you could assign the alias *customer*, *cust*, *company*, *DB1*, or any other name that suits your needs. Aliases make it easier for you to use a report created with a database whose name and/or location has changed since the report was created.

active database

An active database is **a database that has been selected for use in a report**. You activate databases via the File|New and Database|Add Database to Report commands.

argument

An argument is an **item**, or one of a group of items, **that receives the action of a function**. It provides information that the function needs in order to operate. The Truncate function, for example, cannot operate by itself. It needs an argument that identifies the *thing* to be truncated. Thus, in the formula Truncate (*{file.Amount}*) where Truncate is the function and *{file.Amount}* is the argument, it is the value of the *{file.Amount}* field that is the item to be truncated.

array

An array is **a group of values, separated by commas.** Arrays are used with a variety of Crystal Reports functions: Average([array]), Maximum([array]), etc. In these functions, the array is the argument for the function. The function works on the items in the array. Items in an array can be constants, data fields, or formula results.

boilerplate text

Boilerplate text is **text that can be created once, then used again and again**. Using a Crystal Reports' if-then-else formula, you can automatically insert different blocks of boilerplate text in form letters to customers, sales reps, lenders, etc.

Boolean formulas

Boolean formulas are **formulas that return a Yes/No (True/False) value**. For example, the Boolean formula *{file.Qty}* > 6 compares the value in the *{file.Qty}* 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 *{file.Qty}* * 6. In this case Crystal Reports returns a number, the value of *{file.Qty}* times 6.

NOTE: All record and group selection formulas must be Boolean.

calculated data field

A calculated data field is **a field that holds a value that comes from a calculation instead of coming directly from a database**. For example, if the database you are using includes a *{file.Sales}* field and a *{file.Cost}* field but no *Gross Profit* field, you can still show gross profit on your report, if you wish, using a calculated data field. To create a calculated data field, you simply create a formula that subtracts *{file.Cost}* from *{file.Sales}*. The formula calculates a *Gross Profit* value for each row and prints it wherever you place the formula.

case sensitive

Case sensitive means that a program **differentiates between uppercase and lowercase letters** when evaluating a text string. A case sensitive search for the word "house" will return only the value "house," but a non-case sensitive search will return "house," "House," "HOUSE," "HoUsE," and similar mixed-case responses. Crystal Reports operators (Equal, In string, etc.) are case sensitive.

character attributes

A character attribute is something that **defines the look of a character** when it is displayed or printed. Crystal Reports allows you to change such attributes as font and point size, and it allows you to make characters bold or italic, to underline them if you wish, and to make other related changes. You can change the character attributes of selected parts of your report (headings, totals, etc.) to emphasize key information.

column

In Crystal Reports, a column is the display of data **from a single field or formula**. Columns run up and down the page. The words column and field are sometimes used interchangeably in this manual. Contrast with Row.

concatenate

Concatenate means **to join two or more text strings together** to form a single contiguous string.

condition

In an if▪then

▪else formula, the condition is **the *if* part of the formula**, the set of circumstances that must take place (be true) to trigger the *then* (or consequence) part of the formula. In the formula *if x<5 then x else 5*, the expression *x<5* is the condition.

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 value that is fixed and unchanging as opposed to a variable value which can take on different values depending on the circumstances.

The value 5 is a constant; the value of the Quantity field (which sometimes may be 5, sometimes may be a different number) is a variable value. In the formula for converting pounds to ounces (Ounces = Pounds * 16) for example, 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, c**onstants can be numbers, text strings, dates, dollar amounts, or the result of a formula that itself contains no variables**, e.g., 14

▪9. Contrast with Variable.

cursor keys

Cursor keys are the keys on your numeric keypad that can be used to control cursor or highlight movement. These include **Home, End, PgUp, PgDn, and the arrow keys**.

cut

In Crystal Reports, Cut means t**o remove data from a report or formula and move it to the Windows clipboard**. Once in the clipboard it can be pasted (retrieved and placed) somewhere else in the same (or different) report or formula. Contrast with Paste.

data field

A data field (or field) is **the basic building block of a record**. Each record is made up of one or more data fields, and each data field can hold one piece of data (known as a value). A customer record in a typical customer mailing list database might contain data fields similar to these: Name, Address, City, State, Zip, Phone, FAX. A data field can be empty or contain a value. Data field data is generally displayed or printed in columns in the Details section of a Crystal Reports report. See also Database, Record, Value, Row, Column, Details section

data types

A data type is **a classification of the data that appears in a field or formula**. Each piece of data used in a Crystal Reports report or formula has one of the following data types: text, dollar amount, number, date, or Boolean (YES/NO). It is important to understand data types because each function and operator works with only a limited number of data types (often as few as one). For some operators (+ and ■ for example), Crystal Reports uses a different set of calculation rules for one type of data than it uses for another.

database

A database is **a bank of related data**. Each unit (record) of the database is typically organized in a fixed format to make it easier to retrieve selected portions of the data on demand. See also Record, Field, Value

default

A default is a **pre-loaded response to a software request for data**. It is the response the computer accepts automatically if you don't enter different data.

definition, report definition

A report definition is **an overview or *thumbnail sketch* of a given report.** It lists, among other things, the name of the report, the active databases, and any record or group sort fields. For each formula it also lists the formula name and the formula itself.

details section

The details section of a Crystal Reports report is **the core section of the report**. You structure the report in this section by inserting data fields, formulas, and other report elements.

divide by zero protection.

PC's will not allow you to divide a number by zero. If you attempt such a division, you will get a system error message. To protect you from a system error, **Crystal Reports refuses to print a report which contains a formula that divides a value by zero**.

drag

Drag means different things, depending on the context in which the word is used:

- When referring to *moving a field* , drag means to **Click on the field box and, while keeping the left mouse button depressed, to move it to a new position using the mouse**. You release the mouse button when the field is in the position you want it.
- When referring to *resizing a field*, drag means to **Click on one of the field box handles and, while keeping the left mouse button depressed, to make the field bigger or smaller using the mouse.** You release the mouse button when the field is the size you want it.
- When referring to formatting text, drag means **to highlight the text of interest by moving the I beam cursor across it while the left mouse button is depressed**. You release the mouse button when you have finished highlighting.

Dynamic Link Library

A Dynamic Link Library (DLL) is **a special kind of file that contains Windows functions**. DLL's are **used by developers to extend the capabilities of Windows applications.** The library is activated whenever a program or another DLL calls a function in the library. *DLL's 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 DLL's; they don't need to be linked to the program via the compiler. The Crystal Reports print engine can be called as a DLL by developers for use with applications they are developing.

element

The word element is used at times to describe **individual report components such as database fields, formulas, group fields, and text**. The Design Window uses rectangular boxes to represent fields; text entered directly into the Design Window appears as text in the Designer.

empty string

An empty string (designated as **""**) is **a string that contains no characters**. In Crystal Reports, you use an empty string to specify that nothing be printed. For example, in the formula:

```
If Gradepoint = 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.

empty date

An empty date [designated as Date(0,0,0)] is **a date that contains no month, day, or year, and thus does not print**. In Crystal Reports you use an empty date in if-then-else formulas that either return a date or do not. For example, the formula:

```
If PageNumber = 1 Then
PrintDate
Else
Date(0,0,0)
```

prints the print date on the first page and prints nothing on every other page. Since the *then* part of the formula is a date (PrintDate), the *else* part of the formula must be a date as well, but a non-printing date. To create such a non-printing (empty) date you use the Date function and the arguments (0,0,0).

field

A field is **the basic building block of a record**. Each record is made up of one or more fields, and each field can hold one piece of data (known as a value). A customer record in a typical customer mailing list database might contain fields similar to these: Name, Address, City, State, Zip, Phone, FAX. A field can be empty or contain a value. Field data is generally displayed or printed in columns in the Details section of a Crystal Reports report. See also Database, Record, Value, Row, Column, Details section

flag

A flag is **a character or group of characters used to highlight or identify items of interest to call them to the reader's attention**. For example, in an accounts receivable report, the words "past due" might be printed as a flag beside every past due account.

footer

A footer is **a small amount of text that appears at the bottom of a report page**. Footer text often includes page numbers and sometimes other information that describes or identifies the report. Crystal Reports gives you the option of printing the footer on all pages or only on selected pages of your report.

form letter

In Crystal Reports, a form letter is **a letter that can be reproduced, personalized, and customized using Crystal Reports' powerful formula capabilities**. One Crystal Reports formula could be used, for example, to print the current date on the letter. Another formula could print an inside address, pulling the data from an address database. Another formula could insert the proper name in the salutation. Yet another formula could insert an appropriate paragraph of boilerplate text, based on the evaluation of specific field values or the results of certain calculations or comparisons. Form letters, customized in this way, can be very effective tools for marketing, collections, fund raising, etc.

formula

A formula is **a symbolic statement of the manipulations you want performed on certain data before it is printed on your report.** If your report is to contain a *{file.Sales}* field and a *{file.Cost}* field, for example, you may want to create a *GrossProfit* field and designate its value as *{file.Sales}* ■ *{file.Cost}*. *{file.Sales}*

■ *{file.Cost}* is a simple formula that tells Crystal Reports to subtract the value of the *{file.Cost}* field from the value of the *{file.Sales}* field and then to print the result. You can use formulas to calculate numeric values, compare one value to another and select alternative actions based on the comparison, join multiple text strings into a single string, and for a multitude of other purposes. Creating a formula in Crystal Reports is much like creating one in your favorite spreadsheet.

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 operations built into the function without you having to specify each operation separately. In this way, a function is a kind of shorthand that makes it easier and less time consuming for you to create reports. 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 total of all values in a column for the entire report**.

group

A group is **a set of records that are related to each other in some way**. In a customer list, for example, a group could consist of all those customers living in the same ZIP code, or in the same state. In a sales report, a group could consist of all the orders placed by the same customer, or all of the orders generated by a specific sales rep. Crystal Reports offers you a great deal of flexibility in the way you group the data on your report.

sort and group by field

A sort and group by field is **a field that triggers the printing of a subtotal (or a group field value) whenever its own value changes**. On a customer order report, for example, two fields are *{file.Customer}* and *{file.Amount}*. If you want to subtotal by customer (total the orders for each customer), you select the *{file.Amount}* field as the field to subtotal and the *{file.Customer}* 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 *{file.Customer}* field changes (when it changes from one customer to a different customer), Crystal Reports prints a subtotal of the values in the *{file.Amount}* field (a total of orders for the individual customer.) You also select sort and group by fields to trigger summaries.

group value

A group value is **the value generated as the result of an evaluation, a tally, or a calculation performed on data from a single group**. A subtotal is one kind of group value; it is the sum of all of the 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 rep, Crystal Reports gathers all the records that belong to the sales rep and totals the amounts ordered from all the records. In a group average, Crystal Reports averages the values in a group of records; it a group count, it counts the values in a group of records, etc. Group values are important tools for creating powerful reports.

header

A header is **a small amount of text that appears at the top of a report page**, above the body of the report. While a header can contain virtually any information, it often contains such things as the report title, company name, date, range of dates covered by the report, etc. Crystal Reports gives you the option of printing the header on all pages or only on selected pages of your report.

hourglass cursor

The Hourglass pointer is **the Windows cursor that appears whenever Crystal Reports, or another Windows application, is processing a command you selected**. Whenever the cursor is visible, you cannot select any other commands or proceed further with your report.

I-beam cursor

The I-beam cursor is the cursor you will find yourself working with most often in Crystal Reports. Shaped like a stylized letter I, this cursor is active whenever you are working in either the Design Window or the Formula Editor. The I

-beam cursor is **the cursor you use to select free form text, and to set the position of the text cursor.**

icon

In Windows 3.1 or NT, an Icon is a symbol that is used to represent a program or a file, or to appear in place of a minimized window. Crystal Reports displays an application icon in the Program Manager; you double click on the icon to start Crystal Reports. When you double click on the icon for a minimized Window, the Window is restored to its previous size.

In Windows 95, icons appear in two sizes. Large icons appear in folders as applications or on the desktop as shortcuts. Smaller icons appear in the Windows Explorer with files, in the Taskbar folders representing applications, or on the taskbar buttons representing running applications. Crystal reports is initially installed in a folder that will appear on the desktop. Once this has been closed, the Crystal Reports folder will appear only in the programs folder from the Start option on the Taskbar.

index

An index is **a small file that identifies the location of each record in a database**. Since a tiny index file can be searched or sorted much quicker than a large database, Crystal Reports uses index files to speed up the report generation process. In a search, for example, Crystal Reports searches the index for the correct field location. Once found, Crystal Reports goes directly to the database field. Such a search does away with the need for searching every field of every record in a database. A database may have several indexes, each based on a specific field (or fields).

field placement box

A field placement box is **a rectangular cursor that appears as an aid to placing database fields and formulas on your report**. Once you have selected a field or created a formula, the field box appears. When you move the box to the place in the report you want the field or formula to appear and click on the mouse button, Crystal Reports inserts the item at the point specified.

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 clicking on the left mouse button. When typing text for the first time in a Design Window section, Crystal Reports sets the insertion point flush left in the section, regardless of where you Click the I

-beam cursor.

integer

An integer is a **positive or negative whole number or zero**. Integers have no decimal places. Crystal Reports' Truncate function cuts the decimal places off a value, thus converting the value into an integer.

link

A link is **a field that is common to two or more databases and that serves as a connecting point between those databases.** Crystal Reports uses the link to match up records from one database with those from the other(s). For example, if the databases each contain a customer number field (even though the fields might have different names), Crystal Reports can use those fields to electronically connect all records in one database with corresponding records in the other(s). When you create a single report based on multiple databases, the link assures that all the data in each row on that report refers to the same customer (transaction, invoice, etc.).

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 five and the value of B must be less than 10. If the values don't fall into those ranges, then the expression is FALSE. Boolean expressions are useful in if■ then

■else formulas. For example, If A>5 and B<10 then "In Range" else "" is a formula that says, if the Boolean expression A>5 and B<10 is TRUE, print "In Range." Else (if the Boolean expression is FALSE) print nothing (as designated by the empty string "" .)

maximize button

In Windows, the maximize button **expands the active window** so it fills the whole screen.

menu

A menu is a **list of choices** that appears on screen. With menus, you don't have to remember cryptic commands or arcane keystroke combinations; you just call up a menu and select the item of interest. Crystal Reports provides you with several menus that you can use to create your reports quickly and easily.

minimize button

In Windows, the minimize button **reduces the active window** to an icon that can later be activated and recalled. You can use the minimize button to reduce the clutter on your Windows desktop while still keeping all of your needed files close at hand.

nesting

In Crystal Reports, nesting means to **use one if-then-else expression inside another.** For example, If employees degree isn't 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. [Note: this is not an actual formula but a verbal explanation of a formula.] 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 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. See also, Empty string.

numeric

Numeric data is **data on which you can perform arithmetic**. The designation numeric refers to the way the data is treated by Crystal Reports and database programs, not to the way the data looks to you. For example, a serial number 12345 looks numeric, that is, every character is a number. But a serial number is not the kind of data on which you would want to perform arithmetic so you would probably store a serial number as text instead of as numeric data. Numeric is one of several data types. Database programs require you to designate a data type when you create a field for use in a database. The data type you select determines the rules the program follows when dealing with the values stored in that field.

operators

Operators are **special symbols that describe 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.

paste

In Crystal Reports, Paste means **to retrieve and place data from the Windows clipboard into a report or formula**. The data may have been Cut (moved to the Clipboard) from the same report or formula or from a different one. Contrast with Cut.

point

A point is **unit of measure used to define type size**. The higher the point value, the larger the type. In a book, for example, the main body copy may be 10 or 12 point type while the headings may be 15 or 18 points.

population

A population is the **entire** set of values that might be tested statistically as opposed to a sample which is a subset of the population. A population does not necessarily refer to a group of people; it can refer to the number of automobiles produced on an assembly line or the number of construction companies bidding on a project.

For example, a Real Estate Agent might sell twenty houses in one year. The population of houses sold by that Agent in that year is twenty. Compare with **sample**.

population standard deviation

Population standard deviation is a statistical test of **how** the values in an entire population (all values) deviate from the mean or average value for that population. Population standard deviation is most often used when all values are being evaluated as opposed to just a sample of those values (StdDev).

NOTE: This comparison simply suggests typical usage. In practice, some users prefer a calculation based on N values (PopulationStdDev) while others prefer a calculation based on N■1 values (StdDev) . Both forms of standard deviation are provided by Crystal Reports.

Crystal Reports calculates the population standard deviation in the following manner:

1. It calculates the average (mean) value for the items being analyzed.
2. It subtracts the average value from the value of each item.
3. It squares the difference for each item.
4. It adds the squared differences for all the items being analyzed.
5. It divides the sum by the number of items being analyzed (N). (Contrast with StdDev which divides by N■1.) The result is the Population Variance.
6. It calculates the square root of the Population Variance to arrive at the Population Standard Deviation.

NOTE: Also see** standard deviation **for a general discussion on the use of standard deviation.

population variance

Population variance is the square of the population standard deviation . It is a measure of **the amount** by which the values in an entire population vary from the mean (average) value for that population.

Population variance is typically used when all values are being evaluated as opposed to just a sample of those values (Variance).

NOTE: This comparison simply suggests typical usage. In practice, some users prefer a calculation based on N values (PopulationVariance) while others prefer a calculation based on N■1 values (Variance) . Both forms of variance are provided by Crystal Reports.

Crystal Reports calculates the population variance in the following way:

1. It calculates the average (mean) value for the items being analyzed.
2. It subtracts the average value from the value of each item.
3. It squares the difference for each item.
4. It adds the squared differences for all the items being analyzed.
5. It divides the sum by the number of items being analyzed (N). (Contrast with Variance which divides by N■1.) The result is the Population Variance.

NOTE: Also see** variance **for a general discussion on the use of variance.

precedence, 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.

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 means much more. It also refers to invoices, form letters, mailing labels, and other related items that require the organization and output of data.

returns

The word returns refers to **the result of a function, an operation, or a formula**.

- When you use a **function, it performs a calculation or manipulation that results in a data change of some kind.** The data that results is what the function returns. For example, Average(*1,2,3,4,5*) *returns* the average of the array 1,2,3,4,5. Truncate(1.2345) *returns* the integer (whole number) portion of the number 1,2,3,4,5.
- When you use an **operator, the result of the operation using that operation is what the operation returns**. For example, 5*6 equals 30. You can say that the operation 5*6 returns 30. Also, the operation 100<200 compares the two values and returns True; 200<100 compares the two values and returns False.
- When you use a formula that contains functions or operators, **each function or operation within the formula returns a result, but the formula taken as a whole returns a result too. When talking about a formula, it is the result of the formula that is of interest, not the result of individual functions or operations**. For example. in the formula *If {file.Qty} < {file.ReorderAmount} then "Reorder "else ""*, an internal operation compares the value of the *{file.Qty}* field with the value of the *{file.ReorderAmount}* field. If *{file.Qty}* is less than *{file.ReorderAmount},* that individual operation returns the value *True.* but that is *not* what the formula taken as a whole returns. The formula, taken as a whole, returns the flag "Reorder" when the operation internally returns the value True.

row

In Crystal Reports, a row is **the display of data from a single record**. Rows run across the page. The words row and record are sometimes used interchangeably in this manual. Contrast with Column.

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. Compare with **population**.

select

- **With regard to *menu options***, select means to point to a menu option using the mouse and then to click on that option, thus making it active.
- **With regard to a *report element* (data field, formula, etc.),** select means to point to the element with the mouse and then to click the left mouse button to choose the element as the object of the next menu selection. For example, to change font size, you first select the block of data for which you want to change font size. Then you select the Font option from the Format menu to select the new font size. The new font size applies only to the element you selected.
- **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. For example, if you want to build a report based only on the records of those customers who have purchased within the last six months, Crystal Reports will build the report with those records and will ignore all the rest.
- **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.**

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 *{file.ZIP}* code field; that is, the customers would be sorted so that those with the lowest zip codes would appear first and those with the highest ZIP codes would appear last. It could also be sorted in ascending alphabetical order, on the *{file.lastname}* field; that is, customers with last names beginning with A would appear first and those with last names beginning with Z would appear last. In these examples, *{file.ZIP}* and *{file.lastname}* are the sort fields.

sort order

Sort order is an indicator of the **direction in which you want your data to be presented**, once sorted. Data is typically printed in one of two sort orders: **ascending** (lowest to highest, earliest to latest, first to last, a to z, etc.) or **descending** (highest to lowest, latest to earliest, last to first, z to a, etc.).

sorting

Sorting is a **method of organizing the order in which data appears** on your report. Crystal Reports provides you with powerful tools for sorting your report data.

standard deviation

Standard deviation is the square root of the **variance**. It is a statistical test of **how** various values in a set of values deviate from the mean or average value for that set. You can use standard deviation, for example, for assessing the relative difficulty of tests given to students, for evaluating and projecting customer purchase patterns, or for comparing the results delivered by two or more products under evaluation (laboratory blood tests, smoke detectors, radar detectors, etc.) The uses are endless.

Standard deviation (as opposed to population standard deviation) is typically used to project the standard deviation for an entire population (all values) based on testing only a small sample of that population. For example, a company producing batteries with a new manufacturing process might want to test the batteries to determine how long they will last before they go dead. If the company tested all of its batteries, it would have no product left to sell. As an alternative, the company might test thirty batteries selected at random and project the mean burn out time and standard deviation for all batteries based on the results from that thirty battery sample.

NOTE: This comparison simply suggests typical usage. In practice, some users prefer a calculation based on N values (PopulationStdDev) while others prefer a calculation based on N■1 values (StdDev) . Both forms of standard deviation are provided by Crystal Reports.

Crystal Reports calculates the standard deviation in the following manner:

1. It calculates the average (mean) value for the items in the sample.
2. It subtracts the average value from the value of each item.
3. It squares the difference for each item.
4. It adds the squared differences for all of the items in the sample.
5. It divides the sum by one less than the number of items in the sample (N■1). Thus, if there are ten items in the sample, it divides the sum by 9. (Contrast with population standard deviation which divides by N.) The result is the Variance.
6. It calculates the square root of the Variance to arrive at the Standard Deviation.

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.

substring

A substring is simply **a part of a larger string**. "Columbia" is a substring of the string "British Columbia," "1040" is a substring of the customer number "B■1040
■0032456," and "G" is a substring of the string "President George Bush."

subtotal

A subtotal is **a partial total, a total of a specific, limited group of data** in a field. For example, given the following data:

```
1, 2, 3, 4, 5, 6, 7
```

a subtotal after the 3 produces the value 6 (1 + 2 + 3). A second subtotal after the 6 produces the value 15 (4 + 5 + 6).

summary field

A summary field is **a field that determines the sum of the values, the average value, the maximum value, the minimum value, or count of values in a group of values in a given field**. Much like a subtotal, a summary field groups data to your specifications and then performs the requested calculation/determination.

syntax

Syntax, in Crystal Reports, is **a set of rules that specifies the proper way to use functions and operators in formulas.**

text string

A text string is **text that is entered directly onto the report itself** instead of being entered via a data field or formula.

title bar

The title bar is the **bar at the top of a window** that indicates the content of the window (application, document, etc.).

total

A total is a **sum of values**. Subtotals, running totals, and grand totals are three different varieties of totals. See also, Subtotal, Running Total, and Grand Total

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 past. An example of a two pass formula is one that calculates the sales for each sales rep as a percent of total company sales. The first pass sums the sales for each rep to arrive at total company sales. The second pass divides the sales per rep by total company sales to calculate the percent of

- total
- sales.

value

A value is **the data found in a field**. For a field called *{file.FirstName}*, for example, John or Mary might be the value. For a field called *{file.Amount}*, 1234.55 or $200 might be the value.

variance

Variance is the square of the **standard deviation**. It is a measure of **the amount** by which all values in a group vary from the mean (average) value in the group. It is a statistical test that can be used to evaluate the variability in a group of values (for example, the amount bid by each of the bidders on a construction project).

Variance (as opposed to PopulationVariance) is most often used to project the variance for an entire population (all values) based on testing only a small sample of that population. For example, with a limited number of bids in on a construction project, you might want to project the variance for all bids based on the sample already in. Or, based on sales figures for the first three months of the year, you might want to project the variance for orders for the entire year (including the nine months yet to come).

NOTE: These comparisons simply suggest typical usage. In practice, some users prefer a calculation based on N values (PopulationVariance) while others prefer a calculation based on N-1 values (Variance) . Both forms of variance are provided by Crystal Reports. For a more thorough discussion on the use of variance, consult any reliable statistics text.

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

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.

BLOB

A Blob field is a field containing Blob data. A Blob (Binary Large Object) is simply a bitmapped graphic that has been entered into a database. Placing a Blob field on your report allows you to access these graphics and manipulate them as you would other data types.

Experts

Crystal Reports Offers you several Experts. Experts are tools that take you step by step through various aspects of report Creation. In most cases Experts have a series of numbered tabs. Simply begin at step one and proceed to the last step. When you have completed the last step, the Crystal Reports Expert will do the rest of the work. Its that easy!

Summary Section

The Summary Section is the last section of your report in the Design Window. You can place a summary in this section that you want to appear **only** on the last page of your report.

Title Section

The Title Section is the first section of your report in the Design Window. You can place a title in this section, or any data you want to appear **only** on the first page of your report.

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 when you click the tab name.

Ruler

The Ruler is visible in both the Design Window and Preview Window when their respective check boxes are toggled on in File|Options

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.

Grid

In Crystal Reports, the grid is an underlying network of lines that are similar to the lines on graph paper. You can use these lines to help align fields and graphics. If you have Snap to Grid selected as one of your default options in File|Opitons, Crystal Reports will automatically align any fields you insert or resize to the nearest grid line.

Auto Arrange

When you place a field on your report, Crystal Reports allocates a space equal to the field width as specified in your database. Often that field width is far larger than the values that actually appear in each of the fields. For neat looking reports, it is often necessary to resize the fields so the space allotted more closely matches the size of the field values. Once you've resized the fields you often need to reposition them for proper balance.

Auto Arrange will do this for you. Simply select Auto Arrange Report from the format menu.

Experts Index ▪

Click on a topic below for more information:

Report Setup Experts

You can use these experts to build specific kinds of reports from scratch.

Standard Expert

Listing Expert

CrossTab Expert

Mailing Label Expert

Summary Expert

Graph Expert

Top N Expert

Drill Down Expert

Procedure Experts

You can use these experts to help you perform a number of sophisticated reporting tasks:

Change Group Expert

Graph/Chart Expert

Report Style Expert

Select Records Expert

Top N/Sort Group Expert

Visual Linking Expert

Crystal Reports Features List

The following table shows what features are available in the various versions of Crystal Reports, and is subject to change.

	Professional (32 bit)	Professional (16 bit)	Standard (32 bit)
New Features			
OLE Control (OCX)	>80	>35	>35
32-bit Report Engine DLL	>80	>80	>35
Drilldown on graphs	Yes	Yes	Yes
Graphing customization	*	Yes	*
Export to Lotus Notes data	No	Yes	No
Reports from the NT Event Log (NT Only)	Yes	No	Yes
Improved MS Access support	Yes	Yes	Yes
Export to MS Excel 5.0 format	Yes	Yes	Yes
Report Options saved with report	Yes	Yes	Yes
Downward Compatibility with reports	Yes	Yes	Yes
PC Data Access			
ASCII	Yes	Yes	Yes
ACT! 2.0	No	Yes	No
Btrieve (all indices, requires DDF's)	No	Yes	No
Clipper (NTX)	Yes	Yes	Yes
dBASE (NDX, MDX, Visual dBASE & QBE)(QBE16-bit only)	Yes	Yes	Yes
FoxPro (IDX, CDX, Visual FoxPro)	Yes	Yes	Yes
Microsoft Access (1.0-2.0) & Access Queries	Yes	Yes	Yes
Microsoft Excel (XLS)	Yes	Yes	Yes
Paradox (3.5-5.5)**	Yes	Yes	Yes
Lotus Notes	No	Yes	No
SQL DATA ACCESS			
Borland Database Engine (IDAPI)	*	Yes	No
ODBC	Yes	Yes	No
Gupta SQLBase	*	Yes	No
IBM DB2/2	No	Yes	No
MS SQL Server	Yes	Yes	No
Oracle	*	Yes	No
Scaleable SQL	*	Yes	No

Sybase SQL Server	*	Yes	No
Stored Procedure support	Yes	Yes	No
Ability to sort data on SQL Server	Yes	Yes	No
Case Sensitive SQL data	Yes	Yes	No
Ability to convert database drivers	Yes	Yes	Yes

NOTE: 32-bit ready. Requires component not currently available from third party vendor. Paradox via ODBC (Only needed for Paradox version 5.0 and later).

Developer Help

NOTE: Window 95 users: The search operation will not search if you directly jump to the developer's help file from here. If you wish to use the search operation in developer's help, please use File|Open and select the DEVELOPR.HLP file, or click on the Developer's Help icon in your Crystal Folder.

Developer help is now contained in a help file named DEVELOPR.HLP. If you wish to search or browse for developer orientated topics, please see Developer Help Contents

Connecting to Paradox 4.0 with ODBC

When setting up an ODBC data source for Paradox 4.0, you need to supply three pieces of information in the setup facility in order for the OK button to become active:

- The directory name
- The network directory
- The user name

To set up the Network directory and User Name you click the Options button in the ODBC Paradox dialog box. This expands the dialog box so you can specify the needed information.

NOTE: The directory and network directory should be the same.

Unless you supply all three pieces of information, the OK button will remain grayed out and you will not be able to complete your setup.

MENU COMMANDS INDEX (BY MENU)

Menu commands are tools you use to create, customize, print, and save your reports. Each Crystal Reports menu contains commands that are related by function.

- Click on the command of interest from the following list:

File Menu

New

- Standard Expert
- Listing Expert
- Cross-Tab Expert
- Mailing Label Expert
- Summary Expert
- Graph Expert
- Top N Expert
- Drill Down Expert
- Another Report
- Custom Report
- Custom Cross-Tab
- Custom Mailing Label

Open

Save

Save As

Save Data with Report

Close

Print Preview

Print

- Preview Sample
- Printer
- Export
- Mail
- Crystal Reports Server
- Report Definition

Printer Setup

Set Label Layout

Page Margins

Options

Report Options

Exit

Edit Menu

Undo
Redo
Cut
Copy
Paste
Paste Special
Select Fields
Formula
Text Field
Summary Field
Browse Field Data
Show/Hide Sections
Delete Section
Object
Links

Insert Menu

Database Field
Text Field
Formula Field
Special Field
Page Number Field
Record Number Field
Group Number Field
Print Date Field
Subtotal
Grand Total
Summary
Group Section
Group Name Field
Line
Box
Picture
Graph/Chart Expert
Object

Format Menu

Report Style Expert
Auto Arrange Report

Font

Field

Border and Colors

Change Line Height

Line

Box

Picture

Graph/Chart

Section

Database Menu

Visual Linking Expert

Add Database to Report

Remove from Report

Set Location

Set Alias

Verify Database

Verify on Every Print

Log On Server

Log Off Server

Show SQL Query

Stored Procedure Parameters

Report Menu

Select Records Expert

Edit Selection Formula

Record

Group

Change Group Expert

Top N/Sort Group Expert

Sort Records

Search

Search Again

Zoom

Refresh Report Data

Report Title

Set Print Date

Window Menu

Tile Vertically

Tile Horizontally

Cascade

Arrange Icons

Close All

Help Menu

Contents

Search

Using Help

Register/Change Address

Crystal Library

About Crystal Reports

FILE MENU COMMANDS INDEX

The File menu includes commands you can use to open, close, and save files, to save files under a different file name, print the file to a printer, and create new report files. It also includes a command you can use to exit Crystal Reports and a command for making Mailing Labels. Additionally it contains a command that allows you to configure Crystal Reports to your specifications.

- Click on the command of interest from the following list:

New

- Standard Expert
- Listing Expert
- Cross-Tab Expert
- Mailing Label Expert
- Summary Expert
- Graph Expert
- Top N Expert
- Drill Down Expert
- Another Report
- Custom Report
- Custom Cross-Tab
- Custom Mailing Label

Open

Save

Save As

Save Data with Report

Close

Print Preview

Print

- Preview Sample
- Printer
- Export
- Mail
- Crystal Reports Server
- Report Definition

Printer Setup

Set Label Layout

Page Margins

Options

Report Options

Exit

EDIT MENU COMMANDS INDEX

The Edit menu allows you to modify aspects of your report. The menu includes commands you can use to edit formulas, text fields, and summary operations, to review field data, to change the position of items in a stack, or to delete group sections. It also contains commands for cutting, copying, and pasting text.

- Click on the command of interest from the following list:

Undo

Redo

Cut

Copy

Paste

Paste Special

Select Fields

Formula

Text Field

Summary Field

Browse Field Data

Show/Hide Sections

Delete Section

Object

Links

INSERT MENU COMMANDS INDEX

The Insert menu is the central menu you use for creating reports. The menu includes commands you can use to insert database, text, and formula fields; subtotals, grand totals, summaries (counts, averages, etc.), and group sections; print date, page number, record number and group number fields; and pictures, lines, and boxes.

- Click on the command of interest from the following list:

Database Field

Text Field

Formula Field

Special Field

Page Number Field

Record Number Field

Group Number Field

Print Date Field

Subtotal

Grand Total

Summary

Group Section

Group Name Field

Line

Box

Picture

Graph/Chart Expert

Object

FORMAT MENU COMMANDS INDEX

The Format menu includes commands for changing the look of the elements in your report. It includes commands for changing fonts, adding borders and colors and formatting fields, pictures, lines, and boxes. It also includes a command for formatting report sections.

- Click on the command of interest from the following list:

Report Style Expert

Auto Arrange Report

Font

Field

Border and Colors

Change Line Height

Line

Box

Picture

Graph/Chart

Section

DATABASE MENU COMMANDS INDEX

The Database menu is used to select and delete databases for use with your reports, to change the alias you use to identify the database, and to link and unlink databases. It also has a command, Set Location, for directing Crystal Reports to look for databases in new locations. Additionally, the menu contains commands that adapt your reports to changed database structures.

- Click on the command of interest from the following list:

Visual Linking Expert

Add Database to Report

Remove from Report

Set Location

Set Alias

Verify Database

Verify on Every Print

Log On Server

Log Off Server

Show SQL Query

Stored Procedure Parameters

REPORT MENU COMMANDS INDEX

The Report menu includes commands that let you select the records or groups to be included in your report and select the order in which report data is to be sorted (by record or by group). In addition, you can change the report title, change the Print Date, or compile your report into an executable format.

- Click on the command of interest from the following list:

Select Records Expert

Edit Selection Formula

Record

Group

Change Group Expert

Top N/Sort Group Expert

Sort Records

Search

Search Again

Zoom

Refresh Report Data

Report Title

Set Print Date

WINDOW MENU COMMANDS INDEX

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

- Click on the command of interest from the following list:

Tile Vertically

Tile Horizontally

Cascade

Arrange Icons

Close All

HELP MENU COMMANDS INDEX

The Help menu includes a command that takes you to Crystal Reports' main help index, a command to search for a topic in Help, and a command that will give you information about the Crystal Reports version you are using.

- Click on the command of interest from the following list:

Contents

Search

Using Help

Register/Change Address

Crystal Library

About Crystal Reports

New command (File menu)

You can create a new report with two different methods:

- the New Report Button, or
- the File|New Command

Using the File|New Command

1. When you select File|New, a submenu will appear with the following options:

   Standard Expert

   Listing Expert

   Cross-Tab Expert

   Mailing Label Expert

   Summary Expert

   Graph Expert

   Top N Expert

   Drill Down Expert

   Another Report

   Custom Report

   Custom Cross-Tab

   Custom Mailing Label

- Click a report above for further information.

Custom Report command (File|New)

Use **Custom Report** to create a new report from scratch. When you choose the Custom Report command while using another report, Crystal Reports creates a new report window and opens the new report in that window. The report you were working on remains unchanged in its own window.

Custom Cross▪Tab command (File|New)

Use File|New|Custom Cross▪Tab to create a Cross

- ▪Tab report. Cross
- ▪Tabs are reports that present summarized data in a row versus column structure similar to that of a spreadsheet. These reports are easy to read, and they are particularly useful for making rapid comparisons or identifying trends.

Custom mailing Label command (File|New)

Use the Custom Mailing Label command to create mailing labels. When you use the Custom Mailing Label command while using another report, Crystal Reports creates a new report window and opens the new report in that window. The report you were working on remains unchanged in its own window.

Standard Expert command (File|New)

This command leads you through the creation of a **Sort And Total** report with step-by-step help from the Create Report Expert.

A Sort and Total report is a standard report form in which data is broken into groups and the values in each group are totaled. *Orders by customer* reports, *sales by salesrep* reports, and *purchases by vendor* reports are examples of this kind of report.

Listing Expert command (File|New)

This command leads you through the creation of a **Listing** report with step-by-step help from the Create Report Expert

A Listing report is a standard report form in which data is presented in list format. *Customer lists*, *inventory lists*, and *employee lists* are examples of this kind of report.

Cross-Tab Expert command (File|New)

This command leads you through the creation of a **Cross-Tab** report with step

-by

-step help from the Create Report Expert.

A Cross-Tab report is a report that presents summarized data in a row versus column structure similar to that of a spreadsheet. These reports are easy to read, and they are particularly useful for making rapid comparisons or identifying trends.

Mailing Label Expert command (File|New)

This command creates Mailing Labels to your specifications with step-by-step help from the Create Report Expert. When you use the Mailing Label command while using another report. Crystal Reports creates a new report window and opens the new report in that window. The report you were working on remains unchanged in its own window.

Summary Expert command (File|New)

This command leads you through the creation of a Summary report with step-by-step help from the Create Report Expert.

A Summary report is a report that summarizes data without showing the details. In a typical summary report, data is broken into groups and the values in each group are summarized, but only the group totals - not the individual values in each group - are shown.

Graph Expert command (File|New)

This command leads you through the creation of a **Graph** with step-by-step help from the Create Report Expert. The program includes a wide variety of graph types to meet most of your graphing needs. When you use the Graph command while using another report. Crystal Reports creates a new report window and opens the new report in that window. The report you were working on remains unchanged in its own window.

See Also

Graph/chart creation topics

Top N Expert command (File|New)

This command leads you through the creation of a **Top N** report with step-by-step help from the Create Report Expert. A *Top N* report is a report that focuses on the top N groups or the bottom N groups, *N* being a value that you specify. For example, if you want to find who the top five sales reps are in the company, you might do a sales by sales rep report using the Top N expert to print only the information for the top 5 reps. When you use the Top N command while using another report. Crystal Reports creates a new report window and opens the new report in that window. The report you were working on remains unchanged in its own window.

Drill Down Expert command (File|New)

This command leads you through the creation of a Drill Down report with step-by-step help from the Create Report Expert. A Drill Down report is a summary report that you can analyze in the Preview Window using the program's drill down facility. When you double-Click any summary value in the report, the program will create a tab that displays the details behind the selected summary. When you use the Drill Down command while using another report. Crystal Reports creates a new report window and opens the new report in that window. The report you were working on remains unchanged in its own window.

NOTE: When you close the summary report, your Drill Down tabs are not saved.

Another Report command (File|New)

Use the Another Report command to creates a new report based upon one that already exists. The program will make a duplicate of the original report. You can then modify the duplicate report however you please to create your new report. When you use the Another Report command while using another report. Crystal Reports creates a new report window and opens the new report in that window. The report you were working on remains unchanged in its own window.

Configuring Crystal Reports to work like you do

Crystal Reports lets you customize many of the program's default settings to fit the way you work.

These settings affect such things as:

- Your working environment.
- The way you select databases.
- SQL and ODBC access.
- The way various data types are formatted. and
- The fonts you use for fields and text.

To change your default settings, select File|Options and the Options dialog box appears. Then you can select the tab appropriate to the option you want to change. With the options feature these changes are easy to make.

Open command (File menu)

Use **Open** to open an existing report. When you select **File|Open,** the **File Open** dialog box appears. Use this dialog box to select the report file you wish to open.

Save command (File menu)

Use the **Save** command to save the active report to disk under its current name.

- All changes you have made while working on the report will be saved.
- The previous version of the report will be overwritten.

Save As command (File menu)

Use the Save As command from the File menu to save the active report to disk under a new name. All changes you have made while working on the report will be saved to the new file. Your original report file will remain unchanged.

Save Data with Report (File menu)

Typically, when you save a report, Crystal Report saves only the report definition. The definition contains all of the information needed for printing your report, but it doesn't include the required data. When you print the report, export it, or work with it in the Preview Window, the program has to run the report and retrieve the data first. Depending on the complexity of your report and the amount of data involved, this can take anywhere from a few moments to an hour or more. If you plan to work again with the report in the near future (fine tuning it, etc.) or if you plan to share the report with someone else who wouldn't have access to the original data, you can use the Save Data With Report option on the File menu.

NOTE: When the program saves data with the report, it compresses the data to take up less disk space. It decompresses the data when it opens the report.

There are some things to consider before using the Save Data With Report option:

If you save only the definition:

- your report will require somewhat less disk space (enough for the template only), but
- you will need to rerun the report before you print it.
- If you save your data with the report:
- your report will require somewhat more disk space (enough for the report template and the data), but
- you won't need to rerun the report before you print it.

If you have enough disk space, and processing time or access to the raw data are major considerations, you may find Save Data With Report to be a real convenience. Click this command to toggle it on (arrow showing). Click it again to toggle it off.

Close command (File menu)

Use the Close command to close out a report while remaining in the Crystal Reports program or to close an active print window.

Print Preview command (File menu)

Use **Print Preview** to see what your report will look like before you print it and to fine tune the report while working with the actual data.

See Also

The Preview Window

Print command (File menu)

Use File|Print to select a printing option. When you select File|Print, a submenu appears listing the Preview Sample, Printer, Export, Mail, Crystal Reports Server and Report Definition options.

Preview Sample (File|Print menu)

Use File|Print|Preview Sample if you want to print a draft report to a window using only a subset of the data you'll use in your final report. This feature can save you a great deal of time during the report creation process.

Printer command (File|Print)

Use File|Print|Printer to print a hard copy of your report.

NOTE: File|Print|Printer is also available as a button on the Button Bar. Clicking the button has the same effect as selecting File|Print|Printer.

Export command (File|Print)

The Export command enables you to print your report to an electronic file in a format that can be read by another software application. For example, you could export your report in Microsoft Excel format and then open it in Excel as a normal spreadsheet file.

The Export facility supports several popular word processing, database, and spreadsheet formats and a number of standard data interchange formats as well.

The following is a partial list of export formats available. See the complete list when you scroll the drop down list after activating the command:

Word processor specific formats

Word for Windows

Word for DOS (16 bit only)

Word Perfect (16 bit only)

Spreadsheet specific formats

Lotus 1-2-3 (WKS)

Lotus 1-2-3 (WK1)

Excel 2.1 (XLS)

Excel 3.0 (XLS)

Excel 4.0 (XLS)

Excel 5.0 (XLS)

Workgroup specific formats

Lotus Notes (16-bit only)

NOTE: 32-bit versions of Crystal Reports do not support exporting to Lotus Notes format.

With Crystal Reports you have the ability to export data to Lotus Notes format.

To export to Lotus Notes format from Crystal Reports, simply activate File|Print|Export and select Lotus Notes from the Destination drop down list and select the desired export format from the format drop down list in the Export dialog box. For more information, see the Export command topic.

Common data interchange formats

- *Comma separated values* (CSV) encloses alphanumeric field data in quotes and separates fields with commas.
- *Tab separated values* presents data in tabular form. Encloses alphanumeric field data in quotes and separates fields with tabs.
- *Character separated values* encloses alphanumeric field data in quotes and separates fields with the character of your choice. When you make this selection, select your destination, and click OK, the Character Separated Values dialog box appears. Enter your Separator and Quote selections in the respective edit boxes.
- *Separator* specifies the character you want to use to separate the fields in the Character separated value format.
- *Quote* specifies double or single quotation marks to enclose alphanumeric field data in the Character separated value format.
- *Text style* saves the data in ASCII text format with all values separated by spaces. This style looks most like the printed page.

- *Tab separated text style* saves the data in ASCII text format with all values separated by tabs.
- *DIF style* saves the data in DIF (data interchange format) format. This format is often used for the transfer of data between different spreadsheet programs.

If Crystal Reports does not export directly to your application's native format, it more than likely exports to a format your application can read nonetheless. For example, many applications can read data saved in an ASCII or text format. Even though your application isn't on the list of supported formats, you may be able to export to a text file and then have your application read in the data (import it) from that format. Check your application's documentation to see which formats it can import.

Crystal Reports also allows you to attach an exported file to a message sent via an electronic mail system. The message displays an icon that indicates a report is attached. The person who receives the message Double Clicks the icon and the report appears on screen.

NOTE: The Export command replaces and expands upon the Print to File command from earlier versions of Crystal Reports.

Mail command (File|Print)

When you select File|Print|Mail, the Export dialog box appears. You can export your report to mail using this dialog box.

Crystal Reports Server command (File|Print)

If you have the Crystal Report Manager installed on your system, when you select File|Print|Crystal Reports Server, the program opens the Crystal Reports Manager. Once open, you can use the Crystal Report Manager for submitting reports to the Crystal Report Server system.

NOTE: The Crystal Report Manager must be in your PATH in order for the program to find it.

Report Definition command (File|Print)

A report definition is a report on a report; it identifies the components of the report, and it provides important information about each of the components. Report Definition prints a copy of the report definition for the active report. The report definition will be printed to the printer.

Printer Setup command (File menu)

Use Printer Setup to select the printer you want to use to print the report. If you don't select a printer, Crystal Reports will print to the Windows default printer.

Page Margins command (File menu)

Use Page Margins to set page margins for your report. Margins define the white space between your report and the edges of the page.

NOTE: Crystal Reports uses the non-printing areas established for your printer as default printer margins. Those margin settings appear in the Printer Margins dialog box. While you can set margins that fall inside the non-printing areas, parts of your report may be clipped off if you do.

Report Options (File menu)

When you select File|Report Options from the menu, the Options Per Report dialog box appears. You use the check boxes to change options for the report you currently have open.

The options are:

- Convert Date
- Time to Date checkbox
- Convert Null Field to Default check box

Select this option to use the Crystal Reports default when a null value is encountered.

- Use Index for Speed checkbox
- Translate DOS Strings checkbox
- Translate DOS Memos checkbox
- More Print Engine Error Messages check box

Select this option to receive more detailed Print Engine Error Messages when they arise.

- Case Insensitive SQL Data checkbox
- Preview Pages Start With
- Convert Report

Case-insensitive SQL data

The following example assumes the use of a value called "apple".

Sorting, record selection, subtotals, and other summaries can be affected at times by case sensitivity issues (selecting records based on "apple" ignoring records with value "Apple" or "APPLE"). This version of Crystal Reports now enables you to specify that you want SQL data treated on a case insensitive basis ("apple", "Apple", and "APPLE" all treated the same). You do this by toggling the "Case Insensitive SQL Data switch "on" in the options section of the Options Per Report dialog box. You access that dialog box via File|Report Options

Convert database driver on next refresh

Crystal Reports provides a wide range of database drivers. Some of these drivers support more than one kind of database and some databases can be used with two or three different drivers. Depending on your report and the data source youre using, you may find it faster to run the report using one driver instead of the others. To help you make your reporting more efficient, the program has a new feature that allows you to specify the driver the program uses when running your report.

This new option lets you direct the program to use a different database driver (PD*.DLL):

- the next time it refreshes data

or, if you arent saving data with the report,

- the next time you open the report.

NOTE: You can only convert to a different database driver if the original report was created using PDBBDE, PDBXBSE, PDBPDX, or PDBJET drivers. If the original report was created using one of the above drivers, you may convert to either PDBBDE, PDBXBSE, PDBPDX, or PDBJET. If you open any other report that was not created using one of these drivers, or you open a report that uses multiple databases, the Convert Report section will be greyed out.

The option is found in the Convert Report section of the Options Per Report dialog box. You access that dialog box via the File|Report Options command.

You toggle the option on and off using the check box.

When you use the option, the program displays the database driver currently in use in the "From" box. It lists all of the database drivers found on the system in the "To" box. Select the driver you want to convert to in the "To" box and click OK. The program attempts to run the report with the driver you specify.

NOTE: There is no verify feature on this option. You may select a driver that cannot be used for your report. When you attempt to run the report you will get an error message that tells you the report can not be run. To correct for this, return to the Options Per Report dialog box, select the original DLL or another DLL that is valid and run the report again.

NOTE: When you run a report via the print engine, the program does not automatically convert to the new driver. The print engine uses the driver as specified in the report. If you specified a new driver with the report but did not refresh your data or reopen the report (if no data saved) before printing via the print engine, the report will print using the old driver.

Crystal Dictionaries

When you set the switch in Report Options, Crystal dictionaries are not automatically converted. To convert the dictionaries to use with a different driver, you will need to make the following changes to these three lines in the CDW.INI file.

```
DoAutoDbDllConvert=Yes

FromDbDllName="name of dll currently in use"
(in quotes, for example "PDBXBSE.DLL")

ToDbDllName="name of dll you want program to use"
(in quotes, for example "PDBBDE.DLL")
```

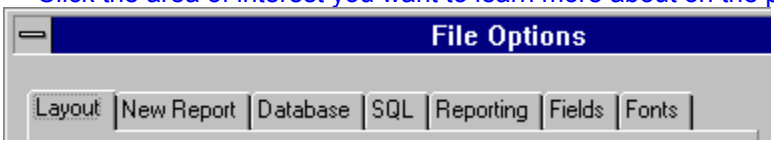
NOTE: When you change these INI settings and open the dictionary, only one kind of driver can be converted at a time. If you need to change the drivers for more than one kind of database (dBASE and Paradox both to IDAPI, for example), you will need to set the INI settings for one of the databases, open and save the dictionary, then revise the settings for the second database, and open and save the dictionary again.

Options command (File menu)

Use Options to change default settings for your work with Crystal Reports. By resetting the defaults, you can customize Crystal Reports to fit the way you work.

NOTE: Crystal Reports stores your default options in the file CRW.INI, located in the Windows directory.

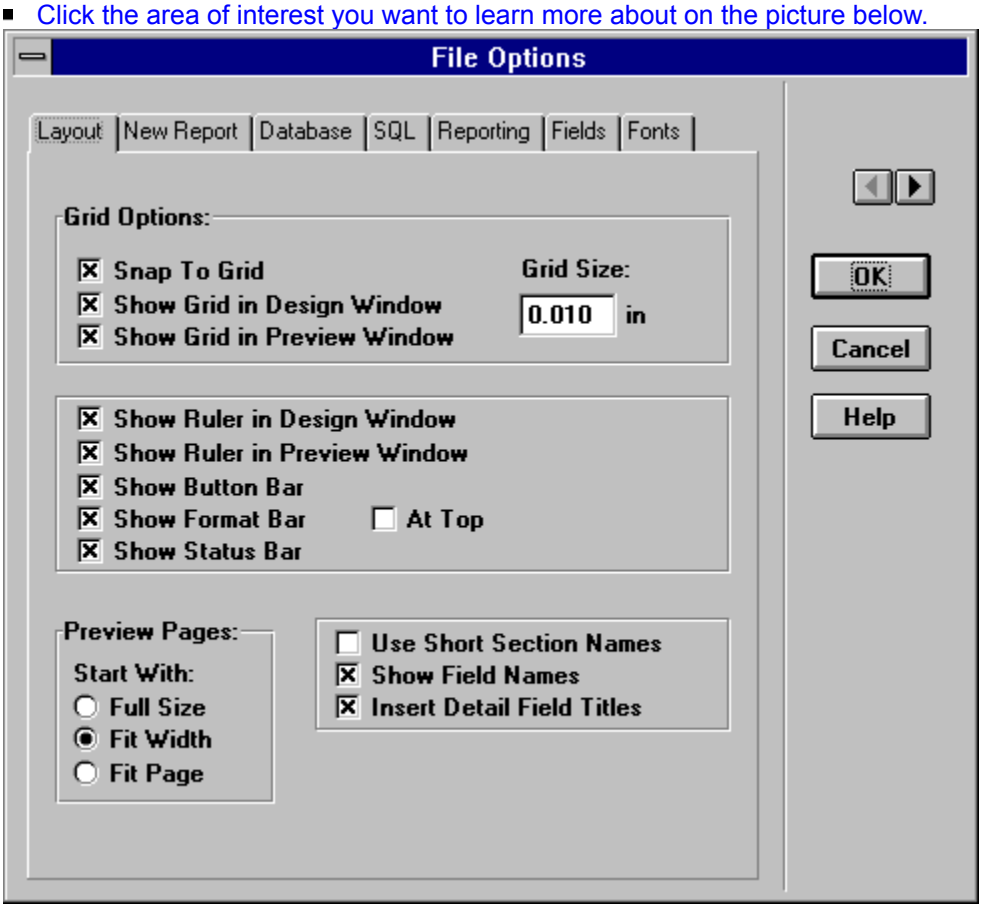
- Click the area of interest you want to learn more about on the picture below.



Layout Tab

The Layout Tab lets you specify the way the program displays the grid, ruler and button bar. The Layout Tab also lets you specify how pages appear in the Preview Window, and how field and section names and titles are displayed.

- Click the area of interest you want to learn more about on the picture below.



The Default Dictionary Browse button

When you click the Default Dictionary Browse button, the File Open dialog box appears. From here, you can select the Dictionary you wish to use as your default.

The Arrow buttons

Use the Left and Right arrow buttons to switch between tabs in File Options.

The Report Directory Browse button

When you click the Report Directory Browse button, the Set Directory dialog box appears. You can use the Directories scroll box to select the default Report Directory of your choice. Click OK to set the Report Directory, and return to Crystal Reports.

The Data Directory Browse button

When you click the Data Directory Browse button, the Set Directory dialog box appears. You can use the Directory edit box or the Directories scroll box to select the default Data Directory of your choice. Click OK to set the Data Directory, and return to Crystal Reports.

Grid Size edit box

This selector lets you specify the size of the grid increments. The unit of measure used for grid increments is based on your settings in the International section of the Windows Control Panel.

Use Short Section Names check box

When you toggle the check mark on, Crystal Reports abbreviates the section names that appear in the gray area on the left side of the Design Window and narrows the gray area at the same time. This adds extra width to the report page, and it allows you to work with more of your report on screen. When you toggle the check mark off, Crystal Reports displays full section names and a narrower working area. By default, the Short Section Names check box is toggled on.

Show Ruler in Design Window check box

When you toggle the check mark on, the program will display the ruler in the Design Window. By default, the check mark is toggled on.

Show Ruler in Preview Window check box

Show Ruler in Preview Window check box is a toggle. When the checkmark is visible the option is toggled on, and the Ruler will appear and be available for use in the Preview Window. When the checkmark is not visible, the option is toggled off, and the Ruler will not be available in the Preview Window.

Preview Pages

The Preview Pages box contains three options that let you specify which level of magnification you want the program to use as a default when displaying reports in the Preview Window. Select the magnification you want.

Start With: Full Size

Full size allows you to view your report at the highest magnification for easy reading. The page is larger than a standard VGA Preview Window. When you resize the Preview Window, the report size remains unchanged. All that changes is the amount of the report you can view at one time. You may need to use scroll bars to see all sections of your report.

Start With: Fit Width

Fit Width adjusts the width of the report to the width of the Preview Window. When you resize the Preview Window (within realistic limits), the program resizes the page image as well. Thus, if you want to see a larger image, you can expand the window

Start With: Fit Page

Fit Page adjusts the width and the length of the report to let you see a full page of your report in the Preview Window. When you resize the Preview Window (within realistic limits), the program resizes the page image as well. Thus, if you want to see a larger image, you can expand the window

By default, the selector is set to Fit Width.

NOTE: All of the functions available in the Preview Window are available at every magnification.

Refresh data on every print

When you toggle the check mark on, the program automatically retrieves new data every time you print. When you toggle the check mark off, the program will use existing data. By default the Refresh data on every print check box is toggled on.

Show field names check box

When you toggle the check mark on, Crystal Reports displays the field name for each field on the report. When you toggle the check mark off, crystal Reports displays symbolic characters (XXXX, 555, etc.) instead of the field name for each field. By default, the Show Field Names check box is toggled on.

Snap to grid check box

When you toggle the check mark on, the program will snap the selected data field to the nearest grid position. If you toggle the check mark off, Crystal Reports will display the selected data field where you place it on the report. By default, the check mark is toggled on.

Show button bar check box

When you toggle the check mark on, Crystal Reports displays the button Bar. By default, the Display button Bar check box is toggled on.

Show Grid in Design Window

When you toggle the check mark on, the program will display the grid in the Design Window. By default, the check mark is toggled on.

Show Grid in Preview Window

When you toggle the check mark on, the program will display the grid in the Preview Window. By default, the check mark is toggled on.

Show Format Bar check box

When you toggle the check mark on, crystal Reports displays the Format Bar. By default, the Display Format Bar check box is toggled on.

Show Status Bar check box

When you toggle the check mark on, crystal Reports displays the Status Bar. by default, the Display Status Bar check box is toggled on.

Insert Detail Field Titles check box

When you toggle the check mark on, Crystal Reports automatically enters a default field title in the Page Header section whenever you insert a field in the Details section of the Design Window. The default title is the name of the field, underlined. When you toggle the check mark off, Crystal Report will not insert field titles. By default, the Insert Detail Field Titles check box is toggled on.

Save Data With Closed Report

When you toggle the check mark on, the program automatically saves database data with your report. When you toggle the check mark off, the program saves the report itself, but it will need to retrieve the report data once again the next time you run the report.

Typically, when you save a report, Crystal Report saves only the report definition. The definition contains all of the information needed for printing your report, but it doesn't include the required data. When you print the report, export it, or work with it in the Preview Window, the program has to run the report and retrieve the data first. Depending on the complexity of your report and the amount of data involved, this can take anywhere from a few moments to an hour or more. If you plan to work again with the report in the near future (fine tuning it, etc.) or if you plan to share the report with someone else who wouldn't have access to the original data, you can use the Save Data With Report option.

NOTE: When the program saves data with the report, it compresses the data to take up less disk space. It decompresses the data when it opens the report.

There are some things to consider before using the Save Data With Report option:

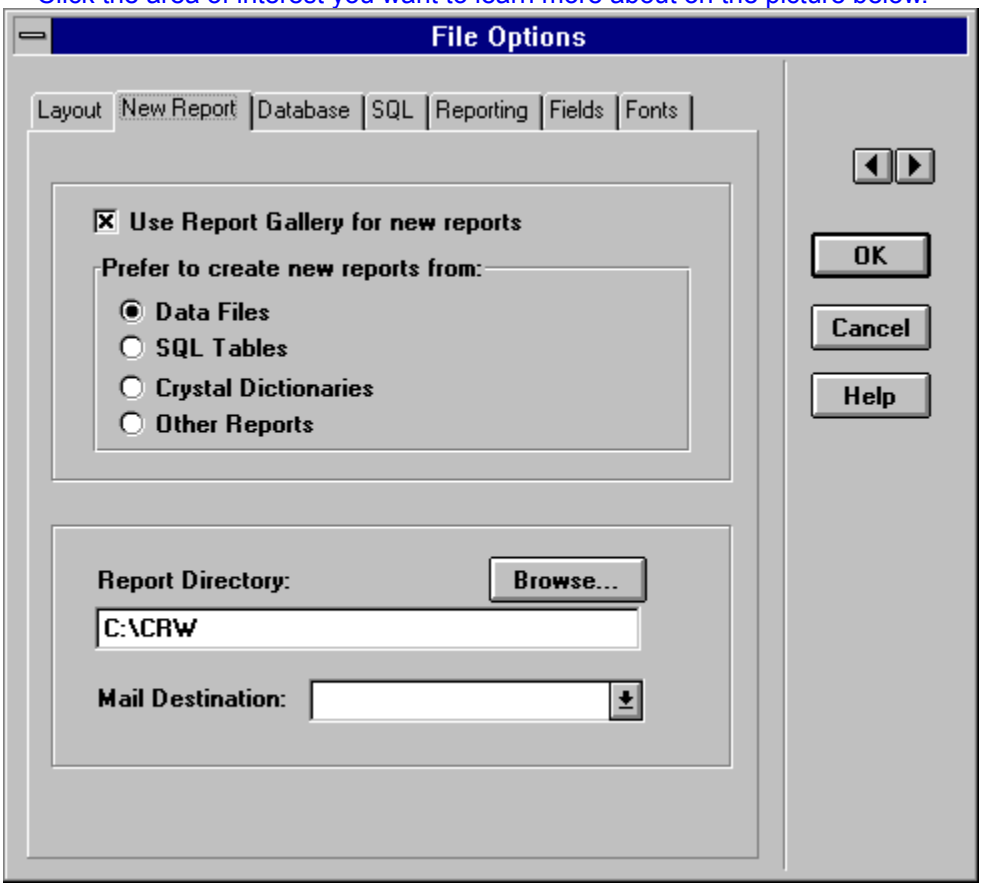
- If you save only the definition:
- your report will require somewhat less disk space (enough for the template only), but
- you will need to rerun the report before you print it.
- If you save your data with the report:
- your report will require somewhat more disk space (enough for the report template and the data), but
- you won't need to rerun the report before you print it.

If you have enough disk space, and processing time or access to the raw data are major considerations, you may find Save Data With Report to be a real convenience.

New Report Tab

The New Report Tab lets you toggle the Report Gallery on and off, specify what type of files will be used when creating new reports, select the default Report Directory, and select the default Mail Destination.

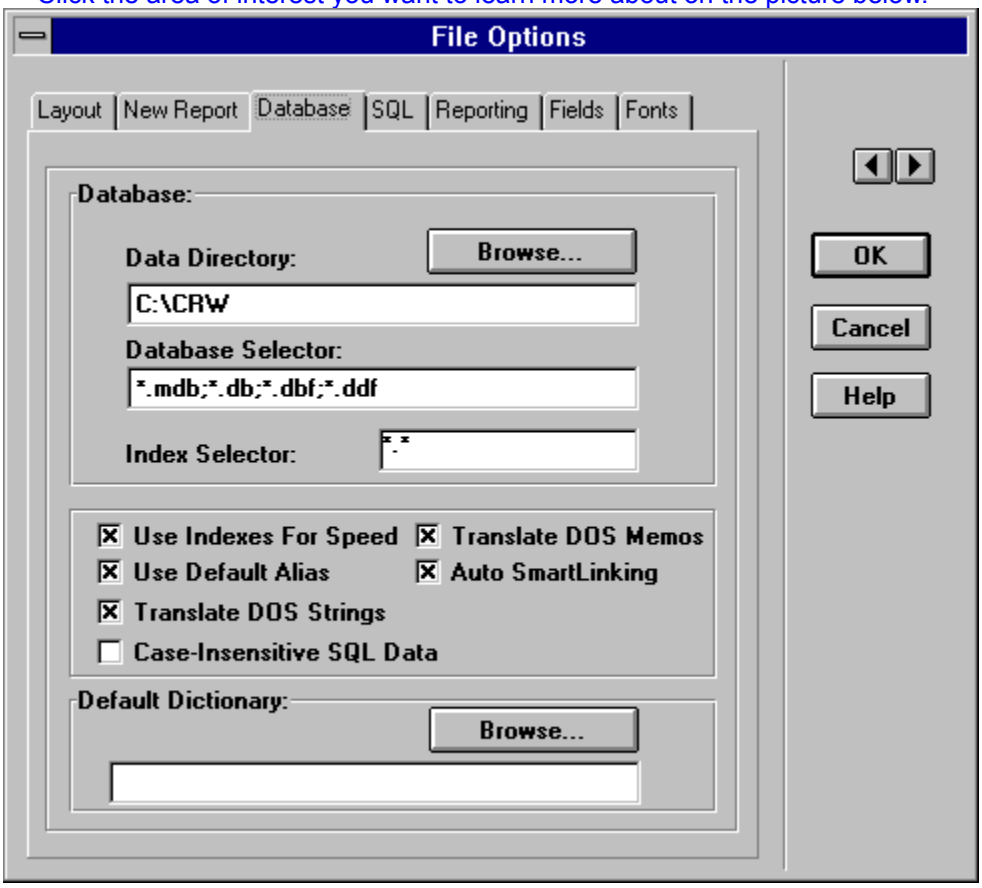
- Click the area of interest you want to learn more about on the picture below.



Database Tab

Database options let you specify the Data Directory, Database Selector, Index Selector and Default Dictionary.

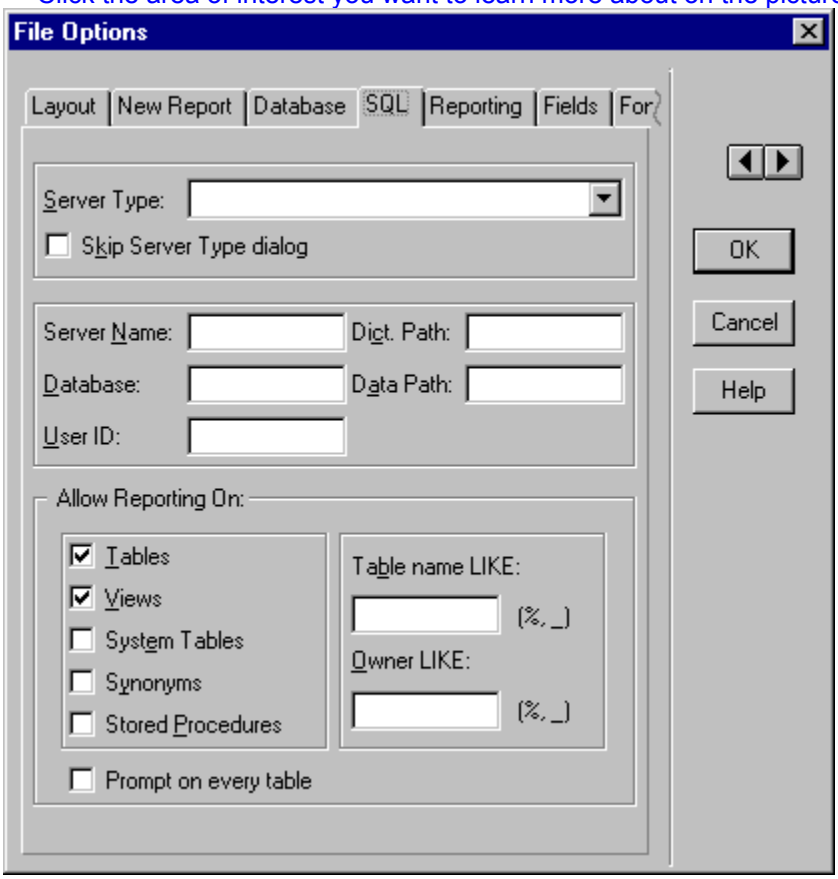
- Click the area of interest you want to learn more about on the picture below.



SQL Tab

SQL options let you specify the Server Type to be used, the Server Name, the Database, the Dictionary and Data paths and several Reporting options.

- Click the area of interest you want to learn more about on the picture below.



Skip Server Type Dialog

Activate this option if you want to bypass the Log On Server dialog box altogether.

NOTE: When you bypass the Log On Server dialog box, you no longer have the option to activate a non-SQL database for the report or to activate a server other than the default server. Activate this option only if you plan to create all of your reports using a single server.

Server Name edit box

Use this box to enter the name of the SQL server you want to log on to.

Database edit box

Use this box to enter the name of the database you want to activate in the specified SQL server.

User ID edit box

Use this box to enter the name you use to log on to the specified server.

Dict Path edit box

When using Netware SQL, use this box to enter the path for the data dictionary

Data Path edit box

When using Netware SQL, use this box to enter the path for the data files. Enter the requested login information and click OK when finished. Crystal Reports logs you onto the specified server and takes you to the Choose SQL table dialog box.

Allow Reporting On:

Use the options in this section to select the data you want to allow reporting on. The six reporting options are activated by check box toggle switches.

Tables:

When you toggle the check mark on, the program will report on database tables. By default, the Tables check box is toggled on.

Views:

When you toggle the check mark on, the program will report on virtual tables. By default, the Views check box is toggled on.

System Tables:

When you toggle the check mark on, the program will report on system tables. These tables are typically used by the system administrator only but are available for use if you have the appropriate permissions. By default, the System Tables check box is toggled off.

Synonyms:

When you toggle the check mark on, the program will allow reporting on virtual tables that are available on some hosts. By default, the Synonyms check box is toggled off.

Stored Procedures:

When you toggle the checkmark on, the program will allow reporting on the result sets from stored procedures if you are using SQL systems that support stored procedures.

Prompt On Every Table

When you toggle the check mark on, the program will display the Allow Report On dialog box before it displays the Choose SQL Table dialog box. The Allow Reporting On dialog box enables you to specify the kinds of data you want to appear in the Choose SQL Table dialog box. All of the "Allow Reporting On" options from the SQL Tab appear in the Allow Reporting On dialog box.

Table name Like

This option enables you to use the SQL LIKE function 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 (state.Companies but not company.Employee).

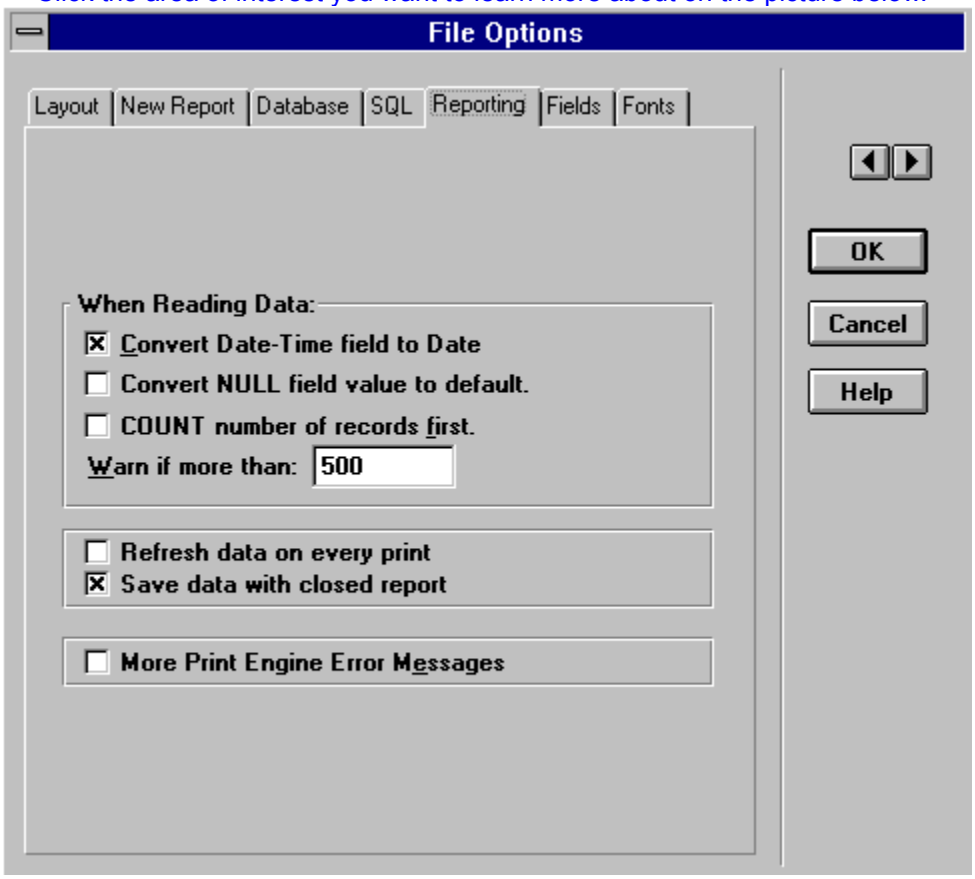
Owner Like

This option works exactly like Table name Like except you use the LIKE function to select the Owner (or Creator or Alias) of the table, not the table name itself. For example Owner Like C% would display only those tables that had an owner beginning with the letter C (company.Employee but not state.Companies).

Reporting Tab

Reporting options let you place a "governor" on data selection and they let you specify how you want Date-Time strings handled if they are used in your report and whether or not data is saved with your report.

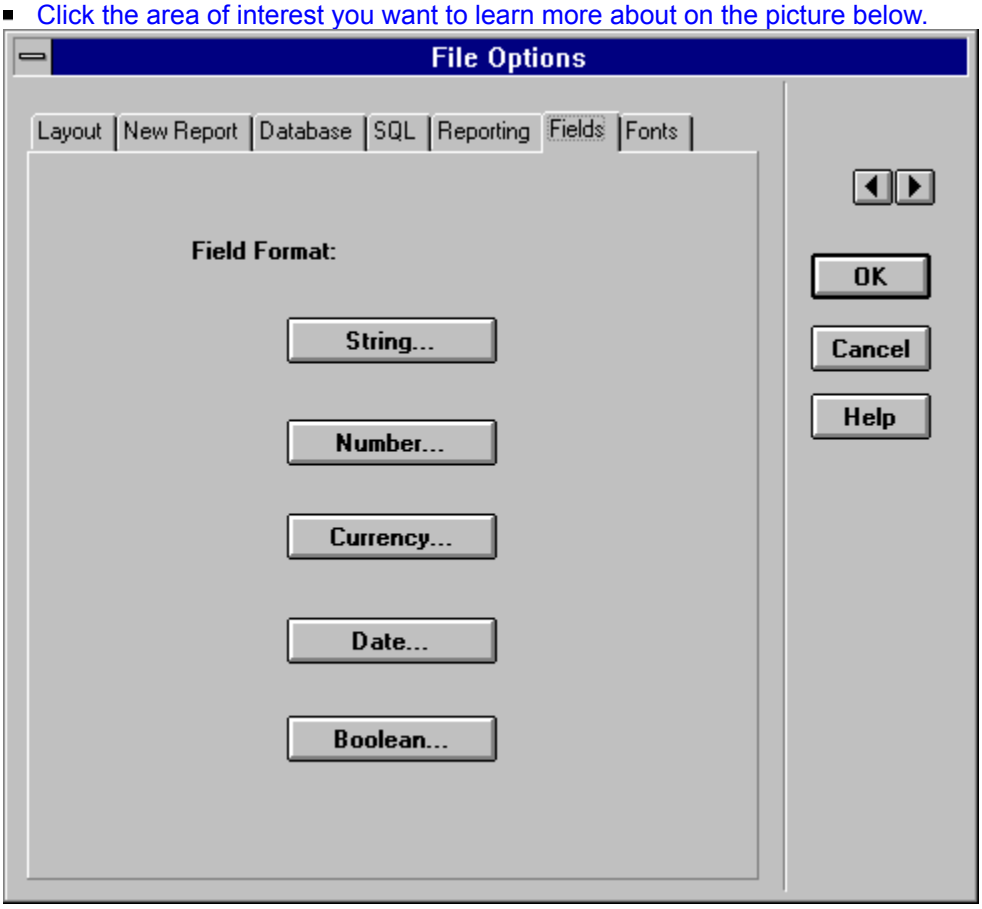
- Click the area of interest you want to learn more about on the picture below.



Fields Tab

Fields options let you specify the way different kinds of fields will appear in your report. When you Click the Fields tab, five buttons appear. The buttons correspond to the five Crystal Reports data types: string, number, currency, date, and Boolean. When you click one of the buttons, the Field formatting options specific for that data type appear.

- Click the area of interest you want to learn more about on the picture below.



When do the changes apply?

The changes you make in this box apply only to elements that you insert after you make the changes. Elements that you entered before you changed the default retain their previous format.

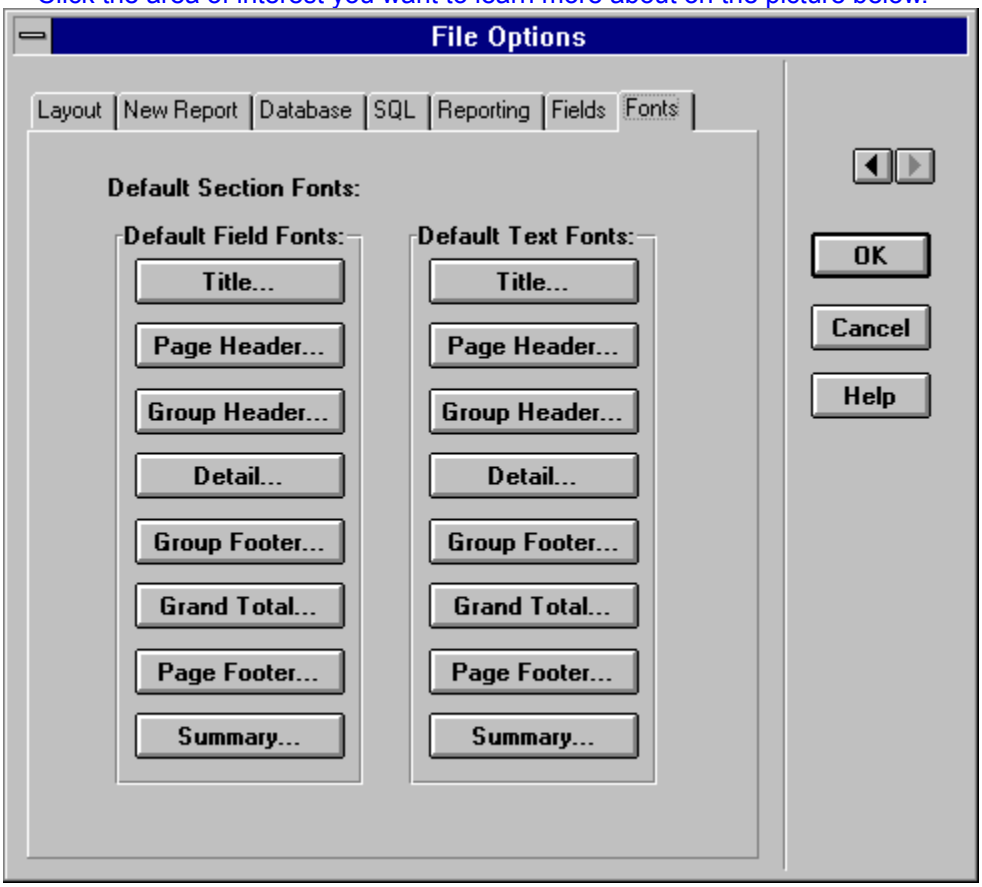
All five data types allow you to:

- change the alignment of a text value within the space allotted for the field in the report,
- suppress the value so it does not print if it is identical to the value that preceded it, and
- cause the value to hide when printing.

The Fonts Tab

The Fonts Tab allows you to change the default fonts for any and all sections of your report.

- Click the area of interest you want to learn more about on the picture below.



Fonts Tab

By default, any text, data field, or formula result entered anywhere on the report is displayed and printed in the Windows default font. The Font Tab allows you to change the default fonts for any and all sections of your report. It allows you:

- to specify different fonts for different sections of your report, and
- to specify a different font for text elements than is used for fields within a given section.
- For new reports, once the defaults are changed, Crystal Reports looks at the section in which each new text element is placed and formats it with the text font specified for that section. It also looks at the section in which each new data field or formula result element is placed and formats it with the field font specified for that section.
- For existing reports, the default font for text in a section stays the same. New reports will get the new default.

For example, you can specify one field font for all group values (subtotals, summary fields, etc.) and a completely different text font for the text labels you use to identify each of the group values. Crystal Reports automatically formats all group values with the default field font, and it formats any label you type in the group section with the default text font.

NOTE: The Font Tab gives you the opportunity to customize the Crystal Reports Design Window to best fit your needs. When you make changes via this button, Crystal Reports simply changes the defaults so the fonts used in each section appear in the format you typically want them in. These default changes don't in any way limit the fonts available for use in any section of your report, however. You still have the ability to reformat text or field elements individually if you wish.

When you click the Font tab, two columns of buttons (your Default Font options) appear in the Options dialog box:

- the column on the left for changing the default fonts for fields, and
- the column on the right for changing the default fonts for text strings.

The buttons in each of the columns match sections of your report.

To change the font for a specific section:

- determine first if you want to change the font for text or the font for fields and formulas. This tells you which list to use.
- determine the section for which you want to change the default.
- Click the button that matches the element and section for which you want to change defaults, and
- change the default font using the Font dialog box.

Title

The Title Section, typically the part of your report that contains the title, and other information you want to appear only on the first page of your report.

Page Header

The Page Header Section, typically the part of your report that contains the date and other identifying information.

Group Header

The Group Header section. Of the two sections created each time you set up a group field (subtotal, summary field), this is the top section, the section that appears above the Details section.

Detail

The Details section, typically the body of your report.

Group Footer

The Group section that displays or prints the group value. Of the two sections created each time you set up a group field (subtotal, summary field), this is the bottom section, the section that appears below the Details section.

Grand Total

The Grand Total section that typically appears at the end of the report.

Page Footer

The Page Footer section, typically the part of your report that contains the page number, words like *(continued)*, and other identifying information.

Summary

The Summary section, typically the part of your report that contains a summary, or other information you want to appear only on the last page of your report.

Field Formatting Options

String

String here refers to text strings drawn directly from a database, or formulas that result in text strings. It does *not* refer to text strings typed directly into the Design Window. For string fields you can configure the program to print the fields on multiple lines if necessary and you can specify the maximum number of lines on which it can print.

Number

Number refers to number fields drawn directly from a database, and formulas that, as an end result, return a number. For number fields you can:

- suppress its value if it is a zero,
- round the value,
- insert or exclude a zero before the decimal point in a fractional number,
- specify the number of decimal places, and the character used to separate decimals from whole numbers,
- insert or exclude thousands separators, and specify the character used as that separator,
- specify the way negative numbers should be displayed,
- select the currency symbol,
- specify whether the symbol appears in a fixed position or floats with the length of the value,
- specify whether the symbol appears before or after the value, and
- cause the currency symbol to print only once per page if you so wish.

For detailed information on number formatting options.

Currency

Currency refers to currency fields and formulas that, as an end result, return a dollar amount. For currency fields you can do everything that you can do with a number field.

Date

Date refers to date fields drawn directly from a database, formulas that result in dates, and dates placed via the Insert|Print Date Field command. For date fields you can:

- set the order of the date elements (month
- day
- year, military date order, etc.),
- specify the separators (commas, hyphens, etc.), if any, between the elements, and
- set the style of the date elements (month as a number, month spelled out, abbreviated year, etc.).

Boolean

Boolean refers to Boolean (Yes/No) fields drawn directly from a database and formulas that return a Boolean value. For Boolean fields you can:

- specify the display of field values as True or False, T or F, Yes or No, etc.

To change the default field format, Click the button that represents the field data type you want to change. Crystal Reports takes you to a dialog box that contains formatting options for the type of field you have selected.

NOTE: This Formats button gives you the opportunity to customize the Crystal Reports Design Window to best fit your needs. When you make changes via this button, Crystal Reports simply changes the defaults so that each new data type appears in the format you typically want it in. These default changes don't in any way limit the formats available for use in your report, however. You still have the ability to reformat elements individually if you wish.

See Also

Format Number

Format Currency

Format Date

Format Boolean

Format String

Count number of records first

You would typically use this and the "Warn if more than" option if you want to limit the number of records that can be included in a report. When you toggle the check mark on, Crystal Reports will count the number of records first before building your report. By default, the checkmark is toggled on.

Convert Date▪Time field to Date

You use this option if you want Date/Time string fields from Microsoft Access automatically converted to date format..

Warn if more than: edit box

You use the Warn if more than: edit box to specify the maximum number of records the program can use in a report without first giving you a warning prompt. This option is only active if you have activated the Count number of records first option as well.

Server Type scroll box

Use the Server Type scroll box to preselect a server type. Use the scroll arrow and select a server type from the scroll list. Then, when the Log On Server dialog box appears, your selection will be highlighted. If you then want to log on to that server, all you need to do is click OK or press Enter.

The Data Directory edit box

You use the Data Directory edit box to set the default directory that Crystal Reports should use in searching for and displaying databases. If you set the directory pointer in this box to the Samples directory, for example, Crystal Reports will display only files from that directory when it displays the Choose Database File dialog box and other boxes that offer you database selection options.

- If you know the name of the directory, you can type it into the Data Directory text box.
- If you don't know the name of the directory, Click the Set Location button and select the directory from the list that appears in the Set Directory dialog box.

Set Directory dialog box

This dialog box appears whenever you Click the Set Location button in either the Data Directory or Report Directory box on the New Report Tabwhen using the File|Options command.

Use this dialog box to select the default Data Directory and/or Report Directory.

The Database Selectors edit box

You use the Database Selectors edit box to set the file specifications that Crystal Reports should use in searching for and displaying databases. If you set the specifications in the Selector text box, for example, to Paradox specifications (*.db), Crystal Reports will display only files with that specification in the Choose Database File dialog box and other boxes that offer you database options.

NOTE: If you want to set multiple specifications, separate them with semicolons (*.db; *.dbf; etc.)

The Index Selector edit box

You use the Index Selector edit box to set the file specifications that Crystal Reports should use in searching for and displaying indexes. If you set the specifications in the Selector text box, for example, to Paradox specifications (*.px), Crystal Reports will display only files with that specification in the Index file dialog box and other boxes that offer you index options. Indexes are of major importance in establishing links between multiple databases.

To change the selections already listed, type your new selection over the existing selection, and delete any old characters that remain.

Use Indexes For Speed check box

The Use Indexes For Speed option is a toggle.

- When you toggle the check mark on, Crystal Reports uses available indexes to speed the record selection process.
- When you toggle the check mark off, Crystal Reports selects records without the use of indexes (a much slower process).

NOTE: If this option is on the File | Options Database tabs this applies only to new reports. It will not affect any currently loaded or previously saved reports. If you wish to change the indexes for speed option for a currently loaded report, use the FILE | Report Options ■ Use Indexes for speed option.

NOTE: If an index has a different name than the database it indexes, you need to identify the index for Crystal Reports. You do this using the Link Options dialog box, which is accessed via the Visual Linking Expert.

Case Insensitive SQL Data Check Box

This check box specifies whether or not the SQL data you use in the report becomes case sensitive. If the data you are using contains fields where the data is in both upper case and lower case, and you do not wish to categorize the data by this aspect, mark this check box. I

NOTE: Case Sensitive SQL data is not supported in the 32-bit standard edition of Crystal Reports

Auto Smart Linking Check box

Use this check box to specify if Crystal Reports automatically smart ■ links your tables when you are using the Visual Linking Expert.

Use Default Alias check box

The Default Alias check box is simply a toggle.

- When you toggle the check mark on, Crystal Reports uses the default alias for each database you activate. The default alias is the name of the database in lower case and without the extension. For example, the default alias for the database *Employee.db* is *employee*. Use this option if you are usually satisfied with the default alias and don't want to have to accept or change the alias whenever you activate a database. By default, the Use Default Alias check box is toggled to *on*.
- When you toggle the check mark off, Crystal Reports displays the Alias Name dialog box whenever you activate a database. This dialog box gives you the option to accept the default alias or to type in a new one.

NOTE: This option does not prevent you from later changing the alias for any active database. If you want to change an alias, you can do so using the Database|Set Alias command.

Translate DOS Strings check box

This option is like the Translate DOS Memos option. The only difference is that it is active for string fields instead of memo fields.

NOTE: If this check box is toggled OFF and you have used upper ASCII characters in your dBASE string fields, the special characters will not be the same in your report as they are in the dBASE string field.

By default, the Translate DOS Strings check box is toggled *On*.

Translate DOS Memos check box

When you insert special characters in dBASE memo fields, you use the upper ASCII character set (those characters with decimal values between 128 and 255). When you insert special characters into Windows programs, you use ANSI codes to do so. If you include upper ASCII characters in dBASE memo fields and then use those fields to create a report in Crystal Reports for Windows, the program would read the codes and assume they were ANSI codes were it not for the Translate DOS Memos option.

- When Translate DOS Memos is activated (a check in the check box), the program assumes that any character code it finds is an ASCII code and it translates that code to a corresponding ANSI value so the same character that appears in dBASE appears in your report.
- When the option is not checked, the program assumes that any character code it finds is already an ANSI code and it does no further translation.

NOTE: If this check box is toggled OFF and you have used upper ASCII characters in your dBASE memo fields, the special characters will not be the same in your report as they are in the dBASE memo field.

By default, the Translate DOS Memos check box is toggled On.

Default Dictionary edit box

You use the Default Dictionary edit box to specify a default Dictionary if you want one. Once you have selected a default Dictionary, the program opens that Dictionary automatically whenever you select File| New, bypassing both the Report Gallery (if you've activated it) and the File Open dialog box.

- If you know the name and path of the Dictionary, you can type it into the Default Dictionary edit box.
- If you don't know the name or the path of the Dictionary, Click the Browse button and select the Dictionary from the list that appears in the File Open dialog box.

Use Report Gallery for new reports check box

This check box is a toggle. When you toggle the check mark on, the program displays the New Report dialog box (the Report Gallery). This special selection window enables you to select the type of data you want to use and the type of report you want to create. If you toggle the check mark off, the program bypasses this dialog box and takes you directly to the Choose Database File dialog box (or a different data source based on your Preferred Data Source selections). By default, the check box is toggled *on*.

Prefer to create new reports from: selector

This selector lets you specify the kind of data source that you want the program to turn to by default. For example, if you generally build reports using Crystal Dictionaries, you can check that source. Then, when the File Open dialog box appears, it will display only Dictionary (.dct) files. Also, when the Report Gallery appears (if you have that option toggled on), it will already have the Dictionary button depressed. This option gets you right to the data you need most often without having to make additional selections each time you create a report. *Data Files* is the default selection.

The Report Directory edit box

You use the Report Directory edit box to set the default directory that Crystal Reports should use in searching for and displaying existing reports. This box (and the Set Location button) work the same way as the Data Directory box.

Mail Destination selector

The Mail Destination selector lets you specify the destination you want the program to use when you export a report to E■mail. Click the arrow on the selector combo

■box and Click a destination from the list that appears. Your choices are:

- Mail via MAPI,
- Mail via VIM, and
- possibly other mail destinations.

Once you've made a selection, when you open a mail■related dialog box in the program it will open with your specified destination already selected.

See Also

Export command(File|Print)

File|Report Options

When you select File|Report Options from the menu, the Options Per Report dialog box appears. You use the check boxes to change options for the report you currently have open.

The options are:

- Convert Date
- Time to Date checkbox
- Convert Null Field to Default check box

Select this option to use the Crystal Reports default when a null value is encountered.

- Use Index for Speed checkbox
- Translate DOS Strings checkbox
- Translate DOS Memos checkbox
- More Print Engine Error Messages check box

Select this option to receive more detailed Print Engine Error Messages when they arise.

- Case Insensitive SQL Data checkbox
- Preview Pages Start With
- Convert Report

Exit command (File menu)

Use **Exit** to end your Crystal Reports session and return to the Windows environment.

When you select **File|Exit**:

- If you have already saved your active file(s), Crystal Reports closes itself down and returns you to Windows.
- If you haven't already saved your active file(s), Crystal Reports asks you if you want to save before closing. Refer to File|Close for your options at this point.

Undo command (Edit menu)

The Undo command is a dynamic command. The text changes on the menu based on the last action you performed in Crystal Reports. The command supports multiple levels of Undo. Undo reverses the effects of an action and restores your document to the way it was before you performed the action.

Redo command (Edit menu)

The Redo command is a dynamic command. The text changes on the menu based on the last Undo you performed in Crystal Reports. The command supports multiple levels of Redo. Redo restores the effects of an Undo and returns your document to the way it was before you used Undo.

Cut command (Edit menu)

Use the Cut command any time you wish to cut or delete selected text temporarily and hold it in the clipboard for later use.

NOTE: The Cut command can be used in the Design Window (for text) and in the Formula Editor (for any formula element) via the Windows keyboard command Shift-Delete.

Copy command (Edit menu)

Use the Copy command any time you wish to send a copy of text to the clipboard for later use without disturbing the text as it appears currently in the report.

NOTE: The Copy command can be used in the Design Window (for text) and in the Formula Editor (for any formula element) via the Windows keyboard command Ctrl+Ins.

Paste command (Edit menu)

Use the Paste command any time you want to paste (insert) text from the clipboard into your report. You can use the Paste command:

- with the Cut command to move text, or
- with the Copy command to complete the copying process.

NOTE: The Paste command can be used in the Design Window (for text) and in the Formula Editor (for any formula element) via the Windows keyboard command Shift+Ins.

Paste Special command (Edit menu)

The Paste Special command allows you to paste an OLE object into your report from the Clipboard. Use the following procedure to paste an OLE object.

1. Use an object application (an application that allows you to create OLE objects) to create the object you would like to paste into your report, or open an existing file in the object application. For example, if you would like to use part of the ARCHES bitmap, a standard bitmap that comes with Windows, open ARCHES.BMP in Paintbrush (the object application).
2. Copy all or part of the object to the clipboard using the Edit|Copy command.
3. Exit the object application.
4. Open your report in Crystal Reports.
5. Select Edit|Paste Special. The Paste Special dialog box appears.

- If you want to embed the object in your report, Click the Paste option button.
- If you want to insert a link to an object, Click the Paste Link option button.

6. In the As type list box, select the type of object you want to link or embed. For example:

- if you want to paste the ARCHES bitmap object from the Paintbrush application, select Paintbrush Picture Object.
- if you want to paste a picture of your object (a picture that can't be edited via OLE), select Metafile or Bitmap.
- If you want the program to use an icon to represent the object in your report, Click the Display As Icon check box to turn it on.

7. Click OK and a placement rectangle appears.
8. Using the mouse, position the rectangle where you want the object to appear in your report and Click the left mouse button to place it. The object is now embedded (or linked) in your report.

See Also

Edit|Object

Insert|Object

Select Fields command (Edit menu)

Use Select Fields to activate the selection lasso. When the lasso is active, you can select any field or group of fields in your report simply by drawing a box around them.

Formula command (Edit menu)

Use Edit Formula to edit a formula once it has been entered in a Crystal Reports report.

See Also

Formulas ■ an overview

Index To Formula Topics

Text Field command (Edit menu)

Use Edit Text Field to change the content of text fields that you entered via the Insert|Text Field command. You can add text, delete text, rearrange text, or change the spelling of text in any text field in your report.

NOTE: The Windows Cut, Copy, and Paste commands are active in the Enter Text edit box.

Summary Field command (Edit menu)

Edit|Summary Field allows you to change the operation for the selected summary. This option, for example, can change a summing operation to one that determines the maximum (highest) value in the group.

Browse Field Data command (Edit menu)

Use Browse Field Data any time you want to take a look at the values in a report field.

NOTE: This command is active only when you have selected a field on your report.

Show/Hide Sections command (Edit menu)

The Show/Hide Sections command allows you to hide a report section that is showing or to show a report section that is hidden. Hide and Show are also available through the right mouse button menu.

Change Group Expert command (Report menu)

Use Report|Change Group Expert to change the sorting and grouping specifications (sort and group by field, sort direction, etc.) for any of the groups on your report. If you insert a group section, a subtotal, or a summary and later want to modify it in any way, you use this command to do so.

See Also

How to identify the "top" groups

Delete Section command (Edit menu)

Use Delete Section to remove group sections from your report. This command is active only if your report contains a group section.

Object command (Edit menu)

Use Edit|Object to modify an OLE object that you have embedded in a report. The command is active only when you select an embedded object.

Links Command (Edit menu)

Use Links to make changes in existing OLE object links. Using this command, you can change the source data for a link and the way a link is updated. This command is only active when you have an object link in the report.

NOTE: This command is not related to editing a database link. To edit a database link you must choose the Visual Linking Expert from the Database menu.

Graphing/Charting with Crystal Reports

Crystal Reports offers you several formats and options for adding graphs and charts to your reports. You can select the Type of graph/chart, the Layout of the graph/chart, the Text to be used in the graph/chart and additional Options for the graph/chart as well. The graph/chart feature and options are available from both the Design and the Preview Window.

- Click the Tab of interest you want to learn more about on the picture below.

See Also

Editing Graph/Chart topics

- # Editing Graph/Chart topics
- Click on a topic of interest from below for more information:

  Specifying TopN or Bottom N groups before graphing

  To insert a Graph/Chart

  To Move a Graph/Chart "by the numbers"

  To Move or Resize a Graph/Chart using the ruler and positioning wand

  To Resize a Graph/Chart using the Graphic Format dialog box

  To Resize a Graph/Chart with the resizing handles

  Graph/chart creation topics

Specifying Top N or Bottom N groups before graphing

You may find it useful to limit the number of groups that appear in your report if you plan to create a graph. By limiting the number of groups you can make your graph easier to read and to understand. You can limit the number of groups using the Top N/Sort Group Expert.

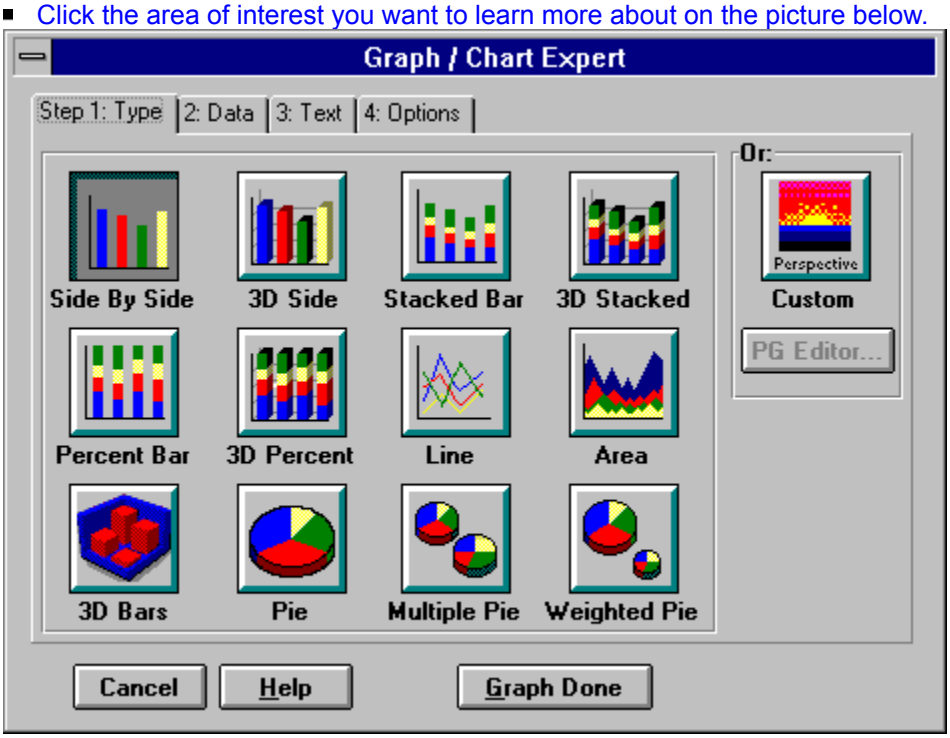
Specifying Top N or Bottom N groups before building a Cross-Tab report

You may find it useful to limit the number of groups that appear in your report if you plan to create a Cross-Tab report. By limiting the number of groups you can make your report easier to read and to understand. You can limit the number of groups using the Top N/Sort Group Expert.

The Type Tab

The Type Tab offers a choice of 12 different graph/chart types to insert in your report. To select a graph/chart type Click on its button. Crystal Reports then builds a graph/chart of that type, which you can further customize.

- Click the area of interest you want to learn more about on the picture below.



See also

Editing graph/chart topics.

Graph/Chart Types

Click on any one of these buttons to choose which of the 12 graph/chart types to add to your report.

PG Editor button

The PGEditor is a powerful graph editor offering you 80 graph types and full formatting control of every graph element. The PGEditor is commonly used to format the graph title, change the colors of graphs, and resize portions of the graph.

A very popular feature of the PGEditor is the ability to save graph formatting templates. These templates are like style sheets for graphs. They contain custom formatting instructions that can be applied instantly to new graphs. If you continually create pie charts, for example, and you need the charts to appear in custom colors, you can create a template with those colors. Then, whenever you create a new graph, you simply apply the template and the graph is formatted to your template specifications.

The Help button

When you Click the Help button, Crystal Reports Help opens.

The Cancel button

When you Click the Cancel button you are returned to Crystal Reports, and any changes you made in the Graph/Chart Expert are not applied.

The Graph Done button

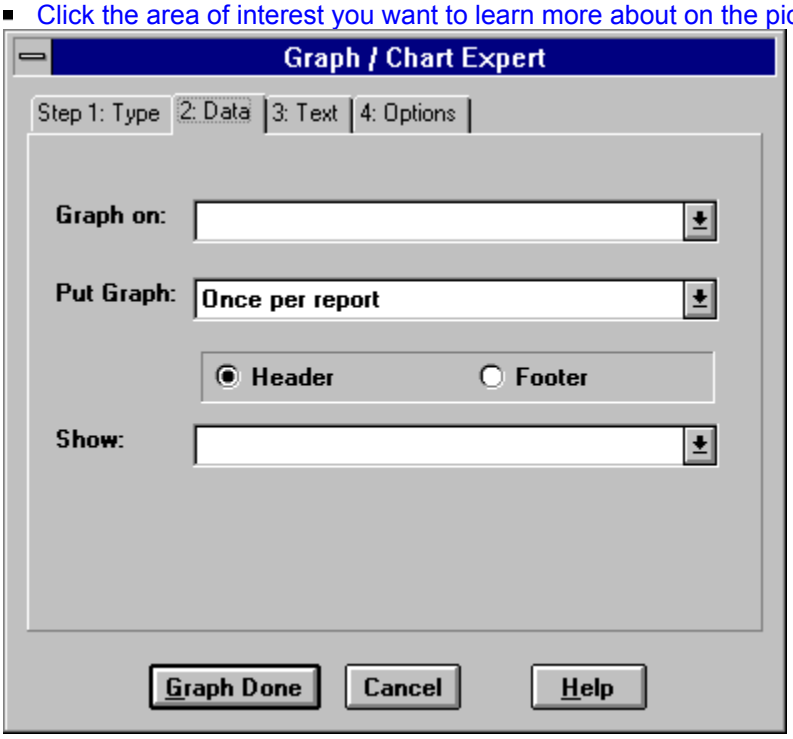
You Click the Graph Done button when you are finished creating or editing a graph/chart. The program returns you to your report and the graph/chart is inserted into your report.

The Data Tab

The Data Tab lets you specify which fields the graph/chart will be based upon.

NOTE: If you are creating a graph/chart for a Cross-Tab report, your Data Tab choices will be different than those pictured below. Click Here to see the graph/chart Data Tab for Cross Tab reports.

- Click the area of interest you want to learn more about on the picture below.



See also

Editing graph/chart topics

- **The Data Tab (For Cross**
- **Tab reports only)**

The Data Tab lest you specify which fields the graph/chart will be based upon.

- Click the area of interest you want to learn more about on the picture below.

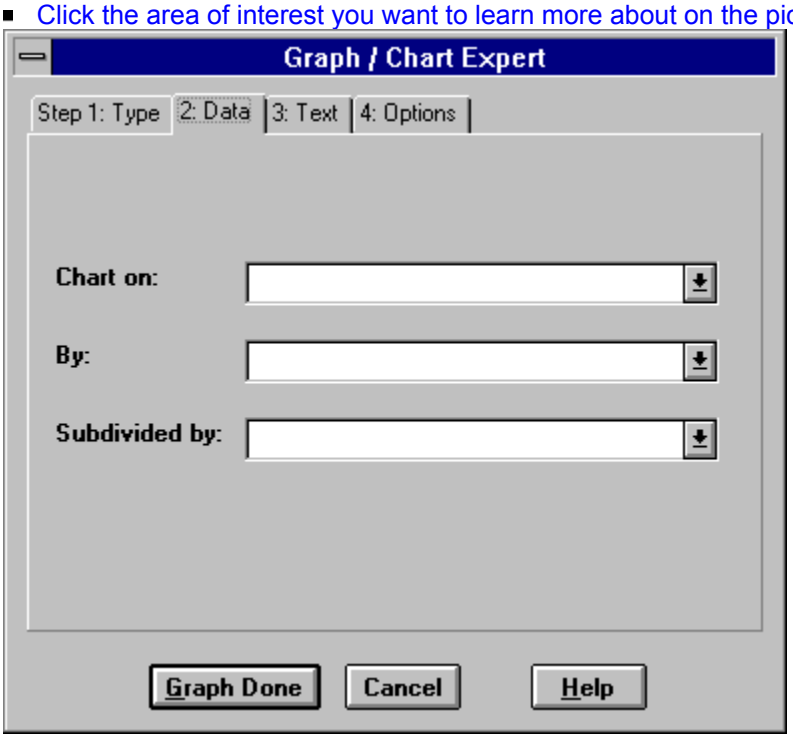


Chart on

Use the **Chart on** edit box to select the value you want to present in your graph/chart. For example, if you wanted a graph/chart that presented the subtotal of sales when the sales representative changes, you would **Chart on** the subtotal of sales, and By the sales representative.

Graph on

Use **Graph on** to choose the summary field to base your graph/chart on. Clicking the **Graph on** box displays a list of summary fields in your report that you can choose from.

Put Graph

Use **Put Graph** to choose the report section where your graph/chart will appear. Your graph/chart can appear either Once per report or once per group. (The group section containing the summary field that the graph/chart is based upon.)

NOTE: Your choice of where to put the graph will affect how the data in the graph can be subdivided.

By

Use the **By** list box to select the field that triggers the subtotal. For example, if you wanted a chart/graph that presented the subtotal of sales when the sales representative changes, you would **Chart on** the subtotal of sales, and By the sales representative.

Subdivided by

If you have at least two subtotal sections in your report, use the Subdivided by list box to specify how you want your graph/chart subdivided. If you do not want subdivisions, select None from the drop down edit box.

Company sales subdivided by regional sales then by representative sales is an example of a subdivision.

NOTE: When a group section has been selected in "By", it will not be available in "Subdivided by".

Header Footer radio buttons

Use the header footer radio buttons to display your graph/chart in either the group header or the group footer. If your graph/chart appears Once per report, then the header footer choice will place it in either the title section or the summary section of the report.

Show

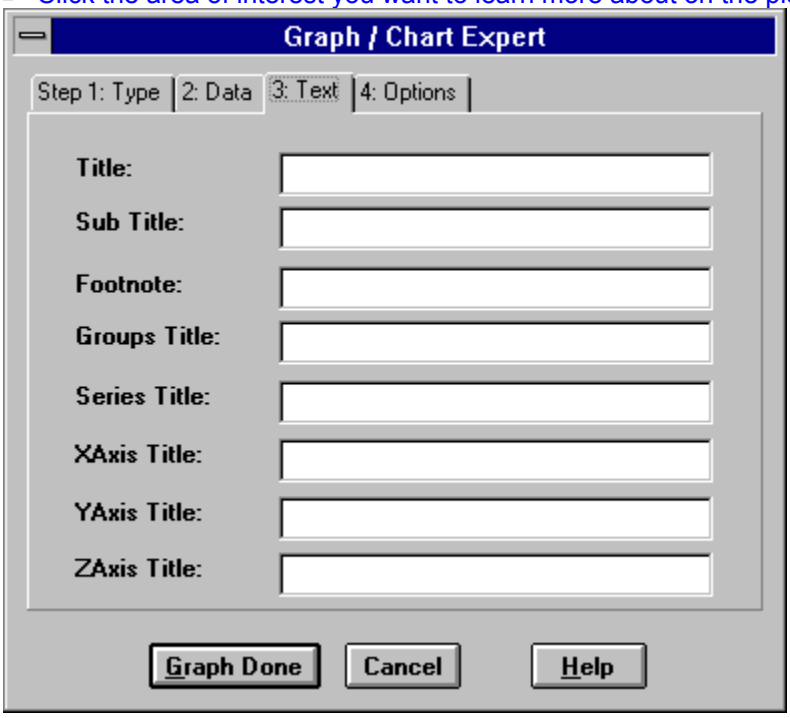
The **Show** list box lists additional ways to subdivide the information in your graph/chart if your data allows for it. This feature is especially useful for stacked bar graphs. For instance, lets say you have created a stacked bar graph showing total sales to each of your customers for the past year. If your database included the sales person who made each sale, the **Show** feature would let you subdivide each bar on your graph by the portion of the total sale made bye each sales person.

The choices available to you with Show feature are affected by where you chose to place your graph with Put Graph.

- # The Text Tab

The Text Tab lets you specify titles, footnotes and axis titles for your graph/chart.

- Click the area of interest you want to learn more about on the picture



See also

Editing graph/chart topics.

Title edit box

In the Title edit box, enter the Title you want to appear above the graph/chart.

Subtitle edit box

In the Subtitle edit box, enter the Subtitle you want to appear below the Title. Toggle the checkmark on to display the subtitle below the Title. Toggle the checkmark off to hide the Subtitle. By default, the checkmark is on.

Footnote edit box

In the Footnote edit box, enter the Footnote you want to appear at the bottom of the graph/chart. Toggle the checkmark on to display the Footnote. Toggle the checkmark off the hide the Footnote. By default, the checkmark is on.

Series Title edit box

In the Series Title edit box, enter the Series Title you want to use for the Series. Toggle the checkmark on to display the Title. Toggle the checkmark off to hide the title. By default, the checkmark is turned off.

Groups Title edit box

In the Groups Title edit box, enter the Group Title you want to use for the Groups. Toggle the checkmark on to display the Title. Toggle the checkmark off to hide the Title. By default, the checkmark is turned off.

X Axis Title edit box

In the X Axis edit box, enter the X Axis for the graph/chart. Toggle the checkmark on to display the X Axis title. Toggle the checkmark off to hide the X Axis title. By default the checkmark is turned off. This option is for two or three dimensional graph/charts

Y Axis Title edit box

In the Y Axis edit box, enter the Y Axis for the chart. Toggle the checkmark on to display the Y Axis title. Toggle the checkmark off to hide the Y Axis title. By default, the checkmark is turned off. This option is for two or three dimensional graph/charts.

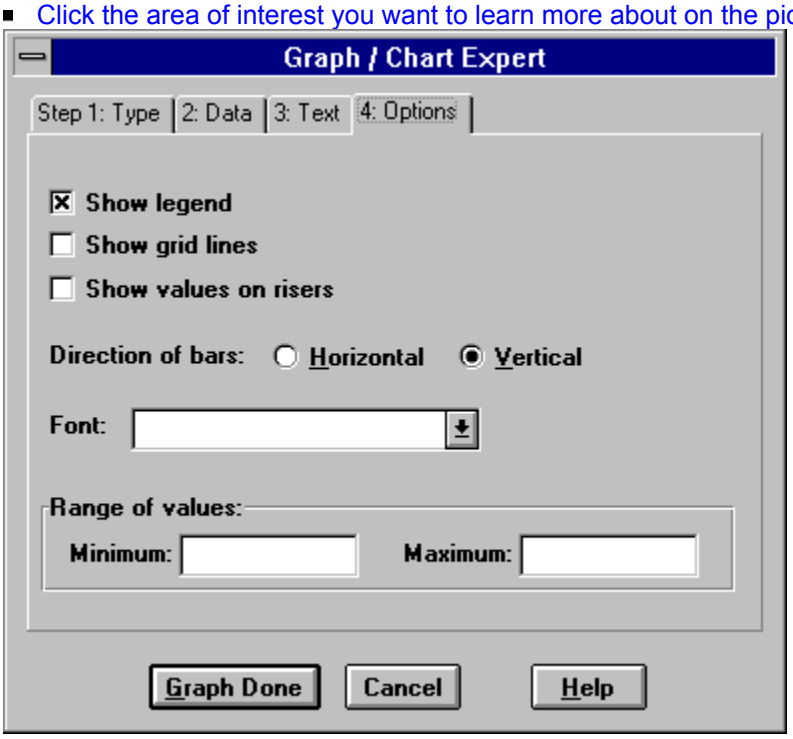
Z Axis Title edit box

In the Z Axis edit box, enter the Z Axis for the Graph/Chart.. Toggle the checkmark on to display the Z Axis title. Toggle the checkmark off to hide the Z Axis title. By default, the checkmark is turned off. This option is available for three dimensional graph/charts only.

The Options Tab

The Options Tab offers choices for displaying your graph/chart.

- Click the area of interest you want to learn more about on the picture.



See also

Editing graph/chart topics.

Show data value on every riser check box

When this check box is toggled on, the number value associated with each riser on the graph/chart is displayed. This is helpful when you want the exact value of a riser to be apparent, rather than estimated by the viewer of your graph/chart. By default, the checkmark is toggled off.

Show grid line check box

The Show grid line check box is a toggle option. When you toggle the option on, grid lines are displayed on graph/charts, unless the graph/chart is a pie graph/chart. When you toggle the option off, no grid lines are displayed on graph/charts. By default, the check box is turned off.

Show Legend

The Show Legend check box is a toggle. When you toggle the option on, the Legend is displayed for your chart/graph. When the checkmark is not visible, the option is toggled off, and no legend will be displayed for your graph/chart. By default, the option is toggled on.

Value ranges

The Range Of Values box has two edit boxes in it: Minimum and Maximum.

- Enter the Minimum value you want Crystal Reports to use for creating your graph/chart in the Minimum box.
- Enter the Maximum value you want Crystal Reports to use for creating your graph/chart in the Maximum box.

Font

Use this box to choose the font for all the text in your graph/chart. Click anywhere on the box to see a list of fonts to choose from.

Bars Direction

- Horizontal
- When this toggle switch is selected, graph/charts that display data with bars will have horizontal bars.
- Vertical
- When this toggle switch is selected, graph/charts that display data with bars will have vertical bars.

By default, Vertical is selected in the Bars Direction section.

To Insert a Graph/Chart

1. Select Insert|Graph/Chart Expert, or click on the Graph/Chart button on the button bar. The Format Graph/Chart dialog box appears with the Type Tab selected.

   Twelve graph/chart types appear as buttons.

NOTE: For the Graph/Chart feature to be available, you must have a report open in either the Design Window, or the Preview Window, and a summary field in the report.. If there is no report open, Crystal Reports will not insert a graph/chart when you select the graph/chart feature..

2. Click the Type of graph/chart that will illustrate your data best. Click OK and Crystal Reports inserts the graph/chart at the upper left hand corner of the report and returns you to the report with the graph/chart selected and resizing handles visible. You can now move the graph/chart anywhere you want it in your report, and resize the graph/chart to fit your needs.

NOTE: When you insert a graph/chart, it may cover a portion of your report. You will need to move the graph/chart and possibly resize it to fit your report the way you want it to.

To move a graph/chart "by the numbers"

Alternatively, you can move a graph/chart "by the numbers" using the Graphic position dialog box. To do this:

1. Position the mouse cursor inside the border of the graph/chart and Click the right mouse button once to select the graph. The New Graph/Chart pop menu appears.
2. Select Change Position from the menu. The Graphic Position dialog box appears.
3. Enter the coordinates you want for the left and top edges of the graph/chart.

- The Left coordinate is the distance from the left margin of the report to the left edge of the graph/chart.
- The Top coordinate is the distance from the top margin of the report to the top edge of the graph/chart.

4. Click OK to return to the report. The program moves the graph/chart to the specified position.

You can also move a graph/chart to the left or right on the report page using the positioning wand in Crystal Reports, located just beneath the ruler.

Moving a graph/chart with the positioning wand requires:

- that Show Ruler in Design Window be selected from the Layout Tab in File Options (if you want to work with the ruler in the Design Window),
- that Show Ruler in Preview Window be selected from the Layout Tab in File Options (if you want to work with the ruler in the Preview Window), and
- that the graph/chart is selected.

- ## To move or resize a graph/chart visually using the positioning wand

Moving or resizing a graph/chart using the repositioning wand requires that the graph/chart be selected first.

To Select a Graph/Chart

If the resizing handles are not visible on the border of the graph/chart, Click once anywhere inside the graph/chart border. When the resizing handles appear, the graph/chart is selected.

To Move a Graph/Chart visually using the ruler and positioning wand

1. Select a graph/chart.
2. Move the graph/chart to the left or right using the ruler and positioning wand. See instructions on using the ruler in Crystal Reports

To Resize a Graph/Chart visually using the ruler and positioning wand

1. Select the Graph/Chart.
2. Position the cursor over one end of the positioning wand. The cursor will change to a resizing cursor.
3. Drag the cursor right or left until the graph/chart is the size you want it and release the mouse button when finished.
4. Repeat steps 2 and 3 until the graph/chart has the desired size.

To resize a graph/chart using the Graphic Format dialog box

You can resize a graph/chart "by the numbers" using the Graphic Format dialog box. To do this:

1. Position the mouse cursor inside the border of the graph/chart and Click the right mouse button once. The New Graph/Chart pop■ menu appears.
2. Select Change Format from the menu. The Graphic format dialog box appears with the following options:

- *Cropping of Original a*llows cropping of the original graph/chart you inserted in your report.
- *Scaling* allows scaling of the original graph/chart you inserted in your report.
- *Size* allows you to specify the size of the picture in your report.
- *Hide When Printing* hides the graph/chart whenever you print the report.

3. Click OK to return to the report page. The graph/chart will be resized to your specifications.

To resize a graph/chart with the resizing handles

You can resize a graph/chart by using the resizing handles located on the border of the graph/chart.

1. Position the mouse cursor inside the border of the graph/chart and click once. The resizing handles appear on the border of the graph/chart.
2. Position the cursor over one of the resizing handles. The cursor will change to a double arrow repositioning cursor.

- Use one of the four resizing handles on the corners of the graph/chart border to increase or decrease the size of the graph/chart, while keeping the scale of the original graph/chart.
- Use one of the four resizing handles on the sides of the graph/chart border to increase or decrease the size of the graph/chart when you are not concerned with keeping the scale of the original. Handles on the sides of the graph/chart border allow you to increase the width of the graph/chart and the height of the graph/chart independently of each other.

3. Click and drag the graph/chart to its new size. Release the mouse button to make the graph/chart take on its new size.
4. Repeat steps 2 and 3 until you are satisfied with the new graph/chart size.

To Edit a Graph/Chart

To Edit a Graph/Chart

1. Select the Graph/Chart you want to edit.
2. Select Format|Format Chart The Format Graph/Chart dialog box appears with the Type Tab selected.
3. Edit the changes to your graph/chart using the tabs in the Format Graph/Chart dialog box.
4. Click OK to apply the changes to your graph/chart.

NOTE: When you select Edit|Chart, Crystal Reports will apply your changes to the graph/chart that was selected at the time of editing. This is fine if you only have one graph/chart in your report. However, if you have more than one graph/chart, be sure that you have selected the graph/chart you want to edit. Otherwise, you may edit a graph/chart that you did not intend to edit.

Alternatively, you can use the right mouse button to open the New Graph/Chart pop■ menu.

1. Position the mouse cursor inside the border of the graph/chart and Click the right mouse button once. The New Graph/Chart pop■ menu appears.
2. Select Edit from the pop■ menu. The Format Graph/Chart Dialog box appears with the Type Tab selected.
3. Edit the changes you want to make to your graph/chart using the tabs in the Format Graph/Chart dialog box.
4. Click OK to apply the changes to your graph/chart, and return to the report.

NOTE: If you are working in the Preview Window and have selected "Hide when printing" from the Graphic format dialog box, you will not see changes applied to the graph/chart when you have finished editing

To Select a Graph/Chart

If the resizing handles are not visible on the border of the graph/chart, Click once anywhere inside the graph/chart border. When the resizing handles appear, the graph/chart is selected.

Database Field command (Insert menu)

Use Insert|Database Field to place data fields from an active database on your report.

Large Btrieve files

Btrieve .ddf files are different than other files used with Crystal Reports in that each .ddf file may contain multiple database files. Large Btrieve files may contain, ten, twenty, or more database files. If the Insert Database Field dialog box were to list all fields in all files as they do with other database files (.dbf, .db, etc.), you could end up with hard to

- manage scroll lists in each of these dialog boxes, each listing hundreds and hundreds of fields. Because of this, Crystal Reports initially lists only the file names
- not the individual field names
- for any .ddf file that includes more than four database files. From the list of files you select and open the files of interest. The program then lists the fields for the selected files only, giving you an abbreviated scroll list that includes only those fields that you might want to include in your report.

To use the Insert Fields dialog box with large Btrieve files

1. Click the file of interest and Click the Open button (or Double Click the file of interest). Crystal Reports opens and lists the fields in the selected database.
2. Repeat the process for each additional database you want to open from the .ddf file.
3. Select and place fields from the open database(s) as you would from any other database.

NOTE: To close an open database, Click the database you want to close and Click the Close button. (When you Click a closed database, the left button changes to Open. When you Click an open database, the left button changes to Close.)

Text Field command (Insert menu)

A text field is simply a field that holds text. It can hold a single character, a single word, entire sentences, or full paragraphs.

Use the Insert|Text Field command to insert text anywhere on your report. This is a useful command for adding labels, titles, footnotes, disclaimers, explanatory statements, seasonal comments, or any other kind of text you need.

NOTE: While the effective limit on text field size is 32K (including carriage return and line feed characters), it is recommended that text fields be used only for more manageable sized blocks of text.

Text Field considerations

A text field is treated just like any other field. That is, you can:

- move it,
- delete it,
- suppress it if duplicated,
- hide it when printing,
- align the text within the field, or
- change the font, font style, font size or font effects.

Like any string field, you can also specify the number of lines on which the field is to print using the Format String dialog box.

A text field is different than freeform text entered directly on your report:

- it uses the same grid that other fields use (making it easy to align titles perfectly with columns of data),
- it prints starting at an absolute location on the report while freeform text always prints relative to the left edge of the report and can move around depending on what spaces and tabs are ahead of the fields,
- it can be moved using a mouse (while freeform text must be moved using keyboard commands), and
- it can be formatted (suppressed, hidden, aligned) while freeform text cannot.

NOTE: You cannot copy text fields via the Windows Copy command, but you can copy freeform text using that command. That is about the only thing that you cannot do with text fields that you can do with freeform text.

Word wrap and return characters

Crystal Reports will word wrap within the space allotted if you have activated the *Print on multiple lines* option in the Format String dialog box. If you expand or narrow the field, word wrap will adjust to the available space (where possible), again, if the Print on multiple lines option is activated.

NOTE: If your text field includes return characters, Crystal Reports interprets them to provide the line breaks you specify in the text field (where possible).

How to insert text fields

To insert a text field:

1. Select Insert|Text Field. The Edit Text Field dialog box appears.
2. Enter your text in the Enter Text edit box, and Click Accept when finished. A field placement box appears.
3. Position the box where you want the text field to appear and Click the left mouse button to place it.

NOTE: Insert|Text Field is also available as a button on the Button Bar. ***Clicking the button***

takes you directly to the Edit Text Field dialog box. Once there, follow the remaining steps described above.

Formula Field command (Insert menu)

Use Insert Formula Field to insert a formula to create a calculated data field.

See Also

Formulas ▪ an overview

Index To Formula Topics

Special Field command (Insert menu)

Use Special Field to insert any of four special fields on your report. When you select Special Field, a sub-menu appears listing your special field options. Those options are:

Page Number Field

Record Number Field

Group Number Field

Print Date Field

- Click on one of the above for more information.

Page Number Field command (Insert|Special Field)

Use Insert|Special Field|Page Number Field to insert a field that prints the current page number.

See Also

Set Print Date command

Record Number Field command (Insert|Special Field)

Use Insert|Special Field|Record Number Field to have Crystal Reports number each record printed in the Details section of your report.

Group Number Field command (Insert|Special Field)

Use the Special Field|Group Number Field command on the Insert menu to have Crystal Reports number each group in your report. (A group is any grouping of data that occurs when you insert a subtotal or summary.)

Print Date Field command (Insert|Special Field)

Use Insert|Special Field|Print Date Field to insert a field that prints whatever is the current date when the report prints.

Subtotal (Insert menu)

Use **Insert Subtotal** to insert a subtotal in your report. Crystal Reports allows you to set the conditions under which a subtotal will print.

Boolean Conditions

When you subtotal based on changes in a Boolean field, Crystal Reports gives you a selection of Boolean conditions that can trigger a subtotal. The program first sorts your data. Then it breaks the data into groups and prints a subtotal whenever the condition you select is met.

NOTE: In the following conditions, Yes means Yes, True or 1 (depending on the Boolean format you have selected for the field) and No means No, False, or 0.

When you select a Boolean field condition from the Insert Summary, Insert Group Section, or Insert Subtotal dialog box, your options are:

- *on change to yes or no* prints a subtotal whenever the value changes from Yes to No or from No to Yes.
- *on change to Yes* prints a subtotal whenever the value changes from No to Yes.
- *on change to No* prints a subtotal whenever the value changes from Yes to No.
- *on every Yes* prints a subtotal every time the value is Yes.
- *on every No* prints a subtotal every time the value is No.
- *on next is Yes* prints a subtotal whenever the next value is a Yes.
- *on next is No* prints a subtotal whenever the next value is a No.

Click OK when finished. Crystal Reports inserts the subtotal in the group section of your report. If you have already set up other subtotals on the same field, Crystal Reports creates a new section for the new subtotal.

Grand Total command (Insert menu)

Use Insert Grand Total to print a report total (or a report average, a report count, etc.) at the end of the report.

Summary command (Insert menu)

Use the Insert|Summary command to summarize data and print the summary in your report. You can use summaries:

- to count the number of values in a group,
- to calculate the sum, average, standard deviation, or variance value in a group, and
- to identify the minimum or maximum value in a group.

This command is active only after you have selected a field to summarize.

See Also

Boolean field conditions.

Group Section command (Insert menu)

Use Insert|Group Section to sort your data and break it into groups *without* creating a group value. This is a useful command, for example, for dividing a customer list into state or ZIP code groups. Crystal Reports allows you to set the conditions under which your data is grouped.

See Also

Boolean field conditions.

Line command (Insert menu)

Use Insert|Line to draw horizontal and vertical lines anywhere on your report. You can use these lines to enclose field data or to create other graphic effects.

While you can create lines via the Format Border and Colors command, the lines you draw via the Insert|Line command differ in several important ways:

- border lines cannot stand alone (they are always attached to a field or a picture) whereas graphic lines can stand alone,
- a border line can be as high as the line, as high as the data, as wide as the field or as wide as the data while a graphic line can be any length or height you wish, and
- graphic lines don't increase the height of a text line (making for fewer lines per inch) whereas lines created as borders do, and
- a border (field border) cannot span multiple sections whereas a graphic line can.

Box command (Insert menu)

Use Insert|Box to draw freeform boxes anywhere on your report. You can use these boxes to enclose field data or to create other graphic effects.

Boxes differ from borders in several important ways:

- borders come in fixed sizes and shapes whereas boxes can be created in any size or shape you need,
- a border can be used to enclose individual field values whereas a box, if it is placed around a field, encloses all the values in the field,
- borders cannot stand alone (they must surround a field or a picture) whereas boxes can,
- boxes don't increase the height of a text line (making for fewer lines per inch) whereas borders do, and
- a border (field border) cannot span multiple sections whereas a box can.

See Also

Format Border and Colors command.

Picture command (Insert menu)

Use Insert|Picture to insert bitmapped artwork (graphics, logos, etc.) in your report. Crystal Reports enables you to enhance the visual impact of your report using artwork in the following popular formats:

.bmp

The Windows bitmap format. Windows .bmp pictures can be generated by a number of popular programs, and they are available through commercial and public domain sources as well. Since this file format offers a great deal of compatibility between programs, it is suggested that, given a choice, you use pictures in this format.

.gif

The CompuServe format. Many free or inexpensive pictures can be downloaded in this format from CompuServe.

.pcx

The PC Paintbrush format. You can create your own pictures in PC Paintbrush for use in your reports, or you can use other pictures in compatible .pcx format.

.tif

The TIFF format. This is a popular format for scanned pictures. You can scan your logo or other picture into this format for use in your report, or you can make use of the wide variety of .tif pictures available commercially and in the public domain.

.tga

The TARGA format. This is a popular format used with many scanners and with video capture devices.

Object command (Insert menu)

The Insert|Object command allows you to embed or link an OLE object in your report.

Crystal Reports is an OLE container application. As such, it can receive objects created in any object application that supports OLE.

Report Style Expert (Format menu)

When you Select Report Style Expert from the Format menu, the Report Style Expert dialog box appears. You use this dialog box to select a report style and apply it to your report.

When you select a report style, the preview box on the right side of the dialog box changes to show you what the report style will look like.

Simply select the report style and click OK. You can change the report style at any time.

Font command (Format menu)

Use the Font command to change the font, font size, style, effects, and/or color for selected elements on your report from the Font dialog box.

Field command (Format menu)

Use the Field command on the Format menu:

- to suppress printing of duplicate data,
- to hide/unhide a field when printing,
- to set fields to conform to Windows' Default Format,
- to change field alignment, and
- to specify number, currency, Boolean, date, string, and memo field formatting.

Choose the dialog box you want from the following list to review those options:

Format String dialog box

Format Memo dialog box

Format Number dialog box

Format Currency dialog box

Format Date dialog box

Format Boolean dialog box

Border and Colors command (Format menu)

You can use Border and Colors to set up borders, background fill, and drop shadows for fields on your report. You can also use it to customize the border for a selected field. For example:

- you can enclose the field in a box,
- you can highlight the field with any portion of a box you want (only the top, only the left side, the top and bottom together with no sides, etc.),
- you can add drop shadows that print below and to the right of the field value,
- you can change the color of the border, text, and/or add a background fill color to a field, and
- you can specify the border width (the full width allotted for the field or the width of the data only), and the border height (the full height allotted for the field or the height of the data only).

Using the border options, you can create a variety of striking effects that can enhance the look of your report and highlight important data.

Change Line Height (Format menu)

Line Height and Descent are measurements describing how text is placed between two lines

By adjusting the Line Height and Descent, you will be able to position text on a pre-printed form in a manner that prevents the top of the text from touching a line, and prevents the bottom of the text from touching a line.

Set Line Height and Descent requires you to enter the desired Line Height in an edit box, and the desired Descent in an edit box.

The instructions that follow use the right mouse button to activate the command. Alternatively you can select Change Line Height from the Format menu.

Line command (Format menu)

Use the Line command on the Format menu to change the style, width, and color of a graphic line.

Box command (Format menu)

Use Box command on the Format menu to change the border and fill attributes for a box. Using this command, you can specify the fill color (if any) for the box, and the style, width, and color of the box border as well (if you want a border on the box).

NOTE: You can create an empty box with a border, a filled box with a border, or a filled box with no border. The border and fill colors are set independently so they can be different from one another.

Picture command (Format menu)

Use the Picture command on the Format menu to size, scale, crop, and fine tune the placement of a picture.

Section command (Format menu)

The Section command on the Format menu allows you to perform a number of functions that affect the formatting of an entire section of your report. This option allows you to:

- hide a section (keep it from printing),
- print subtotals or group values only at the bottom of the page,
- insert a page break before the section is printed,
- insert a page break after the section is printed,
- reset the page number to one (1) after a group value prints,
- prevent page breaks from spreading data from a single record over two pages, and
- prevent blank lines from printing,
- set up multi
- column, "telephone book" style reports, and
- format mailing labels and label type items.

Remove from Report command (Database menu)

Use Remove from Report to delete databases from the active list so they can no longer be used in your report.

Set Location command (Database menu)

Use the Set Location command to change the location of a database that is active in a report. This option is convenient if you need to change the directory or disk location of a database to avoid file name conflicts, better utilize disk space, etc. It is also a handy option to use if someone sends you a report based on databases that were located in different disk/directory locations on their system than they are on yours.

NOTE: This option does not physically move the database(s). It simply directs Crystal Reports to look for the database(s) in a different location than you originally specified when setting up the report.

Set Alias command (Database menu)

In Crystal Reports, an alias is an alternative name assigned to a database. If a database is called customer.db, you could assign the alias *customer*, *cust*, *company*, *DB1*, or any other name that suits your needs. Aliases make it easier for you to use a report created with a database whose name and/or location has changed since the report was created.

Use Set Alias to change the alias you are using for one or more of your active databases.

See Also

Selecting an alias

Set Location command

Verify Database command (Database menu)

Use Verify Database to make certain your report prints 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 that time. It uses the structure of the database (number of fields, field position, data type, etc.) to identify and select those fields you want to appear on the report. If you change the structure of the database (by adding or deleting fields) after you create the report, the program needs to adapt the report to the new structure. The Verify Database command is the tool you use for adapting the report to the new database structure.

- If the current version of the database has more fields than it had when the report was first created, Crystal Reports attempts to identify and use the correct fields from the new database. The aim is to print an unchanged version of the report even though the underlying database has changed.
- If the current version of the database has fewer fields than it had when the report was first created, it uses those fields that are still available when it prints the report and ignores those that are no longer available.

Verify on Every Print command (Database menu)

The Database|Verify on Every Print command is a lock that triggers the Verify Database command every time you print.

If there is a check mark beside the command on the Database menu, the command is active (it will trigger the Verify Database command every time you print).

- If there is no check mark beside it, the command is inactive.
- By default the command is inactive.

Stored Procedure Parameters (Database menu)

Crystal Reports enables you to build reports using stored procedures from those SQL databases that support them. The program assumes that the stored procedure you use for creating a report will generate a result set, and it makes that result set available as a table. If the stored procedure requires parameters, those parameters are stored with the report.

You use this command if you want to change the parameters for the stored procedure you are using in your report.

See Also

Using Stored Procedures

Using Stored Procedures

Crystal Reports enables you to build reports using stored procedures from those SQL databases that support them. The program assumes that the stored procedure you use for creating a report will generate a result set, and it makes that result set available as a table. If the stored procedure requires parameters, those parameters are stored with the report.

See Also

Stored Procedure Parameters

Log On Server command (Database menu)

Use Database|Log On Server to log onto a SQL database server.

Log Off Server command (Database menu)

Use the Log Off Server command to log off of an active SQL database.

Show SQL Query command (Database menu)

Use the Show SQL Query command to view and to edit the SQL query Crystal Reports is sending to your SQL server.

Visual Linking with Crystal Reports

(A Browsable Topic)

You link databases so records from one database will match related records from another. For example, if you activate a Movie database and a Studio database, you link the databases so that each movie (from the Movies database) can be matched up with the studio that made the movie. (from the Studio database)

When we link, we use a field that is common to both databases. The program uses the link to match up records from one database with those from the other. In our example, the link assures that the data in each row of the report refers to the same movie.

Use Visual Linking to link two or more databases together. When you select Database|Add Database to Report from the menu, and select an additional database, The Visual Linking Expert appears and displays the additional database.

See Also

Visual Linking Topical Index.

- ## Visual Linking Topics

Click on the topic of your choice to find out more about visual linking:

Add Database to Report

Visual Linking Expert (Database menu)

Visual Linking Expert dialog box

Choose Tables To Use In Visual Linking dialog box

Link Options dialog box

Allow Partial Text Matches

When Linking To Two Files From This File

Finding existing index for lookup database

SQL Join Types

How to do a self join using Crystal Reports

Outer Joins

A to B, A to C Reports

Add Database to Report command (Database menu)

Use the Add Database to Report command on the Database menu to select an additional database for use in your report. (When creating a new report, you select the first database you want to use in your report from the Choose Database File dialog box that appears when you begin a new report.)

See Also

Visual Linking Topical Index

Visual Linking Expert (Database menu)

Use the dialog boxes included in the **Visual Linking Expert:**

- to create Smart Links automatically between databases (in many cases)
- to display the links that have been set up among active databases,
- to create new links between active databases,
- to update (modify) existing links,
- to delete existing links
- to add new databases/tables to the report
- to select the index to use when multiple indexes are available for a field
- to set the way partial text matches are to be handled, and
- to specify lookup options when on database (A) is linked to two others (B) and (C).

See Also

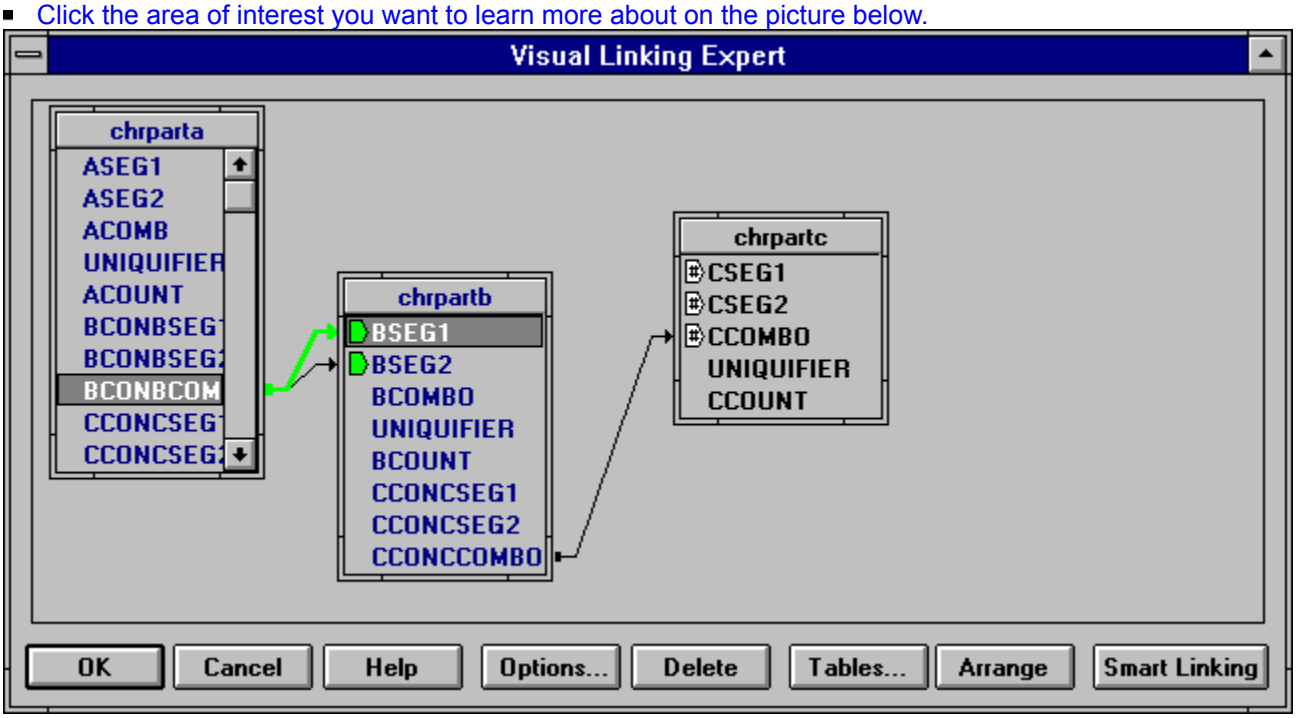
Visual Linking Topical Index

Visual Linking Expert dialog box

The Visual Linking Expert dialog box appears when you select the Visual Linking Expert command from the Database menu or when you click the Links button on the button bar.

Different dialog boxes appear when you Click the buttons at the bottom of the Visual Linking Expert dialog box.

- Click the area of interest you want to learn more about on the picture below.



See Also

Visual Linking Topical Index

Options button

The Options button takes you to the Link Options dialog box that you can use to:

- Identify and modify the index used for a specific link
- To fine tune the way the program performs lookups on linked databases
- In the case of SQL databases, to specify the kind of join made between tables.

The button becomes active whenever you select the line between linked fields.

The Smart Link button

Click the Smart Link button to have Crystal Reports perform Smart Linking on your selected databases. Smart Linking identifies the link fields automatically if possible and creates a link line between linked databases in the dialog box.

See Also

Visual Linking Topical Index

Selecting the Link Line in Visual Linking

To select the link line in Visual Linking, position the tip of the arrow cursor on the line that represents the link between tables, and Click once. When the line is selected, the black line becomes a color other than black, depending upon the indexes in use, and the black text in the tables becomes blue text in the tables. This indicates that the link is selected.

See Also

Visual Linking Topical Index

Delete button

The Delete button allows you to delete an existing link in the Visual Linking Expert dialog box. To delete a link, position the tip of the mouse pointer on the line that represents the link you want to delete and Click once to select the link. When the link is selected, the line and the names of the linked fields change color. Click the Delete button to delete the link.

Choose Tables To Use In Visual Linking dialog box

The Tables button provides you with a list of linked and unlinked tables. When you click the Tables button, the Choose Tables To Use In Visual Linking dialog box appears. The dialog box has two sections:

- Linked Tables, and
- Unlinked Tables

Add Data File button

You use the Add Data File Button, to add another database to your report. When you click the Add Data file button, the Choose File dialog box appears. You use this dialog box to select the database file or files you want to add to your report. Each time you select a file and Click the Add button, the database file is added to the Visible Tables box unless there are more tables in the data file to practically use in a report. In this case the tables are added to the Invisible Tables box.

When you have added the database or databases you need, click Done to return to the Choose Tables To Use In Visual Linking dialog box. The database or databases you added will be in the Visible Tables box.

Add SQL/ODBC button

The Add SQL/ODBC button lets you add a database from a SQL or ODBC source. When you Click the Add SQL/ODBC button, The Log On Server dialog box appears. Use this box to Log onto an SQL/ODBC server and select a database for use in your report.

Perform Smart Linking check box

The Perform Smart Linking Check box is a toggle. The check box is located in the bottom left corner of the dialog box.. When the checkmark is visible, Smart Linking is Activated. When the checkmark is not visible, Smart Linking is inactive.

When Smart Linking is turned on, and you click OK in the Choose Tables To Use In Visual Linking dialog box, Smart Linking identifies suggested link fields automatically and performs the links. You can verify the links performed by Smart Linking in the Visual Linking Expert dialog box.

See Also

Visual Linking Topical Index

Add Index Button

For non-SQL data sources, the Index In Use drop down box includes the following indexes:

- in the case of Paradox and Btrieve databases, it includes all of the available index files
- in the case of dBASE, it identifies all of the index files that have the same filename as the database (for example, for company.dbf, it will identify the index file company.mdx or company.ndx)
- If you are using dBASE and you have indexes that don't match the database name, those indexes won't appear in the Index In Use drop down box automatically. You can add those indexes to the list using the Add Index button.

See Also

Visual Linking Topical Index

Table Description dialog box

The table description dialog box is primarily a read-only dialog that provides information on any table displayed in either the Visual Linking dialog box or the Choose Tables To Use In Visual Linking dialog box.

To call up the dialog box from the Visual Linking dialog box, double-click on any field in the table of interest or Click the field using the right mouse button and then select Description from the pop-up menu that appears..

To call up the dialog box from the Choose Tables To Use In Visual Linking dialog box, select the table of interest and Click the Table Description button.

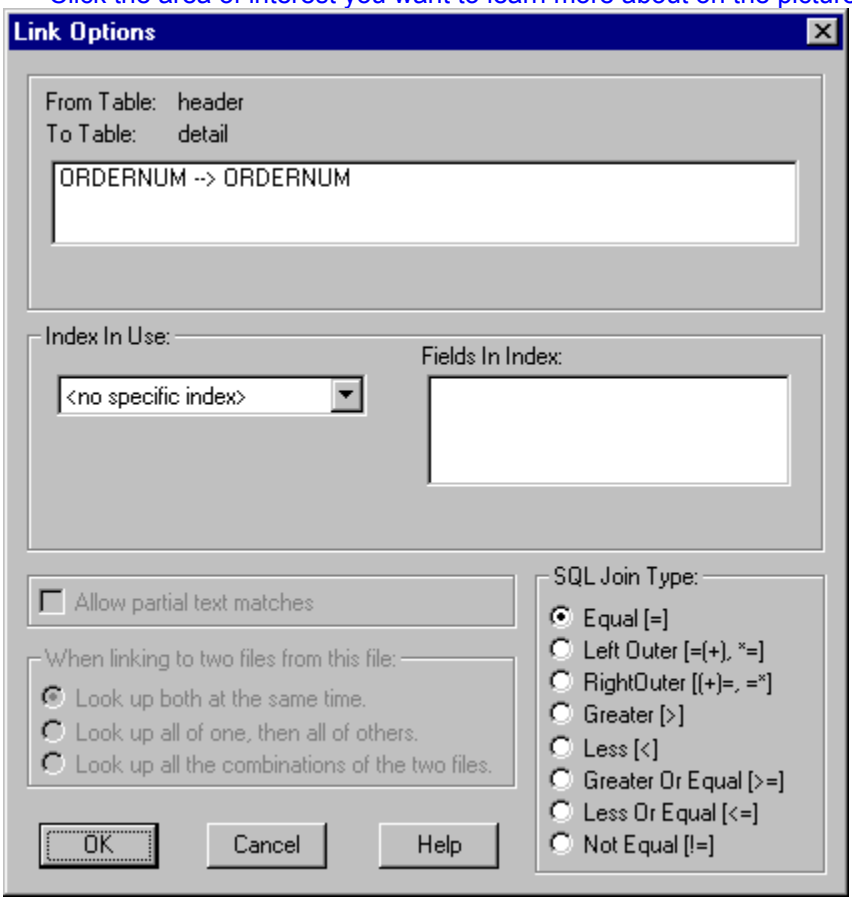
See Also

Visual Linking Topical Index

The Arrange Button

When you have several tables in the Visual Linking Expert dialog box and you click the Arrange button, Crystal Reports brings as many of the tables into view as possible.

- **Link Options dialog box**
- Click the area of interest you want to learn more about on the picture below.



See Also

Visual Linking Topical Index

Link Description box

The link description box describes the selected link. The field(s) listed on the left are the "link from" field(s) and the field(s) listed on the right are the "link to" field(s).

Allow partial text matches

When you create any link, the Link Options dialog box displays the Allow Partial Text Matches check box. This check box is for Paradox and dBASE users. Using it, you specify whether the program should have partial text matching enabled when linking. To understand this option, assume the following data:

File A	**File B**
John	Jonathan
Mary	John
Frank	Frank
Chris	Chris
Gregory	Christopher
	Greg

Partial linking On

If partial linking is On (a check in the check box) when you link File A to File B, the following records will report:

File A	**File B**
John	Jonathan
John	John
Mary	
Frank	Frank
Chris	Chris
Chris	Christopher
Gregory	

- In this case, the program links if it finds an exact text match (John with John, Chris with Chris)
- It also links if it finds a partial text match, a value in File B that begins with a value from File A. Thus John links with Jonathan because Jonathan begins with a value from File A, John.

NOTE: Gregory does not link with Greg because Greg (the File B value) does not begin with Gregory (the File A value). If Greg was in File A and Gregory was in File B, however, there would be a link.

Partial linking Off

If partial linking is Off when you link File A to File B, the following records will report:

File A	File B
John	John
Mary	
Frank	Frank
Chris	Chris
Gregory	

In this case, the program links only if it finds an exact match. John will link to John (an exact match) but John won't link to Jonathan (not an exact match).

See Also

Visual Linking Topical Index

When Linking To Two Files From This File

The Link Options dialog box displays three different options for looking up records:

- Look up both files at the same time.
- Look up all of one file, then all of the other.
- Look up all combinations of the two files.

Examples of each of these options follows:

Sample Data

Examples of the three different options for looking up records will be based on the following data:

- Database A
- Customers

CustNum	Name
1	Jones
2	Smith
3	Miller

- Database B
- Orders

OrderNum	CustNum	Amt1
11	1	10.00
22	1	20.00
33	2	30.00
44	2	40.00
55	3	30.00
66	3	30.00

- Database C
- Credits

CreditNum	CustNum	Amt2
C1	1	10.00
C2	2	30.00
C3	2	40.00
C4	3	30.00

Option 1

Look up both files at the same time.

- For each record in A, this option looks for a matching record in B and a matching record in C, then it looks for the next matching record in B and the next matching record in C, etc. Once it finds all the matching records, it repeats the process with the next record in A, then the next, etc.
- Using the example databases, Crystal Reports presents the data in this manner using this function:

CustNum	Order Num	Amt1	CreditNum	Amt2
1	11	10.00	C1	10.00
1	22	20.00	C1	10.00

2	33	30.00	C2	30.00
2	44	40.00	C3	40.00
3	55	30.00	C4	30.00
3	66	30.00	C4	30.00

Option 2

Look up all of one file, then all of the other.

- For each record in A, this option looks for all the matching records in B and then all the matching records in C, then it repeats the process with the next record in A, then the next, etc.
- Using the example databases, Crystal Reports presents the data in this manner using this function:

CustNum	Order Num	Amt1	Credit Num	Amt2
1	11	10.00		
1	22	20.00		
1			C1	10.00
2	33	30.00		
2	44	40.00		
2			C2	30.00
2			C3	40.00
3	55	30.00		
3	66	30.00		
3			C4	30.00

NOTE: If you want the C data to appear in your report before the B data, you will need to change your links so the A to C link comes first, then the A to B link. You do this via the Links dialog box. To change the order of the links, delete the existing links and set up new links in the order you want.

NOTE: For a step-by-step discussion of building a report using Option 2, see A to B, A to C reports.

Option 3

Look *up* all combinations of the two files.

- For each record in A, this option looks for a matching record in B, then it finds all the matching records in C. Once it finds all the matching records in C, it repeats the process with the next record in B, then the next, etc. When it finds matching C records for all the B records that match the first A record, it moves to the next A record and repeats the process.
- Using the example databases, Crystal Reports presents the data in this manner using this function:

Cust Num	Order Num	Amt1	Credit Num	Amt2
1	11	10.00	C1	10.00
1	22	20.00	C1	10.00
2	33	30.00	C2	30.00
2	33	30.00	C3	40.00
2	44	40.00	C2	30.00

2	44	40.00	C3	40.00
3	55	30.00	C4	30.00
3	66	30.00	C4	30.00

NOTE: If you want the program to look up the first matching record in the C database, then find all matching records in the B database (the reverse of the current process), you will need to change your links so the A to C link comes first, then the A to B link. You do this via the Visual Linking Expert dialog box. To change the order of the links, delete the existing links and set up new links in the order you want.

From Table: To Table

This section of the Link Options dialog box describes the tables that are currently selected.

- *From Table* is the origin table
- *To Table* is the destination table

Fields In Index box

The Fields In Index box is used in conjunction with the Index In Use edit box. When you have specified an index in the Index In Use edit box, the fields associated with the chosen index will appear in the Fields In Index box.

Index In Use Edit Box

The Index In Use edit box allows you to specify the index used for the database link or links. When you select an index for linking, the fields associated with the chosen index appear in the Fields In Index box, located to the right of the Index In Use edit box.

See Also

Finding existing index for lookup database

Finding existing index for lookup database

You may experience a situation when linking databases where the To File database (the lookup database) shows no index even though you know it is indexed. Specifically, the word None appears in the Index In Use scroll box in the Link Options dialog box, and if you Click the scroll arrow on the scroll box no indexes appear. This can happen when you are adding a second (or third, fourth, etc.) database, creating new links between existing databases, or updating an existing link between databases.

If you try to proceed without an index for the lookup database, you get a message stating:

```
An index must be selected for this file link.
```

What causes the problem:

This problem can occur under two sets of circumstances:

- if the index for the lookup file has a different production name (the name of the file without the extension) than the database file (for example, COMPANY.DB indexed on CONAME.PX), or
- if the lookup file (the To File) is a .DBF file and the index for that file is stored in a directory other than the directory in which the .DBF file is stored.

How to solve this problem

If the index file has a different production name, rename either the database file or the index file so that both production names are consistent, or re-index the file and save the index under the same production name as the database file. You can rename the file by using the DOS Rename command, or by using the appropriate command in the database program.

If the index for the .DBF lookup file is stored in a different directory than the .DBF file, move the index to the directory in which the .DBF file is stored. You can move the file using either the DOS Copy or Move command.

See Also

The Add Index topic.

Visual Linking Topical Index

SQL Join Types

Crystal Reports enables you to specify the type of join you want it to use when it joins SQL tables. The SQL Join Type options are located in the bottom right corner of the Link Options dialog box.

The join types are as follow:

- Equal join
- Left outer join
- Right outer join
- Greater join
- Less join
- Greater or Equal join
- Less or Equal join
- Not Equal join

Each of these joins is discussed below, complete with SQL syntax and example result sets.

Sample Data for Equal, Left outer, and Right outer joins

The example result sets for Equal joins, Left outer joins, and Right outer joins are all derived from joining the following tables:

Men

NAME	COUNTRY
Bob	USA
Ralph	Belgium
Mark	UK
Bill	USA
Trevor	UK
Sam	Germany

Women

NAME	COUNTRY
Mary	USA
Sue	UK
Barb	USA
Alice	Belgium
Linda	Australia
Ann	USA

Result sets for Equal joins, Left outer joins, and Right outer joins

Equal join

The result set from an equal join includes all the records where the link field value in both tables is an exact match. It uses the following syntax:

```
SELECT *
FROM MEN, WOMEN
WHERE MEN.COUNTRY = WOMEN.COUNTRY
```

to produce the following results:

MEN.COUNTRY		WOMEN.COUNTRY	
Bob	USA	Mary	USA
Bob	USA	Barb	USA
Bob	USA	Ann	USA
Ralph	Belgium	Alice	Belgium
Mark	UK	Sue	UK
Bill	USA	Mary	USA
Bill	USA	Barb	USA
Bill	USA	Ann	USA
Trevor	UK	Sue	UK

Left Outer Join

The result set from a left outer join includes all the records where the link field value in both tables is an exact match. It also includes a row for every record in the first table whose link field value has no match in the second table. It uses the following syntax:

```
SELECT *
FROM MEN, WOMEN
WHERE MEN.COUNTRY *= WOMEN.COUNTRY
```

to produce the following results:

MEN.COUNTRY		WOMEN.COUNTRY	
Bob	USA	Mary	USA
Bob	USA	Barb	USA
Bob	USA	Ann	USA
Ralph	Belgium	Alice	Belgium
Mark	UK	Sue	UK
Bill	USA	Mary	USA
Bill	USA	Barb	USA
Bill	USA	Ann	USA
Trevor	UK	Sue	UK
Sam	Germany	Null	Null

Right Outer Join

The result set from a right outer join includes all the records where the link field value in both tables is an exact match. It also includes a row for every record in the second table whose link field value has no match in the first table. It uses the following syntax:

```
SELECT *
FROM MEN, WOMEN
WHERE MEN.COUNTRY =* WOMEN.COUNTRY
```

to produce the following results:

MEN.COUNTRY		WOMEN.COUNTRY	
Bob	USA	Mary	USA

Bob	USA	Barb	USA
Bob	USA	Ann	USA
Ralph	Belgium	Alice	Belgium
Mark	UK	Sue	UK
Bill	USA	Mary	USA
Bill	USA	Barb	USA
Bill	USA	Ann	USA
Trevor	UK	Sue	UK
Null	Null	Linda	Australia

Sample Data for Greater, Less, Greater or Equal, Less or Equal, and Not Equal joins

The example result sets for Greater, Less, Greater or Equal, Less or Equal, and Not Equal joins are all derived from joining the following tables:

MEN1

NAME	AMOUNT
Bill	10
Randy	5
Tom	15
Sam	12
Ted	10
Mark	6

WOMEN1

NAME	AMOUNT
Sally	5
Mary	2
Becky	12
Erica	10
Linda	16
Barb	10

Result sets for Greater joins, Less joins, Greater or Equal joins, Less or Equal joins, and Not Equal joins

Greater

The result set from a Greater join includes all records in which the link field value from the first table is greater than the link field value in the second table. It uses the following syntax:

```
SELECT *
FROM MEN, WOMEN
WHERE MEN.AMOUNT > WOMEN.AMOUNT
```

to produce the following results:

MEN.AMOUNT		WOMEN.AMOUNT	
Bill	10	Sally	5
Bill	10	Mary	2
Randy	5	Mary	2
Tom	15	Sally	5
Tom	15	Mary	2
Tom	15	Becky	12
Tom	15	Erica	10
Tom	15	Barb	10
Sam	12	Sally	5
Sam	12	Mary	2
Sam	12	Erica	10
Sam	12	Barb	10
Ted	10	Sally	5
Ted	10	Mary	2
Mark	6	Sally	5
Mark	6	Mary	2

Less

The result set from a Less join includes all records in which the link field value from the first table is less than the link field value in the second table. It uses the following syntax:

```
SELECT *
FROM MEN, WOMEN
WHERE MEN.AMOUNT < WOMEN.AMOUNT
```

to produce the following results:

MEN.AMOUNT		WOMEN.AMOUNT	
Bill	10	Becky	12
Bill	10	Linda	16
Randy	5	Becky	12
Randy	5	Erica	10
Randy	5	Linda	16
Randy	5	Barb	10
Tom	15	Linda	16
Sam	12	Linda	16
Ted	10	Becky	12
Ted	10	Linda	16
Mark	6	Becky	12
Mark	6	Erica	10

Mark	6	Linda	16
Mark7	6	Barb	10

Greater or Equal

The result set from a Greater or equal join includes all records in which the link field value from the first table is greater than or equal to the link field value in the second table. It uses the following syntax:

```
SELECT *
FROM MEN, WOMEN
WHERE MEN.AMOUNT >= WOMEN.AMOUNT
```

to produce the following results:

MEN.AMOUNT		WOMEN.AMOUNT	
Bill	10	Sally	5
Bill	10	Mary	2
Bill	10	Erica	10
Bill	10	Barb	10
Randy	5	Sally	5
Randy	5	Mary	2
Tom	15	Sally	5
Tom	15	Mary	2
Tom	15	Becky	12
Tom	15	Erica	10
Tom	15	Barb	10
Sam	12	Sally	5
Sam	12	Mary	2
Sam	12	Becky	12
Sam	12	Erica	10
Sam	12	Barb	10
Ted	10	Sally	5
Ted	10	Mary	2
Ted	10	Erica	10
Ted	10	Barb	10
Mark	6	Sally	5
Mark	6	Mary	2

Less or equal

The result set from a Less or equal join includes all records in which the link field value from the first table is less than or equal to the link field value in the second table. It uses the following syntax:

```
SELECT *
FROM MEN, WOMEN
WHERE MEN.AMOUNT <= WOMEN.AMOUNT
```

to produce the following results:

MEN.AMOUNT		WOMEN.AMOUNT	
Bill	10	Becky	12
Bill	10	Erica	10
Bill	10	Linda	16
Bill	10	Barb	10
Randy	5	Sally	5
Randy	5	Becky	12
Randy	5	Erica	10
Randy	5	Linda	16
Randy	5	Barb	10
Tom	15	Linda	16
Sam	12	Becky	12
Sam	12	Linda	16
Ted	10	Becky	12
Ted	10	Erica	10
Ted	10	Linda	16
Ted	10	Barb	10
Mark	6	Becky	12
Mark	6	Erica	10
Mark	6	Linda	16
Mark	6	Barb	10

Not equal

The result set from a Not equal join includes all records in which the link field value from the first table is not equal to the link field value in the second table. It uses the following syntax:

```
SELECT *
FROM MEN, WOMEN
WHERE MEN.AMOUNT != WOMEN.AMOUNT
```

to produce the following results:

MEN.AMOUNT		WOMEN.AMOUNT	
Bill	10	Sally	5
Bill	10	Mary	2
Bill	10	Becky	12
Bill	10	Linda	16
Randy	5	Mary	2
Randy	5	Becky	12
Randy	5	Erica	10
Randy	5	Linda	16
Randy	5	Barb	10

Tom	15	Sally	5
Tom	15	Mary	2
Tom	15	Becky	12
Tom	15	Erica	10
Tom	15	Linda	16
Tom	15	Barb	10
Sam	12	Sally	5
Sam	12	Mary	2
Sam	12	Erica	10
Sam	12	Linda	16
Sam	12	Barb	10
Ted	10	Sally	5
Ted	10	Mary	2
Ted	10	Becky	12
Ted	10	Linda	16
Mark	6	Sally	5
Mark	6	Mary	2
Mark	6	Becky	12
Mark	6	Erica	10
Mark	6	Linda	16
Mark	6	Barb	10

See Also

Visual Linking Topical Index

Index considerations when linking

When you Smart Link databases using a field that is a component of multiple indexes, Crystal Reports selects one of the indexes for the link. That index may or may not be the one you want to use. To determine the index in use and to change it if you wish, you use the Index section of the Link Options dialog box. To call up that dialog box:

- double
- click the link line of interest,
- select the link line of interest and Click the Options button at the bottom of the Visual Linking Expert, or
- select the link line of interest, Click the right mouse button, and select Options from the pop
- menu that appears.

The Index section of the dialog box has two parts:

- an Index In Use drop down box that displays the index that is currently in use. If you Click the arrow, it also lists the other indexes that are available for the link as well as the option <no specific index>. If you are using dBASE and you don't see a particular index that you expect to see on the list, see the Add Index Button topic.
- a Fields in Index list box that lists the fields that are included in the index that's currently selected in the Index In Use box.

If you're not happy with the index that's currently in use, Click the arrow on the Index In Use drop down box and select your index from the others that are available.

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 Window.

See Also

Visual Linking Topical Index

How to do a self-join using Crystal Reports

There may be times in which all the information you need for a report is in a single database, but you need to evaluate the data in a way that isn't possible with a single copy of the database. For example:

- in an employee database that lists employees by name and number and the manager for each employee by number only, you may want to generate a report that shows the managers' names as well; and
- in the same employee table, the salaries are listed for each employee, and you want to determine if there are any employees making more than their managers.

These are specialized situations that call for specialized tools. A self-join is the tool you use in situations such as this.

What is a self-join?

In a self join, a database is linked to itself in such a way as to provide the appearance of two different databases. Once the link is made, you can relate data in one copy of the database to data in the other copy, just as if you were using two separate databases.

What you need to setup a self-join

In order to do a self-join:

- your database must have two fields that contain the same data or subsets of the same data, and
- the fields must be the same size and have the same data type.

You use these two fields as linking fields, one from the first copy of the database and the other from the second copy. Thus you setup a relationship between the two fields (and the other data in the databases) that you couldn't do using a single copy of the database.

Some examples will help to illustrate the self-join concept. If you want to reproduce these examples yourself using Crystal Reports, you will find the database **EmpData.db** located in the CRW directory (or the directory in which you specified that Crystal Reports files were to be placed during installation).

Example

The employee database (called EmpData.db) contains data similar to this:

EmpNum	Name	Salary	MgrNum
10032	Rudolph	115000	10032
12345	Smith	35000	13344
13344	Jenkins	80000	10032
23457	Mendell	84600	13344
24689	Bosworth	42500	44444
33333	Bohrman	53250	44444
44444	Randall	90000	10032

The following observations can be made about this data:

- Each employee has a manager, but the manager is shown by employee number only, not by name.
- The *{file.EmpNum}* field and the *{file.MgrNum}* fields are identical: they are both the same size, have the same data type, and contain the same kind of information (employee numbers).
- Employee 10032 (Rudolph) is probably the CEO since the data shows Rudolph's own employee number as that of Rudolph's manager.
- Some employees may be making more than their managers.

In looking at this data, you want to do the following two things:

1. You want to identify each employee's manager by name.

2. You want to identify those employees (if any) who are earning more than their managers.

You can do both of these using a self-join.

Setting up a self-join

1. Choose the Options command from the File menu and then select the Database tab. Set the Use Default Alias switch to Off (no check mark).
2. Select Report. The Choose Database File dialog box appears.
3. Select C:\CRW\EmpData.dbf. The Alias Name dialog box appears.
4. Type in *first* as the alias for this first copy of the EmpData database, and click OK when finished. The Report Designer appears.
5. Select Database|Add Database to Report. The Choose Database File dialog box appears again.
6. Select C:\CRW\EmpData.db again. The Alias Name dialog box appears.
7. Type in *second* as the alias for this second copy of the EmpData database, and click OK when finished. The Visual Linking Expert dialog box appears and the two databases are displayed.
8. Now click the MGRNUM field from the first database, and drag the cursor up to the EMPNUM field in the second database. The cursor changes to the drag and drop cursor while you are dragging. When the drag and drop cursor is positioned on the EMPNUM field, release the mouse button. Visual Linking completes the link, and a white line is visible indicating the link. Click OK to accept the link and return to the Report Window. If you made an error, you can Click delete while the white line is visible and the link is deleted. You can then make the link again.

To create the report using the databases

1. Choose the Database Field command from the Insert menu. The Database Field dialog box appears.
2. Select and place side by side the following fields from the active databases:

```
{first.EmpNum}
{first.Name}
{first.Salary}
{second.Name}
{second.Salary}
```

3. Print to the Preview Window and your report data should now look like this (without the column headings):

Employee#	Name	Salary	Manager	MgrSalary
10032	Rudolph	115,000	Rudolph	115,000
12345	Smith	35,000	Jenkins	80,000
13344	Jenkins	80,000	Rudolph	115,000
23457	Mendell	84,600	Jenkins	80,000
24689	Bosworth	42,500	Randall	90,000
33333	Bohrman	53,250	Randall	90,000
44444	Randall	90,000	Rudolph	115,000

The data now shows the name of each manager along with the manager's salary. All that remains is to identify those employees who are making more than their managers. Here are two ways to do that:

- You can insert a "flag" formula to flag all employees who have a higher salary than their managers.
- You can create a record selection formula that includes in the report only those employees who make more than their managers.

To insert a flag formula

1. Choose the Formula command from the Insert menu. The Insert Formula dialog box appears.
2. Type in the name you want to use to identify the formula. The Formula Editor appears.
3. Enter the following formula:

```
If {first.Salary} > {second.Salary} Then
"Emp$>Mgr$"

Else
""
```

« This formula tells Crystal Reports to see if the value in *{first.Salary}* (the employee's salary) is greater than the value in *{second.Salary}* (the manager's salary). If it is, print the text string "Emp$>Mgr$". If it isn't greater (that is, if the employee's salary is less than the manager's salary), print nothing (as signified by the empty string ""). »

4. Click Accept when done, and when the field placement box appears, position the box to the right of the *{second.Salary}* field and Click the left mouse button to place it.
5. Print to the Print Preview Window to check your data.

Outer joins

In working with multiple databases, you may find yourself setting up a one-to-many link in which some records in the master file find no match in the detail file. In this situation, Crystal Reports automatically does an outer join, which means that it includes all records from both files, even if there isn't a valid link.

Some further explanation will help to clarify this concept.

In the databases that follow, the *Dept* database is a listing of departments in a company, and the *Emp* database is a listing of employees and the departments to which they are assigned. The following things can be said about this data:

- any department can have no employees, one employee, or several employees, that is, any record from the Dept database can be matched with zero, one, or many employee records using DepNum as the linking field (that is, using *{dept.DepNum}* to lookup matching values in *{emp.DepNum}*).
- thus linked, the Dept database is the one (the master) database
- all employees are assigned to one of the departments in the Dept database; each employee is assigned to one and only one department.
- the Emp database is the many (the detail) database
- no employees are currently assigned to the Shipping department (DepNum 03)

Dept database

DepNum	DepName
01	Manufacturing
02	Sales
03	Shipping
04	Accounting

Emp database

EmpNum	EmpName	DepNum
1007	Smith	02
1118	Jones	02
1234	Brown	01
1333	Joyce	04
1456	Winslow	01
1625	Arnold	02
1888	Sanders01	
1904	Mettler	01
1956	Samuels	04
1987	Johnson	01

An outer-join in action

If you activate both databases, link them via the DepNum field (using the values in *{dept.DepNum}* to lookup values in *{emp.DepNum}*), and print a report using the fields *{dept.DepNum}*, *{dept.DepName}*, *{emp.EmpNum}*, and *{emp.EmpName}*, you get the following data:

DepNum	DepName	EmpNum	EmpName
01	Manufacturing	1234	Brown

01	Manufacturing	1456	Winslow
01	Manufacturing	1888	Sanders
01	Manufacturing	1904	Mettler
01	Manufacturing	1987	Johnson
02	Sales	1007	Smith
02	Sales	1118	Jones
02	Sales	1625	Arnold
03	Shipping		
04	Accounting	1333	Joyce
04	Accounting	1956	Samuels

While only ten employees exist in Emp, the report lists eleven records, one record for each employee and one record for the department (Shipping) to which no employee is currently assigned. In using the value of the DepNum field in the Dept database to lookup matching records in the Emp database, the following things happen:

- For DepNum 01 it prints data from the Dept database record, and it finds and prints data from all matching records in the Emp database (5 of them)
- For DepNum 02 it prints data from the Dept database record, and it finds and prints data from all matching records in the Emp database (3 of them)
- For DepNum 04 it prints data from the Dept database record, and it finds and prints data from all matching records in the Emp database (2 of them)
- For DepNum 03 it finds no matching records so it prints data only from the 03 record in the Dept database.

NOTE: To print only those records for which there is a match (for which a department has an employee assigned to it), enter the following as the selection formula using the Report\|Edit Selection Formula\| Record command.

not IsNull(*{emp.EmpNum}*)

This prints only those records that don't have a null value in the EmpNum field.

A many-to-one link

If you were to link both databases in the opposite direction (so Crystal Reports uses the value of *{emp.DepNum}* to lookup matching values in *{dept.DepNum}*, you set up a many-to-one link. Emp becomes the master database and Dept becomes the detail database. In this case, the report will list only 10 records, one for each employee in the Emp database.

- For each value in {emp.EmpNum}, it prints data from the Emp database record, and it finds and prints data from the one record that matches it in the Dept database (data is ultimately printed from a total of 10 matches).
- Since there is no value in the Emp database that could match with the Shipping department (03), that record (Dept, 03) is ignored and doesn't print.

With a many-to-one link, your data will look like this:

DepNum	DepName	EmpNum	EmpName
01	Manufacturing	1234	Brown
01	Manufacturing	1456	Winslow
01	Manufacturing	1888	Sanders

01	Manufacturing	1904	Mettler
01	Manufacturing	1987	Johnson
02	Sales	1007	Smith
02	Sales	1118	Jones
02	Sales	1625	Arnold
04	Accounting	1333	Joyce
04	Accounting	1956	Samuels

There is no reference anywhere in the report to Dept 03, Shipping.

See Also

Visual Linking Topical Index

A to B, A to C reports

An A to B, A to C report is a multiple-database report in which the A database is linked to the B database and the A database is also linked to the C database. The report breaks down data so that for each record in the A database, matching records in the B database are printed first and matching records in the C database are printed last. *These options become available only when you are linking one database to two different databases*

The A to B, A to C format can be used for:

- order reports in which orders may contain both parts and services,
- accounting reports in which each customer (client, department, etc.) can generate both orders and credit memos,
- inventory reports in which each part number can show shipments out to customers, and shipments in from suppliers,
- and for a variety of other similar reporting needs.

Here is a typical A to B, A to C report:

Name	Order#	Order Amt.	RMA #	Credit Amt.
Carter	5	45.00		
	6	24.00		
	Order total	69.00		
Jones	1	10.00		
	2	12.00		
			CR1234	10.00
	Order total	22.00	Credit total	10.00
Smith	3	20.00		
	4	30.00		
			CR3456	23.00
			CR4567	45.00
	Order total	50.00	Credit total	68.00

In this report data is organized by customer and within each customer listing:

- all of the orders are presented together, and
- all of the credits are presented together,
- the orders for each customer are totaled, and
- the credits for each customer are totaled.

To create an A to B, A to C report

In order to create an A to B, A to C report you need to do the following:

- select and link the databases,
- select the correct lookup option,
- set up report fields and field titles,
- set up subtotals and subtotal labels,
- set up conditional subtotals and subtotal labels (optional), and
- print the report.

This section takes you step-by-step through this procedure, using the following data to create an example report:

Database A ▪ customer database (cust.dbf)

NAME	NUMBER
Jones	1
Smith	2
Carter	3

Database B ▪ orders database (orders.dbf)

ORDER	CUSTOMER	DATE	AMOUNT
1	1	10▪12▪92	10.00
2	1	10▪13▪92	12.00
3	2	10▪14▪92	20.00
4	2	10▪12▪92	30.00
5	3	10▪14▪92	45.00
6	3	10▪16▪92	24.00

Database C ▪ credits database (credits.dbf)

CRNOTE	RMA	CUST	AMOUNT
1	CR1234	1	10.00
2	CR3456	2	23.00
3	CR4567	2	45.00

These three databases (cust.dbf, orders.dbf, and credits.dbf) are installed when you install Crystal Reports sample data, and can be used for creating this example report.

NOTE: This tutorial does not take you through all the fine points of report creation (spacing, resizing fields, etc.). It is meant instead to give you general concepts for creating A to B, A to C reports. Your finished report will work correctly, but you will need to adjust the layout, formatting, and field titles to suit your needs.

Selecting and linking the databases

1. Choose the New|Custom Report command from the File menu. The Choose Database File dialog box appears.
2. Select the A database and click OK when finished.
3. In our example, that database is cust.dbf. The Design Window appears.
4. Select Database|Add Database to Report. The Choose Database File dialog box appears again.
5. Select the B database and click OK when finished.
6. In our example, that database is orders.dbf. The Visual Linking Expert appears displaying the cust and orders databases. Click the Number field in database A and drag the cursor to the Customer field in database B. When you release the mouse button, the *NUMBER* field in database A (cust) is linked with the indexed field *CUSTOMER* in database B (orders).
7. The Visual Linking Expert dialog box displays the tables and link. Click OK to return to the Design Window.
8. Repeat steps 3 through 5 to add the C database to the report.

▪ In our example, that database is credits.dbf. The Number field in database A (cust) should link with the indexed field Cust in database C (credits). Complete this by Clicking on the Number field in the A

database and dragging the cursor to the Cust

- You have now linked two lookup databases (B and C) to a primary database A.

The critical step ▪ selecting the correct link option

Even though the databases are linked correctly, there is one final and critical linking step. You must select the correct procedure for the program to use when looking up records in the B and C databases that match records in the A database. Since you want your report printing first the orders for each customer and then the credits, you will select the method that first finds all the matching records (orders) in the B database and then all the matching records (credits) in the C database. You make your selection in the Link Options dialog box

To select the correct link option

1. Select one of the links and click the Options button and the Link Options dialog box appears.
2. In the *When linking to two files from this file* box, Select the second option, *Look up all of one file, then all of the other*, and click OK when finished to return to the Visual Linking Expert dialog box.
3. Click Done in that dialog box to return to the Design Window with the Insert Database Field dialog box displayed. when you set up and then print your report, the program will first find all matching records in the orders database and then all the matching records in the credits database.

Setting up report fields and field titles

Now that the links are in place, you set up your report by entering report fields and subtotals.

To set up the report in our example, enter the following fields from left to right in the Details section of the Design Window:

```
{cust.NAME}
{orders.ORDER}
{orders.AMOUNT}
{credits.RMA}
{credits.AMOUNT}
```

Leave ample space between the fields.

NOTE: Unless you have inactivated the Insert Detail Field Titles switch in the Options dialog box, Crystal Reports will automatically insert the field titles in the Page Header section of the Design Window.

Setting up and labeling subtotals

In most A to B, A to C reports, you will have fields that always contain values, and you want to subtotal these fields for each customer (client, order, etc.) In our example, orders.AMOUNT is such a field. Since we want an order subtotal to print for every customer, we'll set it up so the program subtotals orders.AMOUNT every time cust.NAME changes. To do this:

1. Click the orders.AMOUNT field in the Design Window, and click the right mouse button to call up the pop▪menu.
2. Choose Insert Subtotal and the Insert Subtotal dialog box appears.
3. In the top scroll box, select cust.NAME to sort the report by customer name and to generate the subtotal every time the customer name changes.
4. *In ascending order*, the sort direction in the second scroll box, is fine, so leave it as it is.
5. Click OK when finished to return to the Design Window. Crystal Reports creates a group section in the Design Window and enters the subtotal in that section.
6. To label the subtotal, select Insert|Text Field. The Edit Text Field dialog box appears.
7. Enter the subtotal label, "Order total," and click Accept when finished. The field placement box

appears.

8. Position the box to the left of the subtotal value and Click the left mouse button to place it.

Setting up and labeling conditional subtotals

In some A to B, A to C reports, you will have a field that sometimes contains values and sometimes does not (credits, returns, etc.). When you have a field such as this, you may want to print a subtotal and label only when there are values in the field.

In our example, credits.AMOUNT is such a field. If the customer has credits, you want to subtotal the credits and label the subtotal, but if the customer doesn't have any credits you don't want the zero subtotal and the label to print. To accomplish this, you create a conditional formula for the subtotal and another conditional formula for the label.

The formula for the subtotal:

Use the following formula to print a credits subtotal only when there are credits for the customer:

```
If IsNull(Sum({credits.AMOUNT},{cust.NAME})) Then
0
Else
Sum ({credits.AMOUNT},{cust.NAME})
```

NOTE: If there are credits, the formula totals the credits and prints the total. If there are no credits, the formula returns a zero.

You can modify this formula easily for use with your data.

To enter the subtotal formula

1. From the Design Window, Choose the Formula Field command from the Insert menu. The Insert Formula dialog box appears.
2. Type in the name you want to use to reference the formula and click OK. The Formula Editor appears.
3. Enter the formula.
4. Click Accept when finished. The field placement box appears.
5. Position the formula field box in the GF1 section, immediately below the credits.AMOUNT field, and Click the left mouse button to place it.

Suppressing zero values

To keep the zero value from printing, activate the Suppress if Zero switch in the Format Number dialog box. To activate the switch:

1. With the formula field selected, click the right mouse button to call up the menu.
2. Select Change Format and the Format Number dialog box appears.
3. Click the Suppress if Zero check box to activate that option and click OK when finished to return to the Design Window

The formula for the label

Use the following formula to print a credits subtotal label only when there is a credits subtotal for the customer:

```
If IsNull (Sum ({credits.AMOUNT},{cust.NAME})) Then
""
Else
"Credit total"
```

NOTE: If there is a credit total, this formula prints the label "Credit total." If there is no credit total (because there are no credits), the formula prints an empty string (""), that is, it prints nothing.

You can modify this formula easily for use with your own data.

To enter the subtotal formula

1. From the Design Window, choose the Formula Field command from the Insert menu. The Insert Formula dialog box appears.
2. Type in the name you want to use to reference the formula and click OK. The Formula Editor appears.
3. Enter the formula.
4. Click Accept when finished. The field placement box appears.
5. Position the formula field box in the GF1 section, immediately to the left of the credits subtotal formula, and Click the left mouse button to place it. You may have to move some fields to make room for the new formula.

Printing the report

When finished setting up the report, print it to the print window to review your work.

Dont forget to format your report. You can change fonts, add borders, adjust spacing, and make other field formatting changes as necessary .

When satisfied with the results, print your final copy.

NOTE: If you do not use secondary indexes with your databases, the report will not contain all of your data.

Select Records Expert command (Report menu)

This command enables you to select the records or groups you want to include in your report (if you don't want them all included). For example, you may have a customer database that contains records for customers from every state and Canadian province but you want to do a report only on Texas customers. Select Records enables you to restrict your report so that only Texas customers are included.

When you use Select Records Expert, Crystal Reports automatically generates a record or group selection formula based on settings you specify in the dialog box.

NOTE: This command allows you to set up reasonably complex selection criteria, but to have more control over the formula that is generated, use the formula editor. To open the Formula Editor, click the Show Formula button. The dialog box expands and displays the Formula Editor button. Click the Formula Editor button to open the Formula Editor

Edit Selection Formula command (Report menu)

Use Edit Selection Formula to select the records or groups that you want included in your report (if you don't want them all included). When you select Edit Selection Formula from the menu, a sub■menu appears with the following two commands:

Record (Report|Edit Selection Formula)

Group (Report|Edit Selection Formula)

Click on one of the above for more information.

Record command (Report|Edit Selection Formula menu)

Use Edit Selection Formula|Record to create or edit the Record Selection Formula in your report. Use this command when you have previously specified a selection on a record or records in your report and you want to edit that selection.

Group command (Report|Edit Selection Formula menu)

The Edit Selection Formula|Group command on the Report menu enables you to edit the group selection formula Crystal Reports is currently using to select groups for your report. You use this command when you have previously specified a selection on a group or groups in your report and you want to edit that selection.

Sort Records command (Report menu)

Use Sort Records to change the order in which records appear in your report. You can add and remove sort fields and define the sort direction (ascending or descending) for the data in your report.

See Also

Sorting and grouping topics

Multiple field sorts

Refresh Report Data command (Report menu)

Use Refresh Report Data to rerun your report and display it in the Preview Window.

By design, Crystal Reports only retrieves data when necessary. When you preview your report in the Preview Window, print the report, or export it to a file, the program runs your report and retrieves the required data. Once you are working with the data in the Preview Window, the program reruns the report and retrieves the data again under the following conditions:

- you add fields to the report after the report was run,
- you add a formula that references a field that wasn't in the report when you ran it the first time,
- you expand your record selection criteria to include more records than were needed when you first ran the report, or
- you toggle "Refresh Data on every print" option on.

Aside from these limited circumstances, the program will not automatically rerun a report.

There may be times, however, when you want to retrieve new data for the report. For example, you may have created a report several days earlier and want to see it now with the most current data. Refresh Report Data is intended for times such as this.

Report Title command (Report menu)

Use the Report Title command on the Report menu to:

- change the title that appears on the title bar of your report, and
- to enter and/or edit non
- printing comments that you want to accompany the report.

Set Print Date command (Report menu)

If your report contains a Print Date Field, you can change the date that appears in this field with the Report|Set Print Date command. This allows you, for example, to post-date a report that you need to print prior to the date it will be reviewed in a meeting or conference.

Tile Vertically command (Window menu)

Use the Tile Vertically command to display your Crystal Reports windows side by side on screen. With Tile, windows may be resized so they can all fit on screen; all windows are visible.

Tile Horizontally command (Window menu)

Use the Tile Horizontally command to display your Crystal Reports windows one above the other on screen. With Tile Vertically, windows may be resized so they can all fit on screen; all windows are visible.

Cascade command (Window menu)

Use the Cascade command to stack and overlap your Crystal Reports windows. With Cascade, the entire top window is visible but only the title bars of the remaining windows are.

Arrange Icons command (Window menu)

If you have minimized any document windows, use the Arrange Icons command to arrange the document icons neatly at the bottom of the Crystal Reports window.

Close All command (Window menu)

Use the Close All to close out all open windows. Crystal Reports gives you an opportunity to save any work in each window that you haven't already saved.

When you choose the Close All command from the Window menu:

- If you have saved all your reports, Crystal Reports closes out all the windows.
- If you haven't saved all your reports, Crystal Reports prompts you to save them, report by report.
- Select *Yes* to save the report before closing the window.
- Select *No* to close the window without saving the report.

Contents command (Help menu)

Use the Contents command to call up the main Crystal Reports help index. Using this index as a starting point, you can rapidly find any help topic of interest. Once in the help system, you can always return to the main index by clicking the Contents button in the upper left corner of the help window.

Search command (Help menu)

The Search command displays the Search dialog box for Crystal Reports Help. You can search through every subject in Help to find the information you need quickly. Jump straight to the topic of interest from this dialog box.

Using Help command (Help menu)

If you are not familiar with the Windows Help system, choose the Using Help command from the Help menu. A separate window will appear explaining how to use the Help system. Follow the instructions to learn more about Windows' Help system and how to get around in Crystal Reports Help.

Register/Change Address command (Help menu)

When you select this command, the program displays an electronic form that you can use to register Crystal Reports and/or to notify the company of a change of address. You can then print this form and mail or FAX it in, or you can transmit the information to the company via modem. Instructions for sending the form by mail or FAX are printed right on the form.

NOTE: You must send in the registration form in order to receive technical support on your product.

Crystal Library command (Help menu)

Crystal Services maintains a bulletin board for technical support. Using the bulletin board you can find information on a wide range of topics including updated technical information, fixes for minor problems that users have reported, and comments and advice from Crystal support technicians. We're constantly adding new files to help keep Crystal Reports the most efficient, sophisticated, but easy to use report writer available. You can download any of the bulletin board files for your own use.

Crystal Library boosts your efficiency when using the Crystal bulletin board. It allows you to download a list of files available on the bulletin board so you can review them at your leisure, offline, and select the files of interest. When you're finished, use the Crystal Library command to call the bulletin board and download the files automatically.

About Crystal Reports command (Help menu)

Use the About Crystal Reports command to find the version number and other pertinent information about Crystal Reports.

Zoom command (Report menu)

The Zoom command allows you to switch between three levels of magnification when displaying reports in the Preview Window.

NOTE: The Zoom command is not available in the Design Window.

Set Label Layout command (File menu)

The Set Label Layout command is available on the File menu whenever you are working on mailing labels or other label-type items. The command returns you to the Mailing Labels dialog box, the dialog box you used to set your label specifications when you used the File|New|Custom Mailing Label command. Using the Mailing Labels dialog box, you can adjust the label dimensions, the gaps between labels, the page margins, and/or the label printing direction.

Send Behind Others command

Whenever you're working with stacked fields, boxes, lines, or pictures, use Send Behind Others to send the selected item to the bottom of the stack. This command gives you the ability to work with items that are buried in a stack without having to first move aside the items on top.

NOTE: An advanced reporting practice involves stacking multiple alternative formulas. This is the type of situation for which this command was designed.

Top N/Sort Group Expert command (Report menu)

The Top N/Sort Group Expert Command enables you to set up reports that print only the top N or bottom N groups, N being a number you specify.

Before you can use the Top N Expert, your data must be subtotaled or summarized.

See Also

How to identify the "top" groups

More Print Engine Error Messages check box

Using this check box tells Crystal Reports to use more verbose error messages when a print engine error occurs.

Convert Null Field Values To Default

Using this check box forces Crystal Reports to convert any null values to the database default. Some databases treat a null as a zero, some as a blank, and some as a special null value.

Select Records Expert

The Select Records Expert is a dynamic box. Different data types and selection criteria change your dialog box options.

Using the scroll box, select the condition that best finishes this sentence:

```
I want to select all records where the value
in the field I have selected is
```

You can select from any of the following conditions (depending on the data type of the field you have selected):

Primary conditions

Date conditions

Boolean conditions

Primary conditions

These conditions appear whenever you use Select Records with number, currency, string, or memo fields. By default, each condition is based on the "is" radio button. To reverse the condition, simply choose the "is not" radio button.

any value

Select records that have any value in the selected field. Use this option to include all records in the report; no records are excluded.

equal to

Select records for which the value in the selected field is equal to another value to be specified.

```
Customer is equal to Acme
```

«Include all records that show Acme as the customer.»

```
OrderDate is equal to January 15, 1992
```

«Include all records that show an order date of January 15, 1992.»

one of

Select records for which the value in the selected field is one of two or more values to be specified.

```
OrderDate is one of January 15, 1992, or January 16, 1992
```

«Include all records in which the order date is either January 15, 1992, or January 16, 1992.»

```
ZIP is one of 85201 or 85202 or 85203.
```

«Include all records in which the ZIP code is either 85201, 85202, or 85203.»

less than

Select records for which the value in the selected field is less than another value to be specified.

```
Quantity is less than 5
```

«Include all records that show a quantity smaller than five.»

```
Amount is less than 1000.00
```

«Include all records that show an amount smaller than 1000.00.»

greater than

Select records for which the value in the selected field is greater than another value to be specified.

```
Quantity is greater than 5
```

«Include all records that show a quantity bigger than 5.»

```
Amount is greater than 1000.00
```

«Include all records that show an amount bigger than 1000.00.»

between

Select records for which the value in the selected field falls between or matches one of two values to be specified.

```
OrderDate is between January 1, 1992 and March 31, 1992
```

«Include all records in which the order date falls between (or matches either) January 1, 1992 and March 31, 1992.»

```
ZIP is between 90000 and 99999
```

«Include all records in which the ZIP code falls between (or matches either) 90000 and 99999.»

starting with

Select records for which the value in the selected field begins with the value specified.

NOTE: The specified value is not included in the selection. It is only the starting point. The values included in the selection are all those values after the highlighted value.

like

Select records for which the value in the selected field matches the pattern of the value specified. The value specified can use the DOS wildcards (? for a single wildcard character, * for a character string) to provide a partial match for the values of interest. This is the equivalent of the SQL Like predicate.

```
CUSTOMER is like Sm?th
```

«Include all records in which the Customer name is like Sm?th. This will include records with customers named Smith and Smyth but not Smithe or Smythe.»

```
CUSTOMER is like Sm*
```

«Include all records in which the Customer name is like Sm*. This will include records with customers named Smith and Smyth as well as customers named Smithe and Smythe.»

NOTE: If you want to select values that include the literal * or ? characters, use a backslash before the literal character. For example:

Password is like Sm/?* will include only those records in which the password begins with the characters "Sm?". It will include records with passwords like Sm?WEj, Sm?oot but not Smith or Smoot. The program sees the question mark as a literal character (because it is preceded by a backslash) but it views the asterisk as a wildcard (because it is not preceded by a backslash).

formula

Formula brings up a formula edit box.

If you have not already created a record selection formula using the selected field, the formula edit box will be empty. While you can enter a record selection formula in this box, it is far easier to do it using the Report|Edit Record Selection Formula command. With that command you get a full-featured Formula Editor that enables you to select fields, functions, and operators form list boxes. If you have already created a record selection formula using the selected field, the formula edit box will display that formula. You can modify the formula as needed in that edit box.

Date conditions

If you use Select Records on a date field, seven of the primary condition options will appear plus the following option:

in the period

Select records for which the value in the selected field falls within the date range specified. When you select this condition, the dialog box displays a scroll list of all Crystal Reports date conditions. Select the condition you want from the list.

```
header.DATE is in the period Calendar1stQtr
```

«Include all records in which the date falls within the calendar first quarter of the year. Dates from January 1 to April 30 (including January 1 and April 30) will be included; all other dates will be excluded.»

Boolean conditions

If you use Select Records on a Boolean field, your condition options will include only is any value and satisfies the test below from the primary condition list above plus the following two options:

is True

Select records for which the value in the selected field is true.

```
file.REGISTERED is True
```

«Include all records in which the value in the file.REGISTERED field is True.»

is False

Select records for which the value in the selected field is false.

file.REGISTERED is False

«Include all records in which the value in the file.REGISTERED field is false.»

Default Fonts dialog box

Options for changing the Default Section fonts appear when you Click the Fonts tab while using the File| Options command.

Use the Fonts tab to change the default fonts for any and all sections of your report. The Fonts tab allows you:

- to specify different fonts for different sections of your report, and
- to specify one font for text elements and a different font for fields within a given section.
- For new reports, once the defaults are changed, Crystal Reports looks at the section in which each new text element is placed and formats it with the text font specified for that section. It also looks at the section in which each new data field or formula result element is placed and formats it with the field font specified for that section.
- For existing reports, the default font for text in a section stays the same. New reports will get the new default.

For example, you can specify one field font for all group values (subtotals, summary fields, etc.) and a completely different text font for the text labels you use to identify each of the group values. Crystal Reports automatically formats all group values with the default field font, and it formats any label you type in the group section with the default text font.

NOTE: The Default Fonts dialog box gives you the opportunity to customize the Crystal Reports Design Window to best fit your needs. When you make changes via the dialog box, Crystal Reports simply changes the defaults so the fonts used in each section appear in the format you typically want them in. These default changes don't in any way limit the fonts available for use in any section of your report, however. You still have the ability to reformat text or field elements individually if you wish.

Font dialog box

The Font dialog box appears under several circumstances:

- when you select text or a field and then select the Format|Font command (or Change Font from the right mouse key menu), and
- when you select a section from the Default Fonts dialog box while using the File|Options command.

Use the Font dialog box to change the fonts, the font size, and/or the font style you use for text and data fields on your reports.

NOTE: You can change the default field and text fonts used for each section of your report via the File|Options command.

NOTE: When you change fonts, Crystal Reports automatically adjusts line and letter spacing to accommodate your change.

See Also

Default Fonts dialog box

Formatting options common to most fields

When you format a field using the Format|Field command, a dialog box appears with formatting options specific to the data type of the selected field. The following options are available in all or most of those dialog boxes:

Format options available for all field types

Suppress if Duplicated

When activated, nothing is printed in a column if it duplicates data on the previous line; the data only prints once. For example, to print the customer number only once for each customer, activate the option for the customer number field.

Hide when printing

When activated, nothing will print in the column. This is useful if, for example, you want to put a field on the report to be used in calculations or sorting, but do not want the field to print.

Alignment

Alignment refers to the placement of the field value within the space allotted for the field on the report. You have the following choices:

- *Default* restores the default alignment (flush left for text, date, and Boolean fields, flush right for number and currency fields).
- *Left* places all field values flush left in the space allotted. The first character in the value is flush against the left margin of the column. Thus, when you select Left, the first character in each value is aligned.
- *Centered* centers the field value within the space allotted.
- *Right* places all field values flush right in the space allotted. The last character in the value is flush against the right margin of the column. Thus, when you select Right, the last character in each value is aligned.

Format options available only for number, currency, and date fields

Use Windows Default Format

Uses the default date, number, and currency formats as specified in the International dialog box in the Windows' Control Panel.

NOTE: When the Use Windows Default Format switch is activated, your options for formatting fields "on the fly" are limited. Several of the options for each data type are not selectable in the Format Number (Format Date, etc.) dialog boxes when the switch is toggled on. If you want to format "on the fly" and the options you need are not selectable, first toggle the Use Windows Default Format switch off, and then select the formatting options you need.

NOTE: When the Use Windows Default Format switch is toggled Off, Crystal Reports uses the format defaults as specified in the Options dialog box. You can override these defaults, however, to format individual fields "on the fly" via the Format|Field command.

Format Memo dialog box

The Format Memo dialog box appears whenever you select a memo field and then select the Format| Field command (or the Change Format option from the right mouse key menu).

Use the Format Memo dialog box for formatting memo fields in your report. You use this dialog box to specify the way you want the selected memo field value to appear on your report.

Make your formatting selections and Click OK when finished.

See Also

Using memo fields with Crystal Reports

Format Currency dialog box

This dialog box appears in two different situations:

- when you click the Currency button on the Fields tab while using the File|Options command, and
- when you select a currency field and then select the Format|Field command (or the Change Format option from the right mouse button menu).

You use this dialog box to specify the way you want the values in the selected field to appear on your report.

The dialog box is identical to the Format Number dialog box. Please see Format Number for a complete discussion of all the options.

NOTE: dBASE does not offer a currency data type. To format currency fields from dBASE files, use the currency options in the Format Number dialog box.

Format Boolean dialog box

This dialog box appears in two different situations:

- when you Click the Boolean button on the Fields tab while using the File|Options command, and
- when you select a Boolean field and then select the Format|Field command (or the Change Format option from the right mouse button menu).

Use this dialog box whenever you wish to change the format of a Boolean (Yes/No) field on your report.

Format String dialog box

This dialog box appears in two different situations:

- when you select the String option on the Fields tab while using File|Options
- when you select a string field and then select Format|Field (or Change Format option from the right mouse key menu).

Use the Format String dialog box for formatting string fields in your report. You use this dialog box to specify the way you want the selected string field value to appear on your report.

Format Number dialog box

This dialog box appears in two different situations:

- when you Click the Fields tab while using the File|Options command, and
- when you select a number field and then select the Format|Field command (or the Change Format option from the right mouse button menu).

Use the Format Number dialog box for formatting numbers and currency values in your report.

NOTE: The currency provision is included in this dialog box to make it easier for you to work with fields from dBASE databases, since dBASE doesn't offer a* currency *data type.

Format Date dialog box

This dialog box appears in two different situations:

- when you click the Date button on the Fields tab while using the File|Options command, and
- when you select a date field and then select the Format|Field command (or the Change Format option from the right mouse button menu).

You use this dialog box to specify the way you want the selected date value to appear on your report.

The dialog box contains three options for formatting dates.

- The top box allows you to set the order in which the elements of the date (Month, Day, and Year) are to appear.
- The center box allows you to set the style (numbers, abbreviated numbers, text, etc.) for each of the elements.
- The bottom box shows you the results of your formatting selections.

Remove from Report dialog box

The Remove from Report dialog box appears when you select the Database|Remove from Report command. Use this dialog box to delete databases from the active list so they can no longer be used in your report.

- The alias for each of the active databases appears in the **Databases** box.
- The name of the file highlighted in the Databases box (and its path if different than the directory where CRW resides) appears in the **Locations** box.

See Also

Finding fields Remove says to remove

Deleting files using Remove from Report command

Choose New Location dialog box

The Choose New Location dialog box appears when you Click the Set Location button in the Set Location dialog box while using the Database|Set Location command. Use this dialog box to specify a new location for a database used in a report.

To choose a new location:

- If you wish to choose a database with a different name, select the database from the list in the File Name scroll box or type the new name in the File Name text box at the top.
- If you wish to change the directory or disk location of the file, type in the new location beneath the Directories heading or set the new location using the Directories scroll box. When you make a change in this box, Crystal Reports displays the new location beneath the Directories heading.
- If you wish to change both the name and location, you can type in the new name and include the new path in the File Name text box or, you can type in the new name in the File Name text box and select the new location using the Directories scroll box.
- Click the OK button. Crystal Reports changes the report to reflect the new name/location you have chosen.

NOTE: This option does not physically move the database(s). It simply directs Crystal Reports to look for the database(s) in a different location than you originally specified when setting up the report.

Alias Name dialog box

Use the Alias Name dialog box to change the alias assigned to the selected database.

This dialog box appears automatically whenever you select a new database if you have toggled the Use Default Alias option *Off* on the Database tab while using the File|Options command. That option is toggled *On* by default. The Use Default Alias option tells Crystal Reports to automatically use the database name (less the extension) as the default alias. For example, the database **company.dbf** will automatically generate the alias **company**.

See Also

Selecting an alias

Set Alias dialog box

The Set Alias dialog box appears when you select the Database|Set Alias command. Use the Set Alias dialog box to select an alias that you want to change.

- The **Databases** box contains a list of aliases used in the current report, and the first alias on the list is highlighted.
- The **Location** box at the bottom of the dialog box displays the name (and the path, if different from the directory holding CRW.EXE or CRW32.EXE) of the database referenced by the highlighted alias.
- The **Set Alias** button takes you to the Alias Name dialog box. You use that dialog box to change the alias for the selected database.

See Also

Selecting an alias

Format Section (formatting) dialog box

The Format Section (formatting) dialog box appears whenever you select a section to format from the Format Section (sections) dialog box while using the Format|Section command.

Use the Format|Section (formatting) dialog box to make formatting changes that affect entire sections of your report. This dialog box allows you to:

- hide a section (keep it from printing),
- print subtotals or group values only at the bottom of the page,
- insert a page break before the section is printed,
- insert a page break after the section is printed,
- reset the page number to one (1) after a group value prints,
- prevent page breaks from spreading data from a single record over two pages,
- prevent blank lines from printing,

create multiple columns, and

- format mailing labels and label type items.

Print Report Definition

A report definition is printed when you select File|Print|Report Definition.

A report definition is a report on a report; it identifies the components of the report, and it provides important information about each of the components. Print Report Definition prints a copy of the report definition for the active report. The report is only printed to the default printer.

See Also

Print Report Definition command

Export dialog box

The Export dialog box appears when you select the File|Print|Export command. This dialog box lets you print your report to an electronic file in a format that can be read by another software application. For example, you could export your report in Microsoft Excel format and then open it in Excel as a normal spreadsheet file.

The Export facility supports several popular word processing, database, and spreadsheet formats and a number of standard data interchange formats as well. Your format choices are:

Word processor specific formats

Word for Windows

Word for DOS

Word Perfect

Rich Text Format (RTF)

Spreadsheet specific formats

Lotus 1■2
■3 (WKS)

Lotus 1■2
■3 (WK1)

Excel 2.1 (XLS)

Excel 3.0 (XLS)

Excel 4.0 (XLS)

Common data interchange formats

Comma separated values (CSV)

Encloses alphanumeric field data in quotes and separates fields with commas.

Tab separated values

Presents data in tabular form. Encloses alphanumeric field data in quotes and separates fields with tabs.

Character separated values

Encloses alphanumeric field data in quotes and separates fields with the character of your choice. When you make this selection, select your destination, and Click OK, the Character Separated Values dialog box appears. Enter your Separator and Quote selections in the respective edit boxes.

- *Separator* specifies the character you want to use to separate the fields in the Character separated value format.
- *Quote s*pecifies double or single quotation marks to enclose alphanumeric field data in the Character separated value format.

Text style

Saves the data in ASCII text format with all values separated by spaces. This style looks most like the printed page.

Tab separated text style

Saves the data in ASCII text format with all values separated by tabs.

DIF style

Saves the data in DIF (data interchange format) format. This format is often used for the transfer of data between different spreadsheet programs.

If Crystal Reports does not export directly to your application's native format, it more than likely exports to a format your application can read nonetheless. For example, many applications can read data saved in an ASCII or text format. Even though your application isn't on the list of supported formats, you may be able to export to a text file and then have your application read in the data (import it) from that format. Check your application's documentation to see which formats it can import.

Crystal Reports also allows you to attach an exported file to a message sent via an electronic mail system. The message displays an icon that indicates a report is attached. The person who receives the message Double Clicks the icon and the report appears on screen.

Record Sort Order dialog box

The Record Sort Order dialog box appears whenever you select the Report|Sort Order command.

Use the (Record) Sort Order dialog box to define how you want the records in your report to be sorted for printing. You can add and remove sort fields and define the sort direction (ascending or descending) for the data in your report.

See Also

How data types are sorted

Multiple field sorts

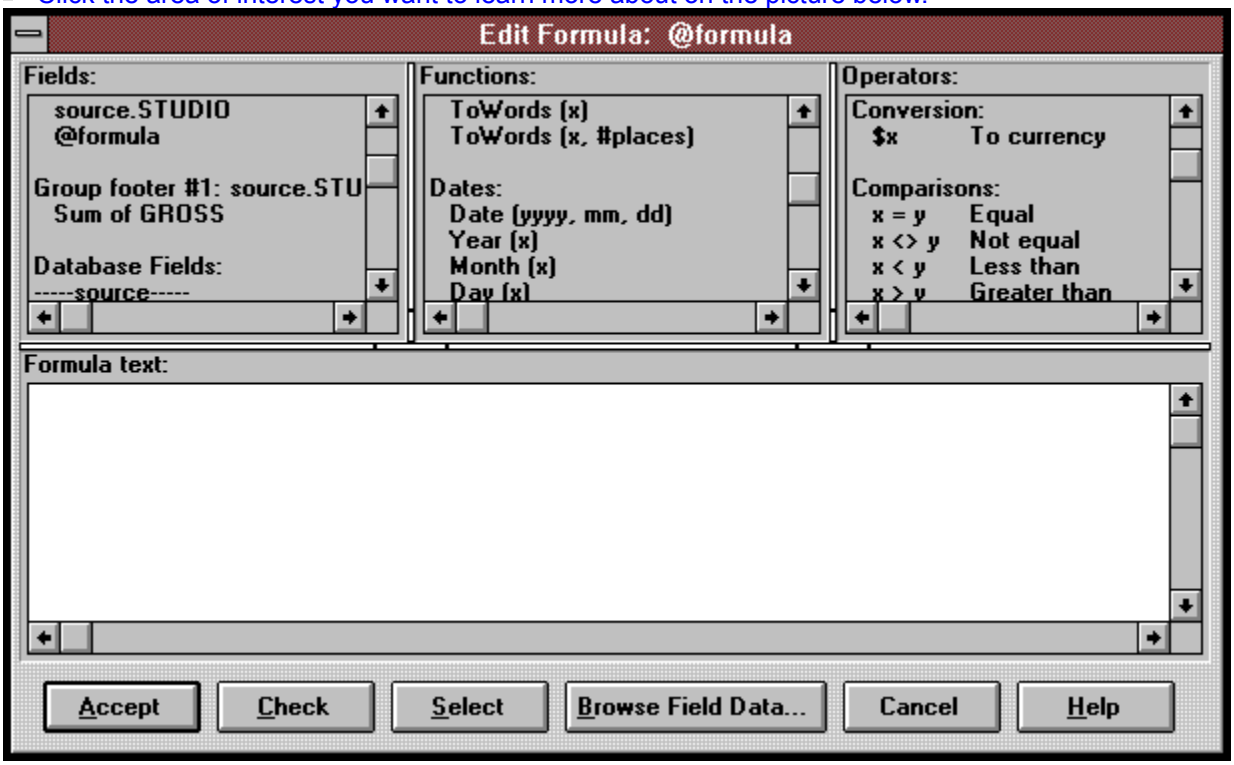
How to identify the "top" groups

Using Group Sort, Record Sort Order together

Edit Formula dialog box ▪ the Formula Editor

Use the **Edit Formula dialog box** to create and edit formulas for use in your report. A variation of this dialog box appears when you select the Insert|Formula Field command, the Edit|Formula command, the Report|Edit Selection Formula|Record, and the Report|Edit Selection Formula|Group command.

- Click the area of interest you want to learn more about on the picture below.



Formula text box

You edit your formula in the large white formula text box.

Accept button

The *Accept* button tests the syntax of your formula, and if correct, enters the formula in the report, replacing the earlier version of the formula (if any). If the syntax is incorrect, Crystal Reports gives you the opportunity to correct the error(s) prior to entering the formula in the report.

Check button

The *Check* button tests the syntax of your formula and identifies syntax errors if they are found. Unlike the Accept button, this button does not enter your formula in the report if the syntax is correct; Check is intended for interim syntax checks while you are building a formula.

Select button

The Select button inserts an item once you have highlighted it in the Fields, Functions, or Operators box.

Browse Field Data

Browse Field Data enables you to review values for the selected field and to paste individual values directly into your formula if you wish. When you Click the Browse Field Data button, a dialog box appears with a scroll list of those values.

- The name of the field selected is at the top of the dialog box.
- The data type of the field (number, string, etc.) is listed immediately below the field name.
- The length of the field is listed below the Type.
- Field values are listed in the scroll box.
- To paste a field value directly into your formula, highlight the value of interest and Click the Paste Data button (or Double Click the value of interest). Crystal Reports pastes the value at the insertion point in the formula.

Fields box

The *Fields* box displays a list of those database fields that are available for using in the report. Fields are listed in the following formats:

database fields	file.*fieldname*
formulas	@Formulaname
group fields	Grouped by *condition field*
	Sum of *fieldname*

When you select a field from the list, Crystal Reports inserts in the in the report at the insertion point.

NOTE: For large Btrieve .ddf files (.ddf files that contain four or more database files), Crystal Reports displays the names of the files in the .ddf file, not the individual field names. To review the field names in individual files:

- Double

Click the file name to select the file of interest. The Formula Editor Select button changes to an Open button.

- Click the Open button and the program lists all of the fields in the selected database.
- Select the field(s) of interest as you would from any other kind of database (.dbf, .db, etc.)

Functions box

The *Functions* box displays a list of Crystal Reports functions available for use in the formula. When you select a function from the list, Crystal Reports inserts in the in the report at the insertion point. The function is inserted complete with its required syntax items (parentheses, commas, quotation marks, etc.) to make your work easier. For detailed information on individual functions, see the Functions Index.

Operators box

The *Operators* box displays a list of Crystal Reports operators available for use in the formula. When you select an operator from the list, Crystal Reports inserts in the in the report at the insertion point. For detailed information on individual operators, see the Operators Index.

Entering a formula in the Formula Editor

1. Enter your formula in the Formula text box.

- If your formula is to include text strings, type those strings wherever they are to appear in the formula. Each such string must be surrounded by single or double quotation marks.
- If your formula is to include functions and/or operators, you may either type them in or select them from the lists that appear in the Functions and Operators boxes.
- If your formula is to include field names, you can type them in or select the names from the list that appears in the Fields box. If you choose to type them in, you must enter them in the format

```
{alias.fieldname}
```

- If your formula is to include other formulas, you can type them in or select the names from the list that appears in the Fields box. If you choose to type them in, you must enter them in the format

```
{@formulaname}
```

- If your formula is to include groups, it is recommended that you select the group from the list that appears in the field box rather than type it in (due to the complexity of the syntax).

2. Use the Check option to check the syntax of your formula at any time while you build it.
3. When you have completed your formula, select Accept to enter the completed formula in your report. Crystal Reports checks the syntax and, if it is correct, displays the field placement box. (If the syntax is not correct, you will need to make the necessary syntax changes first. Then, when you select Accept, the pointer will appear.)
4. Move the field box to the place you want to insert the formula and Click your left mouse button to insert it.

NOTE: You can't use commas in numbers you enter in a formula. Enter only the number itself.

See Also

Index To Formula Topics

Insert Database Field dialog box

The Insert Database Field dialog box appears in two situations:

- when you select the Insert|Database Field command, and
- when you select File|New|Custom Report to begin a new report or File|New|Custom Mailing Labellabelsrpt and then select a database from the Choose Database File dialog box.

Use the Insert Database Field dialog box to select fields for inclusion in your report.

- The name of the current report appears at the top of the dialog box, just below the title bar.
- The available fields are listed in the scroll box. Fields are grouped by database in the scroll box list, and each group is headed by the alias selected for the database from which the fields come. You use the scroll box list to select (highlight) the field you want to insert in your report.

Insert Formula dialog box

The Insert Formula dialog box appears whenever you select the Insert|Formula Field command.

Use the Insert Formula dialog box to:

- select an existing formula for inclusion in your report,
- delete an existing formula specification or
- to specify a name for a new formula.

See Also

Deleting formulas from your report

Formulas ■ an overview

Index To Formula Topics

Number and Date Format Dialog box

This dialog box appears after you click OK in the Export File dialog box. Two check boxes appear in the dialog box:

- Select the Same number formats as in report check box if you want the number formats you have defined in your report to be used to export your data to the new disk file.
- Select the Same date formats as in report check box if you want the date formats you have defined in your report to be used to export your data to the new disk file.

Insert Subtotal dialog box

You use this dialog box to set the conditions that trigger the printing of a subtotal.

The top scroll box

The top scroll box contains a list of the fields and formulas being used in the report. Click the scroll arrow to reveal the list, and select from that list the field that you want the program to use for triggering subtotals.

- The program will first sort report data based on this field.
- Then it will go down the report and group and subtotal the data whenever the value in the field changes.

If you select a date or Boolean field, the program gives you the ability to further narrow your selection. See Date and Boolean conditions below.

The sort order scroll box

The next scroll box lists the four sort direction options

The default option is *in ascending order*. If you want to change the sort direction, Click the scroll arrow to reveal the options and make your selection from the list.

NOTE: If you select* in Specified order*, the dialog box will display more options.

Date and Boolean conditions

When you subtotal based on changes in a Date or Boolean field, Crystal Reports displays another scroll box at the bottom of the dialog box. This scroll box enables you to further define your subtotal specification.

Date Conditions

This new scroll box contains a list of date ranges that define typical subtotaling periods. When you select a date condition, Crystal Reports first sorts your data by date. Then it breaks the data into groups and prints a subtotal whenever the date condition that you select is met.

Click the scroll arrow to reveal the list of date conditions. Your options are:

for each day	Prints a subtotal whenever the date changes
weekly	Prints a subtotal at the change from one week to the next (a week runs from Sunday through Saturday).
for each two weeks	Prints a subtotal every two weeks (weeks run from Sunday through Saturday).
for each half month	Prints a subtotal twice a month
for each month	Prints a subtotal at the end of each month.
for each quarter	Prints a subtotal at the end of each calendar quarter.
for each half year	Prints a subtotal at the end of each calendar half year.
for each year	Prints a subtotal at the end of each year.

Boolean Conditions

When you subtotal based on changes in a Boolean field, Crystal Reports gives you a selection of Boolean conditions that can trigger a subtotal. The program first sorts your data. Then it breaks the data into groups and prints a subtotal whenever the condition you select is met. Click the scroll arrow to reveal the list of Boolean conditions.

NOTE: In the following conditions, Yes means Yes, True, or 1 (depending on the Boolean format you have selected for the field) and No means No, False, or 0).

Your options are:

any change

Prints a subtotal whenever the value changes from *Yes* to *No* or from *No* to *Yes*.

Subtotal calculation with *any change* condition
Yes
Yes
Subtotal
No
Subtotal
Yes
Subtotal
No
No
Subtotal

change to Yes

Prints a subtotal whenever the value changes from *No* to *Yes.*

Subtotal calculation with *change to Yes* condition
Yes
Yes
No
Subtotal
Yes
No
No
Subtotal

change to No

Prints a subtotal whenever the value changes from *Yes* to *No*.

Subtotal calculation with *change to No* condition
Yes
Yes
Subtotal
No
Yes
Subtotal
No

No

Subtotal

every Yes

Prints a subtotal every time the value is *Yes*.

Subtotal calculation with *every Yes* condition.

Yes

Subtotal

Yes

Subtotal

No

Yes

Subtotal

No

No

Subtotal

every No

Prints a subtotal every time the value is *No.*

Subtotal calculation with *every No* condition.

Yes

Yes

No

Subtotal

Yes

No

Subtotal

No

Subtotal

next is Yes

Prints a subtotal whenever the next value is a *Yes*.

Subtotal calculation with *next is Yes* condition.

Yes

Subtotal

Yes

No

Subtotal

Yes

No

No

Subtotal

next is No

Prints a subtotal whenever the next value is a *No*.

Subtotal calculation with *next is No* condition.

Yes

Yes

Subtotal

No

Yes

Subtotal

No

Subtotal

No

Subtotal

Insert Grand Total dialog box

The Insert Grand Total dialog box appears when you select the Insert|Grand Total command.

You use this dialog box to select the kind of grand total you want to appear on your report. Your options are:

- *Sum* totals all the values in the selected field for the entire report.
- *Average* calculates the average (mean) value in the selected field for the entire report.
- *Maximum* identifies the highest value in the selected field for the entire report.
- *Minimum* identifies the lowest value in the selected field for the entire report.
- *Count* counts the number of values in the selected field for the entire report.
- *Sample Variance* determines the variance of all values in a set of values that is typically a subset of an entire population. This can be used for projecting the variance for an entire population based on just a sample of that population. It uses (*N*
1) in its calculations. When you make this grand total selection, the program determines the sample variance for all values in the selected field for the entire report.
- *Sample Standard Deviation* determines the standard deviation of all values in a set of values that is typically a subset of an entire population. This can be used for projecting the standard deviation for an entire population based on just a sample of that population. It uses (*N*
1) in its calculations. When you make this grand total selection, the program determines the sample standard deviation for all values in the selected field for the entire report.
- *Population Variance* determines the variance of all values in an entire population. It uses (*N*) in its calculations. When you make this grand total selection, the program determines the population variance for all values in the selected field for the entire report.
- *Population Standard Deviation* determines the standard deviation of all values in an entire population. It uses (*N*) in its calculations. When you make this grand total selection, the program determines the population standard deviation for all values in the selected field for the entire report.

NOTE: You can't sum or average a text, Boolean, or date field.

Insert Summary dialog box

Use the Insert Summary dialog box to set up a summary operation on a field in your report.

The top scroll box contains a list of the available summary operations. Click the scroll arrow to reveal the list. You have nine options in the scroll box:

- *Sum* totals the values in the group. This is the same as a subtotal.
- *Average* calculates the average (mean) value in the group.
- *Maximum* identifies the highest value in the group.
- *Minimum* identifies the lowest value in the group.
- *Count* counts the number of values in the group.
- *Sample Variance* determines the variance of all values in a set of values that is typically a subset of an entire population. This can be used for projecting the variance for an entire population based on just a sample of that population. It uses (*N*
1) in its calculations.
- *Sample Standard Deviation* determines the standard deviation of all values in a set of values that is typically a subset of an entire population. This can be used for projecting the standard deviation for an entire population based on just a sample of that population. It uses (*N*
1) in its calculations.
- *Population Variance* determines the variance of all values in an entire population. It uses (*N*) in its calculations.
- *Population Standard Deviation* determines the standard deviation of all values in an entire population. It uses (*N*) in its calculations.

NOTE: You can't sum or average a text, Boolean, or date field.

Select the option of interest.

The second scroll box contains a list of the fields and formulas being used in the report. Click the scroll arrow to reveal the list, and select from that list the field that you want the program to use for triggering summaries.

- The program will first sort report data based on this field.
- Then it will go down the report and group and summarize the data whenever the value in the sort and group by field (the trigger field) changes.

If you select a date or Boolean field, the program gives you the ability to further narrow your selection. See *Date and Boolean* conditions below.

The next scroll box lists the four sort direction options. The default option is *in ascending order*. If you want to change the sort direction, click the scroll arrow to reveal the options and make your selection from the list.

NOTE: If you select* in Specified order*, the dialog box will display more options.

Date and Boolean conditions

When you summarize based on changes in a Date or Boolean field, Crystal Reports displays another scroll box at the bottom of the dialog box. This scroll box enables you to further define your summary specification.

Date conditions

This scroll box contains a list of date ranges that define typical summarizing periods. When you select a date condition, Crystal Reports first sorts your data by date. Then it breaks the data into groups and prints a summary whenever the date condition you select is met. Click the scroll arrow to reveal the list of date conditions and select the condition you want. For further information, see the discussion of Date field conditions.

Boolean conditions

When you subtotal based on changes in a Boolean field, Crystal Reports gives you a selection of Boolean conditions that can trigger a subtotal. The program sorts your data. Then it groups the data and prints a subtotal whenever the condition you select is met. Click the scroll arrow to reveal those conditions and select the condition you want. For further information, see the discussion of Boolean field conditions.

- Select OK when finished. Crystal Reports sorts, groups, and summarizes your data and inserts the summary in the group section of your report. You can then move it into position, wherever you want it to print in that section.

NOTE: Once you have set up your summary and closed this dialog box, if you then want to modify your summary you must use Insert|Group Section to do so. If you call up Insert|Summary again, you will only be able to review your settings, not change them.

Technical Support Request dialog box

The Technical Support Request dialog box appears whenever you select the Help|Technical Support Request command.

Use the Technical Support Request dialog box to request assistance with Crystal Reports from the technical support department.

The dialog box provides you with a step▪by

▪step procedure for getting technical support as well as a technical support request form. You can fill out the form in your computer, print it out, and then FAX it or mail it to the technical support department for a timely response.

File Open dialog box

The File Open dialog box appears whenever you select the File|Open command. Use the File Open dialog box to select the report file you wish to open.

File Save As dialog box

This dialog box occurs in two situations:

- when you select the File|Save command while working on a new report that you haven't yet saved, and
- whenever you select the File|Save As command.
- If you are working on a new report and haven't yet saved it, use this dialog box to specify the name (and path) under which you want to save your report file.
- If you have made changes to an existing report file and want to save the changed version of your file under a different name than that of the original version, use this dialog box to specify the name under which you want to save your changed file. All changes you have made while working on the report will be saved to the new file. Your original report file will remain unchanged. If your original report is a new report, however, i.e., untitled, that file will not be saved.

NOTE: Saving the report under the name of an existing report overwrites the contents of the existing report file.

Choose Database File dialog box

The Choose Database File dialog box appears:

- when you create a new report using the File|New command, and
- when you select the Database|Add Database to Report command.

Use the **Choose Database File dialog box** to activate a database for use in your report.

Print Setup dialog box

The Print Setup dialog box appears whenever you select the File|Printer Setup command.

Use the **Print Setup dialog box** to select the printer, page orientation, and paper size you want to use to print the report. If you don't select a printer, Crystal Reports will print to the Windows default printer.

Choose Export File dialog box

The Choose Export File dialog box appears when you click the OK button in the Number and Date Format dialog box when using the File|Print|Export command.

Use the Choose Export File dialog box to specify the file name (and path) to which you want to print the file.

The Button Bar

Crystal Reports groups several commonly used commands on a Button Bar that remains on screen at all times (unless you choose to turn it off using the File|Options Display Button Bar checkbox). Each command is represented by an individual button, and each button displays a graphic that visually describes the command. You activate Button Bar commands by clicking the appropriate button one time with the left mouse button.

The Button Bar eliminates some of the steps needed to activate the included commands, and it can thus greatly speed your work in creating reports. Each command available via the Button Bar is discussed in its own Help topic.

The buttons on the button bar perform the following functions:

Create a new report

Open an existing report

Save the report

Print report to the Preview window

Print report to a printer

Refresh Report data

Export report to a file

Send the file to E-Mail

Launch report to server

Select fields

Undo Button

Change Views

Search

Insert a text field

Insert a database field

Insert a summary

Insert a formula field

Set record or group sort order

Set Record Selection Criteria

Insert Graph/Chart

Visual Linking

Draw a Line on report

Insert a picture

Draw a box on report

Format Bar

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

To learn more, click the button you are interested in on the picture below.

Search Notes

When you click the Search button or select Report|Search, the program calls up one of three dialog boxes. The dialog box that appears is determined by what (if anything) you have selected in the report at the time you select Search.

- If you have a field selected and then select Search, the Search Records dialog box appears. You can use this dialog box to set up your search criteria based on the selected field.
- If you have no field selected or if you have a summary field selected, the Choose Field dialog box appears. This dialog box lists fields from the Details section of the report. You can only search the Details section through this dialog box. You can use this dialog box to select the field on which you want to base your search criteria.
- In a number of other circumstances, the Search Records in Details Section dialog box appears. You can use this dialog box to set up or modify your search criteria.

NOTE: If you have hidden the Details section of your report, the program will allow you to set up search criteria. If you are basing your search on a field in the hidden Details section, however, the program will not find records for you.

NOTE: Search is section specific. You can base search criteria on multiple fields but the fields must be in the same section. Consider the following section from a report.

	NSW	Australia	200,000
	Australia		200,000
Group Footer # 2	AL	USA	250,000
	AZ	USA	100,000
Group Footer #1	USA		350,000

In this example, the data is subtotaled by state and also by country. The State field and the Country field are placed in the group footer #2 section as "live" labels. To search for data for AZ USA, you have to set up a search for AZ in the Group Footer #2 section and for USA in the same section. If you search for AZ in the Group Footer #2 section and USA in the Group Footer #1 section, your search will not produce the results you want.

Use the New Select button in the Search Records or Search Records in Details Section dialog boxes to set up a search criterion based on more than one field in a section..

Search (Report menu)

You can use the Search feature to search for particular values in a report. Setting up a search is an easy two step process:

- You first select the field you want to perform the search in,
- Then you specify the value you want to search for.

NOTE: Search can only be performed while in the Preview Window, or in a Drill Down Window.

See Also

The Search Records dialog box

The Choose Field dialog box

Search Notes

- **The Search Records dialog box**
- Click the area of interest you want to learn more about on the picture below.

NOTE: The Search Records dialog box will sometimes appear as the Search Records in Detail Section dialog box depending on your selections.

See Also

Search Notes

The *is, is not* drop down box

You select is, or is not from this drop down list box. As an example, you could search for a value that is equal to a specified value, or that is not equal to a specified value. Use the is and is not selectors to help make your searches more specific.

The Find Again button

Click the Find Again button to locate another instance of the value you have previously searched for in the report. If the value is found again, it will be highlighted in black in the Preview Window. If the value is not found, a message stating "Record Not Found" will be displayed.

The Browse Data button

Click the Browse Data button to Browse the data for the field you are basing your search on. When you Click the Browse Data button, a dialog box with the field name will appear. You can scroll down the list and select values to paste to the Search Records dialog box by Clicking the Paste Data button. Click Done to return to the Search Records dialog box when you are finished Browsing Data.

Show Formula button

Click the Show Formula button to look at the formula Crystal Reports has generated for your search based on your selections in the Search Records dialog box.

The Up and Down radio buttons

You select one of these radio buttons to specify the direction of the search. You may use **Down** (the default direction) to search through your report, and after reaching the end select **Up** to return to a value in your report that you have already found, but want to look at again.

The New Select button

Click the New Select button when you want to specify another search on a different database field in your report. When you Click the New Select button, the Choose Field dialog box appears. Choose the field you want to perform a search on and Click OK to return to the Search Records Dialog box. The new field appears as a tab in the Search Records Dialog box.

The Arrow buttons

Use the arrow buttons to move left or right among the tabs in the Search Records Dialog box. The arrow buttons become active (darkened in) when there are more tabs than will fit along the top of the dialog box.

The Drop Down Operators Box

Use this box to select a condition for your search.

The following conditions appear whenever you use Search Records with number, currency, string, or memo fields. These conditions can also be reversed in some circumstances by selecting the **is not** condition from the drop down list box.

is equal to

```
company.State is equal to CA
```

«Perform a search that finds only records that have CA(lifornia) as the State field value.»

is one of

```
company.State is one of AZ, CA, or CO
```

«Perform a search that finds only records that have a value in the State field that matches either AZ, CA or CO»

is less than

```
company.State is less than AZ
```

«Perform a search that finds only records that have a value in the State field less than Arizona (alphabetically)»

```
company.LastYrSales is less than 100000.00
```

«Perform a search that finds only records that have a value in the LastYrSales field less than 100000 (numerically)»

NOTE: If you check the or equal to box, the search will find values that are less than or equal to.

is greater than

```
company.State is greater than AZ
```

«Perform a search that finds only records that have a value in the State field greater than Arizona (alphabetically)»

```
company.LastYrSales is greater than 100000.00
```

«Perform a search that finds only records that have a value in the LastYrSales field greater than 100000 (numerically)»

NOTE: If you check the or equal to box, the search will find values that are greater than or equal to.

is between

```
company.State is between AZ and CA
```

«Perform a search that finds only records that have a value in the State field that falls between AZ and CA (alphabetically) or that matches either.»

```
company.LastYrSales is between 10000 and 100000
```

«Perform a search that finds only records that have a value in the LastYrSales field that falls between 10000 and 100000 (numerically) or that matches either.»

is starting with

```
company.State is starting with C
```

«Perform a search that finds only records that have a value in the State field that starts with the letter

"C".»

```
company.Coname is starting with Cu
```

«Build a group that contains only records that have a value in the Coname field that starts with the letters "Cu"»

like

```
company.Coname is like C?m*
```

«Perform a search that finds only records that have a value in the Coname field that matches the pattern "C?m*", where "?" represents any single character and "*" represents any number of characters. This would find "Computer" and "Camera" but not "Coombs".»

```
company.Coname is like C*m*
```

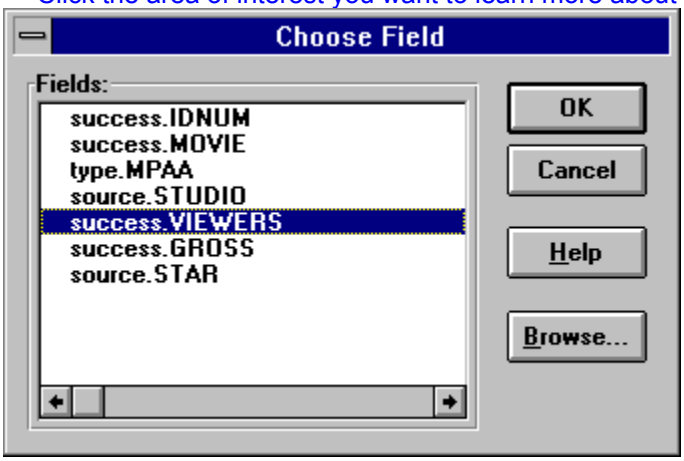
«Perform a search that finds only records that have a value in the Coname field that matches the pattern "C*m*" where "*" represents any number of characters. This would find "Computer" and "Camera" as well as "Coombs".»

The Drop Down Edit Box

You use the Drop Down Edit Box to enter the value you are basing your search on. You can either type the value directly into the box, or Click the down arrow on the right side of the drop down box to open a list of values. Select the value from the list, and it appears in the edit box.

Alternatively, you can Click the Browse Data button. A dialog box appears displaying a list of values. Simply select the value you want, and Click the Paste button to paste it to the edit box.

- **The Choose Field Dialog box**
- Click the area of interest you want to learn more about on the picture below.



The Browse button

Click the Browse button to Browse the data for the field you have selected. A dialog box with the field name will appear. You can scroll down the list to preview what kind of data is stored in the selected field. Click Done to return to the Choose Field dialog box when you are finished Browsing Data.

The Fields box

Use the Fields box to Choose the field you want to base your search on. All fields used in the report you have open at the time will appear in the Fields box. When you Click on a field name in the Fields box, the name becomes highlighted. Click OK to use the highlighted field for your search.

The Change Views button

Use this button to switch between the three views available to you in the Preview Window

The Undo button

The Undo command is a dynamic command. The text changes on the menu based on the last action you performed in Crystal Reports. The command supports multiple levels of Undo. Each time you Click the Undo button, you Undo the most recent action you have performed in Crystal reports.

Drop down lists

Two drop down lists appear at the left side of the Format Bar.

You use these for selecting new fonts and font sizes if you want to use something different than the default. To make a selection from either box, Click the arrow to reveal your options and then Click the option you want.

The Bold button

Changes the selected data to boldface.

The Italic button

Italicizes the selected data.

The Underline button

Underlines the selected data.

Increase Font Size button

Increases the font size one point each time you click the button.

Decrease Font Size button

Decreases the font size one point each time you click the button.

The Left Justify button

Aligns the selected data flush left.

The center text button

Centers the selected data.

The Right Justify button

Aligns the selected data flush right.

The Dollar Sign button

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

The Comma button

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

The Percent button

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

The Add Decimal button

Adds one decimal place to a number.

The Subtract Decimal button

Subtracts one decimal place from a number.

The Styles button

When you click the Styles button, the Report Style Expert dialog box appears. Select a style from the list to apply to your report. As you select different styles from the list, the example box on the right changes to show you what the selected style looks like. Click OK to apply the changes to your report.

The Cut button

Cuts the current text and places it on the clipboard.

The Copy button

Copies the selected text to the clipboard.

The Clipboard button

Pastes the text currently on the clipboard.

Crystal Reports cursors

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

The **Arrow** cursor is the primary cursor. You will use this cursor everywhere but where you can type text. The cursor is used for making menu selections, selecting options from dialog boxes, working with scroll bars, etc.

A **Double-Arrow** cursor is a resizing cursor. The cursor changes to one of a number of different double{bmc endash.bmp} arrow cursors whenever it is over a resizing handle on a bit-mapped graphic, a graphic box, a graphic line, or a field.

The **Pencil** cursor is a drawing cursor. It appears whenever you select Insert|Box or Insert|Line. The point of the pencil marks the spot where the drawing begins and is used to define the size and shape of the object drawn.

The **I-beam** cursor is active whenever you are working in the text entry sections of the Report Designer, the Formula Editor, and some of the dialog boxes. The I-beam cursor is the cursor you use to select report elements, and to set the position of the insertion point.

The **Drag** (or **Stop**) cursor. This cursor appears whenever the item you are dragging is over an area in which it cannot be dropped.

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

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

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

A **Section Sizing** cursor. The cursor changes to the section sizing cursor whenever it is positioned over one of the lines dividing report sections. Using this cursor you can drag a section boundary line to expand or reduce the size of a section.

The **Insertion Point** identifies the location for entering text. This cursor is available in the text entry sections of the Design Window, the Formula Editor, and some of the dialog boxes. You set the insertion point by positioning the I-beam cursor and then clicking.

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

The **Rectangle** cursor is the cursor you will use for placing fields and formulas in the Report Designer. This rectangle is a graphic approximation of the field and formula markers Crystal Reports uses to

represent fields and formulas in the Designer.

The **Crosshair** cursor is the cursor that appears when you choose Edit|Select Fields or you Click the lasso button. You use this cursor for
lassooing or marquis selecting fields.

Crystal Reports Registration dialog box

The Crystal Reports Registration dialog box appears whenever you start Crystal Reports unless you have registered your copy of the program with the company, received a serial number, and entered it in the Enter serial number... edit box to disable this opening dialog box.

There are two good reasons to register your copy of Crystal Reports:

- Registration entitles you to technical support should you ever require assistance in using the product.
- Registration assures you that you will be notified whenever the product is upgraded to offer new features, benefits, and efficiencies.

Format Section (sections) dialog box

The Format Section (sections) dialog box appears when you select the Format|Section command. Use this dialog box to select the report section you wish to format.

Format Border & Colors dialog box

The dialog box contains a variety of buttons and switches for formatting field text, the field fill (background), the field border, for adding drop shadows, and for controlling the size of the border box that encloses the selected field.

NOTE: You can also change text color via the Format\|Font command.

Box Format dialog box

Dialog box options

The dialog box contains four rows of formatting buttons in addition to the standard OK, Cancel, and Help buttons.

Fill Color buttons and check box

The Fill Color check box and the Fill Color buttons have somewhat different functions.

The Fill Color check box

Fill is the color (if any) that you want the program to use to fill up the selected box. Use this check box to toggle fill on and off.

- check mark in the box means that Fill is turned on.
- No check mark means that Fill is turned off.

You simply Click the check box to toggle Fill *on*, and Click it again to toggle Fill *off*. By default Fill is toggled *Off*.

NOTE: Whenever you Click a Fill Color button, the program automatically toggles the Fill Color check box on (puts a check mark in the box). It assumes that if you want to specify the color, you want your box to be filled.

NOTE: When you toggle the Fill Color check box Off (no check mark), you turn fill off and the description None appears to the right of the Fill Color buttons. It is important that you specify a border when you turn the fill off. Otherwise there is no border or fill color to define your box and the box disappears.

Fill Color buttons

Use the Color buttons to specify the color you want to fill the selected box. You can choose from any of the 16 standard Windows colors. When you Click a Fill Color button:

- a black box appears around the button as a highlight (a white box appears around the color *black*), and
- the program displays the name of the highlighted color to the right of the Fill Color buttons.

Border Color buttons and check box

The Border Color check box and the Border Color buttons have somewhat different functions.

The Border Color check box

Use this check box to toggle the border on and off.

- A check mark in the box means the border is turned on.
- No check mark means the border is turned off.

You simply click the check box to toggle the border *off*, and Click it again to toggle the border *on*. By default the border is toggled On.

NOTE: Whenever you Click a border Color, Width, or Style button, the program automatically toggles the Border Color check box on (puts a check mark in the box). It assumes that if you want to specify the color, width, or style, you want a border to appear.

NOTE: When you toggle the Border Color check box Off (no check mark), you turn the border off and the description None appears to the right of the Border Color buttons. It is important that you specify a fill color when you turn the border off, otherwise there is no border or color to define your box and the box disappears.

Border Color buttons

Use the Border Color buttons to specify the color for the border of the selected box. You can choose from any of the 16 standard Windows colors. When you Click a Border Color button:

- a black box appears around the button as a highlight (a white box appears around the color *black*), and
- the program displays the name of the highlighted color to the right of the style buttons.

Width buttons

Use the Width buttons to specify the width (thickness) for the border of the selected box. When you Click a width button,

- black box appears around the button as a highlight, and
- the program displays a text description of the highlighted width to the right of the style buttons.

Your choices include hairline and a number of point sizes from 0.50 points to 3.50 points.

NOTE: A hairline is one pixel wide, based on the output device in use. For example, if your video output shows 70 pixels to the inch, a hairline will display as 1/70th of an inch wide. The same hairline, when printing on a 300 dpi laser printer will print as 1/300th of an inch wide.

Style buttons

Use the Style buttons to specify the line style you want the program to use as a border for the selected box. When you Click a Style button:

- a black box appears around the button as a highlight, and
- the program displays a text description of the highlighted style to the right of the style buttons. Your choices are:

single line	Prints a single solid line.
dashed line	Prints a single dashed line.
dotted line	Prints a single dotted line.

NOTE: All colors are fixed; they cannot be edited.

NOTE: While a box with a white fill looks no different than a box with no fill when the two boxes are standing alone, there is a major difference between them. The fill color white is a solid color and thus it can block out parts of boxes it overlaps. If you want a transparent box, toggle the Fill Color check box off (None).

NOTE: The program uses a transparent box (no fill) with a black single line border of hairline width as a default.

NOTE: Text always prints over the top of boxes, as if the boxes were on a separate, lower layer.

Verify Database dialog box

This dialog box appears when you use the Verify Database command to check the structure of the underlying database, looking for changes:

- If the underlying database is unchanged, you will get the following message:

  ```
  Database is up to date.
  ```

  Click OK to return to the Design Window.

- If the underlying database has changed, you will get the following message:

  ```
  The database file (filename) has changed.
  Proceed to fix up the report?
  ```

- If you select *Yes*, Crystal Reports adapts the database to the current version of the database.
- If you select *No*, Crystal Reports attempts to print the report without first adapting it to the current version of the database.

Edit Record Selection Formula dialog box

This dialog box appears whenever you select the Edit Record Selection Formula command. You use the dialog box to set up a record selection formula that limits the records the program uses when preparing your report. This is the same dialog box that appears when you use the Insert|Formula Field command.

NOTE: No field placement box appears when you create a selection formula; the formula is simply stored in the Crystal Reports report.

NOTE: Your record selection formula must be Boolean, that is, it must result in a Yes or No answer.

NOTE: You can not use commas in numbers you enter in a formula. Enter only the number itself.

Edit Group Selection Formula dialog box

This dialog box appears whenever you select the Report|Edit Selection Formula|Group command. Use the dialog box to edit a group selection formula that limits the groups the program uses when preparing your report.

1. You can use the full range of operators, functions, and data fields to create a formula which restricts the range of data to print.
2. Enter your formula.
3. Click Accept when finished. Crystal Reports limits the report to the groups specified.

NOTE: You can use Page Number and Record Number fields in a group selection formula via the PageNumber and RecordNumber functions.

NOTE: No field placement box appears when you create a selection formula; the formula is simply stored in the report.

NOTE: Your group selection formula must be Boolean, that is, it must result in a Yes or No answer.

NOTE: You can't use commas in numbers you enter in a formula. Enter only the number itself.

Browse Field Data dialog box

This dialog box appears when you Click the Browse Field Data button in the Insert Database Field dialog box or when you choose the Browse Field Data command.

You use the dialog box to preview the data type, length, and/or content of any field available in the Insert Database Field dialog box or in the Design Window.

- The source of the field (alias) and the field name appear as the title for the dialog box.
- The data type for the field (string, number, etc.) appears just below the title.
- A list of field values appears in the scroll box.

When you're finished reviewing the field data, click the Done button.

NOTE: A Browse Field Data button also appears in the Select Records expert. A Browse Field Data option also appears on the right mouse button pop-up menu whenever you select a report field, in the Insert Database Field dialog box, in the Formula Editor, and in other dialog boxes in which reviewing field data might be useful.

Browse and Paste Field Data dialog box

This dialog box appears when you click the Browse Field Data button in the Formula Editor or in the Select Records and Select Groups dialog boxes.

You use the dialog box to preview the data type, length, and/or content of any field and to paste a value into your formula or selection criteria.

- The source of the field (alias) and the field name appear as the title for the dialog box.
- The data type for the field (string, number, etc.) appears just below the title.
- A list of field values appears in the scroll box.

When you're finished reviewing the field data, Click the Done button or highlight a value and Click the Paste Data button to paste the value into your formula.

NOTE: If you need to select multiple values (for example, if you want to include records in which the field value is one of three values), the insertion point automatically moves to the next text box as soon as it finishes pasting data in the previous box.

NOTE: A Browse Field Data button also appears in the Select Records expert. A Browse Field Data option also appears on the right mouse button pop-up menu whenever you select a report field, in the Insert Database Field dialog box, in the Formula Editor, and in other dialog boxes in which reviewing field data might be useful.

Graphic Format dialog box

You use this dialog box to format bitmapped graphics. The dialog box contains a number of smaller boxes, buttons, and check boxes:

Cropping of original

Cropping refers to "cutting" away those portions of your graphic that you don't want to print (although you can use cropping to add white space between the graphic and the frame that surrounds it as well). Using the cropping box, specify the size of the piece you want to cut off the top, bottom, left, and/or right side of your graphic.

NOTE: All cropping activities begin at the outer edge of the graphic.

- Positive numbers cut into the graphic the amount specified.
- Negative numbers add the specified amount of white space between the outer edge of the graphic and the frame.

For an examination of the cropping process and a discussion about adding white space around a graphic, see Cropping graphics.

Scaling

Scaling refers to the length and width of a graphic as a percentage of the original length and width. For example, if the original graphic is one inch wide, that width is automatically assigned a width scaling value of 100% by the program. To double the width of the graphic using the Scaling options, you would change Scaling Width to 200% (twice the size of the original). Likewise, to reduce the width of the graphic to one half inch (half the size of the original), you would change Scaling Width to 50%. Scaling Height works in the same way.

If you want to resize the graphic as a percentage of the original height and width, enter the new scaling percentages.

NOTE: Crystal Reports stores a copy of each graphic in its original size. All scaling settings refer back to that original size. For example, if you have a graphic that was originally a four inch square and you have resized it to a two inch square, the Scaling box will show settings of Width = 50% and Height = 50%. Those percentages refer back to the original. If you want to resize the graphic again to a one inch square, you will have to enter scaling values that again refer back to the original Width and Height, not the current values. Entering values of Width = 25% and Length = 25% will reduce the original four inch square to a one inch square.

Size

Size refers to the absolute (measured) length and width of a graphic. For example, if a graphic is originally a one inch square, each of the Size settings will initially be set at one inch. To double the length and width of the graphic (to make a two inch square), reset the Size settings to two inches each. To reduce the size of the graphic to a half inch square, reset the Size settings to a half inch each.

NOTE: When you change the Size settings, Crystal Reports automatically recalculates the Scaling settings, and when you change the Scaling settings, Crystal Reports automatically recalculates the Size settings. The recalculated settings appear when you next open the Graphic Format dialog box.

Original Size

The Original Size box displays the original dimensions of the graphic (its dimensions when first inserted into the report). Sizing, scaling, and cropping don't affect this figure nor can you change it directly; it is simply provided as a reference.

Hide when printing check box

Graphics contain masses of data that take time for computers and printers to process. It follows, then, that when you have graphics in your report, the report pages print more slowly than they would without graphics. While slower printing shouldn't be a problem on your final printing, it may be a bit of an annoyance when doing multiple test prints (to window or printer) while developing your report. This check box was provided to eliminate that annoyance.

When you click the check box to activate it, Crystal Reports ignores all graphics when it prints. With the check box inactive, the program prints the graphics it finds. A typical way to use the Hide when printing option is this:

- Leave it inactive while you are placing, sizing, and cropping your graphic.
- Once you have the graphic the way you want it, Click the check box so no graphics print when you make test prints.
- When you're ready to print your final report, Click the check box off and print the report, graphics and all.

Position button

Use the Position button any time you want to reposition a graphic by specifying its absolute position within a section.

Graphic Position dialog box

You can use this dialog box to set the absolute position of a bitmapped graphic in your report. The dialog box has two settings:

Left Use Left to set the position of the left side of the graphic relative to the left edge of the section.

Top Use Top to set the position of the top of the graphic relative to the top of the section.

NOTE: All settings are in either inches or centimeters, based on your settings in the International section of the Windows' Control Panel.

Insert your new settings and click OK when finished to return to the Graphic Format dialog box.

NOTE: The numeric position of a graphic (as shown in the Graphic Position dialog box) is relative to the page margins you have set. For example, if you have set a left page margin of 1.00 inches and you place your graphic, numerically, with a* Left *setting of 0.5 inches, the graphic will print 1.5 inches in from the left edge of the paper, 0.5 inches in from the left margin.

NOTE: The settings displayed when you first call up this dialog box indicate the current position of the graphic in the section.

NOTE: A setting of Top = 0.00, Left = 0.00 positions the graphic flush in the upper left hand corner of the section.

NOTE: You can also reposition a graphic using a mouse if you want to determine its final position visually rather than "by the numbers." For complete instructions on repositioning a graphic with a mouse, see Inserting, moving, and deleting graphics.

Line Format dialog box

Use this dialog box to format a selected line. The dialog box contains three sets of formatting buttons and the standard OK, Cancel, and Help buttons.

Mailing Labels dialog box

The Mailing Labels dialog box is a powerful control panel. The dialog box enables you to print your data on virtually any labels that are commercially available for line printers (dot matrix) or page printers (laser, ink jet). Using the dialog box settings and Crystal Reports formatting commands, you can fine tune your printing so your labels come out just the way you want them. The Mailing Labels dialog box offers two ways to set up labels:

- automatically, by Avery label number, and
- manually.

Automatically, by Avery label number

To make your work easier, Crystal Reports has come with preset templates for the most popular Avery labels. Those labels are listed, by number, in the Choose Mailing Label Type scroll box.

Manually

If your label isn't on the Choose Mailing Label Type scroll list, you will need to set up the label manually using the various settings in the Mailing Labels dialog box. Select User Defined Label and then change the remaining dialog box settings to conform to the label you're going to use.

Multi Column Layout dialog box

The Multi-Column Layout dialog box enables you to set up your report in a multi-column format. That is, instead of having the data print straight down the page, you can set up multiple columns and have the data flow from column to column. You can also have your data print across then down the page, printing one record in each column, then printing a second record in each column, then a third, etc. The dialog box is divided into four smaller boxes.

Detail Size

The Detail Size box enables you to specify the dimensions (height and width) of one detail. Determine how wide you want each detail to be (based on number of characters, font size, etc.) and enter that value in the Width edit box.

Determine how high you want each detail to be (based on number of lines in the detail, font size, etc.) and enter that value in the Height edit box.

Gap Between Details

The Gap Between Details box enables you to specify the empty area (gap, gutter, etc.) you want to allow between details. Horizontal = the gaps between details going across the page, Vertical = the gaps between details going down the page.

Determine the gaps you want to allow. Enter the horizontal gap in the Horizontal edit box and enter the vertical gap in the Vertical edit box.

Printing Direction

The Printing direction box enables you to specify the path Crystal Reports follows when printing the details on a report page.

- *Across then Down* Prints details across the columns, one detail in the first column, one in the next, one in the next, etc. Then, when all the columns have a detail, the program moves down the page and prints a second detail in the first column, then in the second, etc.
- *Down then Across* Prints details down the first column, then down the second column, etc.

Number of Details

The Number of Details box displays the number of details that can print across the page and the number that can print down the page based on your settings in the Detail Size and Gap Between Details boxes and in the Printer Margins dialog box.

NOTE: The program displays only the number of complete details it can print. For example, if it determines that there is space available to print 2.75 details across the page, it displays 2 as the Across Page setting and prints only two details across the page.

Page Margins dialog box

Use this dialog box to set the page margins for your report (the white space between your report and the edge of the page) The dialog box displays default margins when it first appears. To use this dialog box, accept the default margins or enter your new margins.

- To accept the default margins, click OK and the program returns you to the Design Window.
- To change the margins, enter your new margins in the Top, Left, Bottom, and/or Right Margin edit boxes. When you change any of the default margins, the program turns the Use Default Margins check box *Off.*
- To restore changed margins to their default settings, click the Use Default Margins check box.

NOTE: Crystal Reports uses the non-printing areas established for your printer as default printer margins. Those margin settings appear in the Printer Margins dialog box. While you can set margins that fall inside the non-printing areas, parts of your report may be clipped off if you do.

Print dialog box

Use the Print dialog box to set up the specifications for your print job. The dialog box appears whenever you select the File|Print|Printer command (or click the printer icon in the Preview Window).

Edit Text Field dialog box

You use the Edit Text Field dialog box to insert or change the content of text fields. You can add text, delete text, rearrange text, or change the spelling of text in any text field in your report. Enter your text in the Enter Text edit box, and Click Accept when finished.

NOTE: You cannot copy text fields via the Windows Copy command, but you can copy freeform text using that command. That is about the only thing that you cannot do with text fields that you can do with freeform text.

Word wrap and return characters

Crystal Reports will word wrap within the space allotted if you have activated the *Print on multiple lines* option in the Format String dialog box. If you expand or narrow the field, word wrap will adjust to the available space (where possible), again, if the Print on multiple lines option is activated.

NOTE: If your text field includes return characters, Crystal Reports interprets them to provide the line breaks you specify in the text field (where possible).

NOTE: While the effective limit on text field size is 32K (including carriage return and line feed characters), it is recommended that text fields be used only for more manageable sized blocks of text.

Edit Report Title dialog box

Use the Edit Report Title dialog box to:

- change the title that appears on the title bar of your report, and
- to enter and/or edit non
- printing comments that you want to accompany the report.

Choose New Index dialog box

Use the Choose New Index dialog box to select the report file you wish to change the location of. Using the Drives, Directories (Folders), and File Name list boxes, choose the new location and click OK when finished.

Date field conditions

When you subtotal based on changes in a Date field, Crystal Reports first sorts your data by date. Then it breaks the data into groups and prints a subtotal whenever the date condition that you select is met.

When you select a Boolean field condition from the Insert Summary, Insert Group Section, or Insert Subtotal dialog box, your options are:

- *for each day* prints a subtotal whenever the date changes.
- *weekly* prints a subtotal at the change from one week to the next (a week runs from Sunday through Saturday).
- *for each two weeks* prints a subtotal every two weeks (weeks run from Sunday through Saturday).
- *for each half month* prints a subtotal twice a month.
- *for each month* prints a subtotal at the end of each month.
- *for each quarter* prints a subtotal at the end of each calendar quarter.
- *for each half year* prints a subtotal at the end of each calendar half year.
- *for each year* prints a subtotal at the end of each year.

Group Name Field command (Insert menu)

When you Select Group Name Field from the Insert menu, the Insert Group Name Field dialog box appears. You use this dialog box to select a group name from the list of groups in your report. At the end of each group listed in the dialog box there is a dash followed by a letter. The letter describes how the group is sorted.

A refers to in *ascending order*

D refers to in *descending order*

S refers to in *specified order*

O refers to in *original order*

You can use this information to help select the correct field when you are in the Insert Group Name Field dialog box.

Sorting Your Groups (overview)

There are a variety of ways you can sort groups using Crystal Reports.

- You set group sort order whenever you use Insert|Group Section, Insert|Subtotal, or Insert|Summary. Each of these commands is used to set up grouping. Specifying the order in which the groups are to appear is part of the set up process.
- You can also sort groups when creating your reports using one of the Report Creation experts.
- You sort groups when you select Report|Top N/Sort Group Expert and when you select Report|Change Group Expert.

See Also

Grouping and Sorting topics index

Creating Custom Groups

Crystal Reports enables you to create freeform groups and specify exactly the individual components of each group. For example, if you have State and City fields on your reports, it is easy to break your data down by State and, within each state, to break it down by City. But if you want your data broken down by County, and there is no County field on your report, you have no options using most report writers. With Crystal Reports, however, you can create a County group and specify the individual cities that belong to each county. The process is totally flexible and it enables you to present your data in exactly the way you want it presented, with virtually no limitations. We'll create a sample report to walk you through the process.

Assume the following sample data:

City	State	Orders
Tustin	California	1000000
Long Beach	California	2500000
Bellevue	Washington	3000000
Seattle	Washington	1000000
Mission Viejo	California	2000000
Spokane	Washington	100000
Los Angeles	California	7000000
Fresno	California	50000
Garden Grove	California	1000000
Austin	Texas	100000
Cerritos	California	2000000
Atlanta	Georgia	150000
Bellingham	Washington	50000

This sample data can be found in *specify.dbf*, a sample database that was installed when you installed Crystal Reports. We're going to use this sample data to create the following report:

Orders by state and by county within each state

California		
Los Angeles County		
Cerritos	California	2 000 000
Long Beach	California	2 500 000
Los Angeles	California	7 000 000
	Total for County	**11 500 000**
Orange County		
Garden Grove	California	1 000 000
Mission Viejo	California	2 000 000
Tustin	California	1 000 000
	Total for County	**4 000 000**
	Total California	**15 500 000**
Washington		
King County		
Bellevue	Washington	3 000 000
Seattle	Washington	1 000 000
	Total for County	**4 000 000**
	Total Washington	**4 000 000**

This report, *specify.rpt*, was also installed when you installed Crystal Reports. You can open that report using Crystal Reports and study the way it was put together or you can follow the step by step process to build the report from scratch.

Building the report

1. You begin by opening Crystal Reports, selecting specify.dbf as your data source, and placing the *specify.City*, *specify.State*, and *specify.Orders* field in the Details section of your report.
2. The first thing we'll do is break our data into State groups and subtotal the values in each group. Click somewhere outside the field boxes to deselect them and then click the *specify.Orders* field box to select it.
3. Select Insert|Subtotal. The Insert Subtotal dialog box appears.
4. Click the arrow on the top drop down box and select *specify.STATE* as your sort and group by field.
5. If *in ascending order* isn't set as your sort order, click the arrow on the bottom drop down box and select *in ascending order*, and click OK when finished. Were you to print or preview the program at this point, your data would be broken into State groups and each State group would be subtotaled.
6. We want our report to show data only for two states (California and Washington). In California we want to show data for only two counties (Los Angeles and Orange. In Washington we want to show data for only one county (King). We'll use the specified sort order facility to set this up.
7. Since we'll be moving city data into counties and then subtotaling the orders in each county, we'll set up a subtotal based on the values in the *specify.CITY* field. If the specify.Orders field isn't still selected, select it and then select Insert|Subtotal again. The Insert Subtotal dialog box appears with the Group #1: Specify State group showing in the drop down box.
8. Click the arrow on the drop down box and select *specify.CITY* from the list that appears. A second drop down box, a sort order box, appears below the first drop down box.
9. Click the arrow on the bottom drop down box and select *in specified order* from the list that appears. Some additional boxes and buttons appear. These boxes and buttons are your gateway to setting up custom groups and sort orders.
10. The first county group we want to set up is Los Angeles County and we need to define what's in that group so click the New Named Group button. The Define Named Group dialog box appears.
11. Type Los Angeles County in the Group Name box to assign that name to the group we're about to

create.

12. Now we'll set up the group using the boxes immediately beneath the Group Name box. The selection criteria in the first two drop down boxes is set to *is equal to.* That would be what we would want if we were going to add only a single city to the group. But we're going to add several cities, so click the arrow on the second drop down box and change the setting from *equal to* to *one of*. A large list box appears below the drop down boxes. As you select the cities for the group, the city names will appear in that list box.
13. To assign cities to the group, click the arrow on the third drop down box. A list of city values from the *specify.CITY* field appears. Click *Los Angeles* to select it. The name *Los Angeles* appears in the bottom list box to indicate that it has been selected.
14. Long Beach and Cerritos are also in Los Angeles county so click the arrow on the right drop down box and select each of those cities as well. You will have to click the arrow and select two times to do this. When you're finished, Cerritos, Long Beach, and Los Angeles should appear in the list box. These will be the three members of the Los Angeles County group.
15. Click OK when finished and the program returns you to the Insert Subtotal dialog box with the group name *Los Angeles County* appearing in the bottom list box. Congratulations, you've set up and named your first custom group.

NOTE: If you want to create a more complex query, for example, is one of Cerritos, Long Beach, and Los Angeles or is starting with San, set up the first part of the query is one of Cerritos, Long Beach, and Los Angeles as shown in Steps 10-15, then click the New tab. The program will create a second tab called OR specify.CITY. You can then add the next part of your query starting with San. Click the new tab again if you need to expand the query even further. Using this method, you can create extremely sophisticated groupings (file.lastname is one of Johnson, Jones, and Brown OR is like Sm*th* OR is starting with Mac OR is starting with Mc OR is greater than Wayne.)

16. Follow the procedure outlined in Steps 10 through 15 to set up two additional groups using this data.

NOTE: Don't close the Insert|Subtotal dialog box when you're finished; we're not done with it yet.

Group Name	**Group Members**
Orange County	Tustin
	Mission Viejo
	Garden Grove

Group Name	**Group Members**
King County	Bellevue
	Seattle

NOTE: The groups will appear in the list box in the order you created them. If you don't like that order for your report, highlight a group you want to move and use the arrows to the right of the list box to move it up or down in the list.

17. Since we only want the specified groups to appear in the report, we have to dispose of the other groups that we don't want to appear, so click the Others button. The Others dialog box appears.
18. Select Discard All Others and click the Done button when finished to return to the Insert Subtotal dialog box. This tells the program not to include any records or groups that you haven't specified.

NOTE: If you want to change the composition of any group before you leave the Insert Subtotal dialog box, highlight the group name and click the Edit Named Group button. The program calls up the Define Named group dialog box with the selected group active and ready for you to modify. Once you've closed the Insert Subtotal dialog box, you must use Report|Change Group Expert to

make changes to any of the groups.

19. Since we're finished with our work in the Insert Subtotal dialog box, click OK to return to the Design Window. The groups and records you have specified appear in that window.
20. Now we need to add some descriptive text. We'll begin by adding a live group header for each State group. To do this, select Insert|Database Field. When the Insert Database Field dialog box appears, drag the State field into the *#1: STATE* section. The state name will appear above each state group when you print or preview the report.
21. Change the font for the State Group Header field to Arial 12 point Bold Italic using the Format|Font command.
22. To put in names for each of the county groups, we'll use the Insert|Group Name Field command. When you select this command, the Insert Group Name Field dialog box appears.
23. Two groups appear on the list: *Group 1, specify.STATE A* and *Group 2, specify.CITY S.* The A and S refer to the sort order used for each group. Since we specified groups based on the City field, the group we want to use the names from is the *Group 2, specify.CITY S*. group. Select that group and a rectangular placement box appears. Drag that box into position in the *#2: CITY* section, immediately under the State group header and change the font to Arial 12 point Bold.
24. To identify each of the subtotals, we'll begin by identifying the County subtotals.
25. Select Insert|Text Field and type *Total for County* in the edit box in the Edit Text Field dialog box when it appears and click Accept when finished.
26. When the rectangular placement box appears, drag it into position just to the left of the county subtotals (in the Group Footer *#2: City* section).
27. Change the font to Arial 12 point Bold.
28. Now we'll identify the State subtotal and we'll do it using a formula. Select Insert|Formula Field, type the formula name *State total* in the Insert Formula dialog box when it appears, and enter the following formula in the Formula Editor when it appears:

```
"Total " + {specify.STATE}
```

29. Click Accept and when the rectangular placement box appears, drag it into position just to the left of the State subtotals (in the Group Footer *#1:State* section).
30. Change the font to Arial 12 point Bold.
31. Align the headings so they look the way you want them.
32. To add a report title, select Insert|Text Field, type in *Orders by state and by county within each state* in the Edit Text Field dialog box, click Accept to accept it, and place the field in the Page Header section of your report.
33. Change the title font to Arial 14 point Bold Italic.
34. Resize the title text field box so it's as wide as the report, select Format|Field, and set the alignment to Centered. This centers the title nicely above your report.
35. One final step is needed to complete the report. The records in each group don't yet appear in alphabetical order. To fix that, select Report|Sort Records. The Record Sort Order dialog box appears.
36. Since we want the City records to appear in alphabetical order (Ascending), select specify.CITY from the Report Fields box and click Add to move it to the Sort Fields box. Make sure the sort order is set to Ascending and click OK when finished. When you print or preview your report, the records in each report group will appear in alphabetical order.

That's all there is to creating a powerful report with custom groups and a custom sort order using Crystal Reports.

See Also

Grouping and Sorting topics index

Custom Grouping And Sorting

Most of the time you sort and group your data based on the values in some field in your report. For example, if you have a customer list and you want to sort and group it by state, the program first sorts the list by state and then breaks the list into state groups whenever the value in the State field changes. Sometimes, however, you may not want to group based on the values found in one of the fields on your report.

- Your report may not contain the field you want to group on. For example, your report contains a City field and a State field but no County field but you want to group by county.
- Your report may contain the field you want to group on, but you're not happy with the grouping based on the values in that field. For example, you have a Color field on your report that includes specific color names (Logan Green, Sky Blue, Emerald Green, Navy Blue, etc.) but you want all "flavors" of each color to appear as a single group (Greens, Blues, Reds, etc.). In this case you can build custom groups and manually assign the records you want to be in each group.
- Your report may contain the field you want to group on, but you want to select specific values or ranges of values for each group. For example, you might want one group to contain records where gross sales are less than a certain value, a second group where gross sales are greater than a certain value, and a final group where gross sales fall between two values. In this case, you can build your groups using the same range selection facilities that are available to you for building record selection queries.

Crystal Reports provides Specified Order sorting as a solution to these custom sorting and grouping challenges. Specified sorting enables you to create the groups you want to appear on your report and the records that each group contains. Your only real limitation is that a record can be assigned to only one group.

To use Specified Order sorting, you select in specified order as your sort option whenever the program provides you with that option. The program gives you the in specified order option whenever you create groups using Insert|Group Section, Insert|Subtotal, or Insert|Summary, whenever you create groups while building a report using one of the Report Creation experts, or whenever you select Report|Change Group Expert.

See Also

Creating Custom Groups

Grouping and Sorting topics index

Sorting and Grouping Topics index

Click on a topic of your choice for more information:

Sorting

Change Group Expert

Custom Grouping and Sorting

How data types are sorted

Insert Group Section Command

Insert Subtotal Command

Insert Summary Command

Multiple field sorts

Record Sort Order dialog box

Report|Record Sort Order

Sorting Your Groups

Top N/Sort Group Expert

Using Group Sort, Record Sort Order together

Grouping

Change Group Expert

Creating Custom Groups

Edit Group Section (sections) dialog box

Edit Group Sections (edit) dialog box

How to identify the "top" groups

Insert Group Name Field Command

Insert Group Section Command (insert menu)

Insert Group Section Command

Insert Group Section dialog box

Insert Subtotal Command

Insert Summary Command

Top N/Sort Group Expert

Edit Group Section (sections) dialog box

The Edit Group Section (sections) dialog box appears when you select the Report|Change Group Expert command. Use this dialog box to select the group section you wish to edit.

See Also

Grouping and Sorting topics index

Edit Group Section (edit) dialog box

The Edit Group Section (edit) dialog box appears once you have selected a group section to edit from the Edit Group Section (sections) dialog box. Use this dialog box to change the sorting and grouping specifications (sort and group by field, sort direction, etc.) for any of the groups on your report.

See Also

Grouping and Sorting topics index

Insert Group Section dialog box

You use this dialog box to set the conditions under which data is to be grouped.

The top scroll box contains a list of the fields and formulas being used in the report. Click the scroll arrow to reveal the list, and select from that list the field that you want the program to use for triggering groups.

Fields you have previously specified grouping for will not be available for editing in this box. To edit a group, you must choose Report|Change Group Expert from the menu and select the group you want to edit from the dialog box that appears.

- The program will first sort report data based on this field.
- Then it will go down the report and group the data whenever the value in the field changes.

If you select a date or Boolean field, the program gives you the ability to further narrow your selection. See *Date and Boolean* conditions below.

The next scroll box lists the four sort direction options. The default option is *in ascending order*. If you want to change the sort direction, click the scroll arrow to reveal the options and make your selection from the list.

NOTE: If you select in specified order as a sort direction, there are additional options to the dialog box.

Date and Boolean conditions

When you group based on changes in a Date or Boolean field, Crystal Reports displays another scroll box at the bottom of the dialog box. This scroll box enables you to further define your grouping specification.

- *Date Conditions scroll box* contains a list of date ranges that define typical grouping periods. When you select a date condition, Crystal Reports first sorts your data by date. Then it breaks the data into groups and prints a subtotal whenever the date condition that you select is met. Click the scroll arrow to reveal the list of Date conditions.
- *Boolean Conditions scroll box* contains a list of Boolean conditions that define typical grouping situations. When you select a date condition, the program first sorts your data. Then it breaks the data into groups whenever the Boolean condition you select is met. Click the scroll arrow to reveal the list of Boolean conditions.

When you select OK, Crystal Reports creates a group section in the Design Window for this group, and groups your data according to the conditions you have specified.

NOTE: Once you have set up your group section and closed this dialog box, if you then want to modify your group section you must use Edit|Group Section to do so. If you call up Insert|Group Section again, you will only be able to review your settings, not change them.

See Also

Grouping and Sorting topics index

Sort Direction Options

Sort Direction Options appear in several dialog boxes: the Edit Group Section dialog box, the Insert Group Section dialog box, the Insert Summary dialog box and the Insert Subtotal dialog box.

The Options Are:

in ascending order

in descending order

in specified order

in original order

Ascending order

Ascending is (1 to 9, A to Z) Select ascending order to have your records grouped in this order.

Descending order

Descending is (9 to 1, Z to A) Select descending order to have your records grouped in this order.

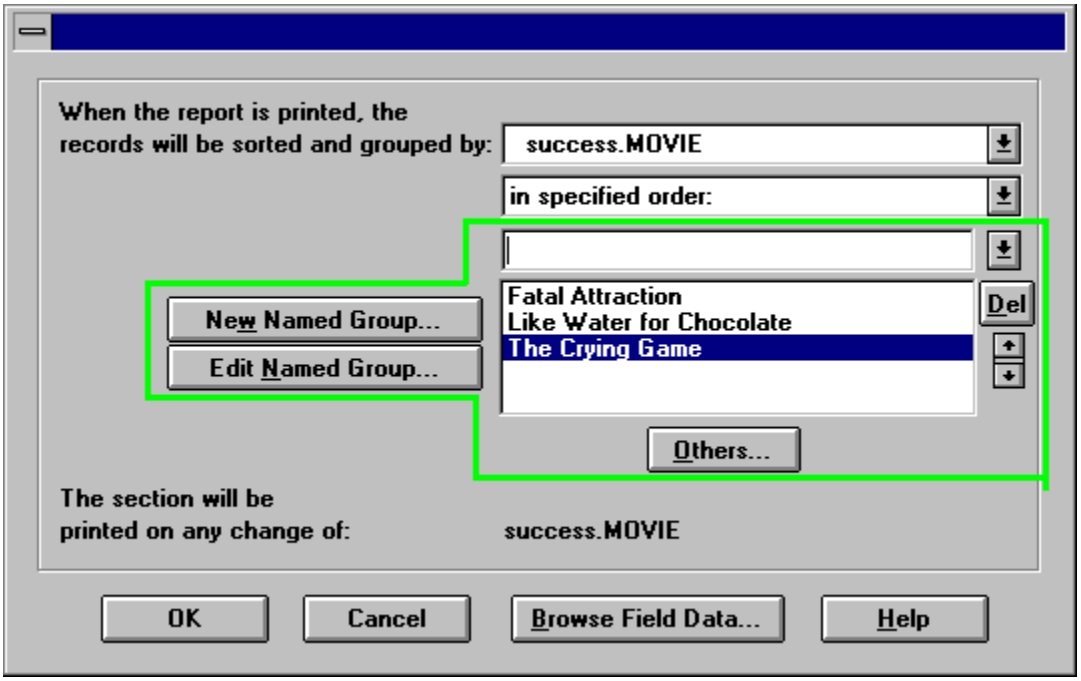
Original order

Original order is how the data is currently organized in the database. It may be ascending, or descending or in no particular order. To look at the order, just click the Browse Field Data button to open the Browse Field Data dialog box. Click Done to return to the Insert Group Section dialog box.

In Specified Order

When you select *in Specified order* as your sort direction option, additional options will appear in the dialog box that you can use for setting up your groups.. **The additional dialog box options that appear are inside the green square on the picture below**.

Click on the options inside the green square on the picture below to learn more.



See Also

Grouping and Sorting topics index

The Drop Down edit selection box

The Drop Down edit selection box lists the values in the field you have selected as your sort and group by field. If you want to specify a group based on a single value from this field, Click the arrow and select the field from the list that appears.

If you want to set up additional groups based on individual field values, repeat the process until you have added all the groups you want.

The Selection scroll box

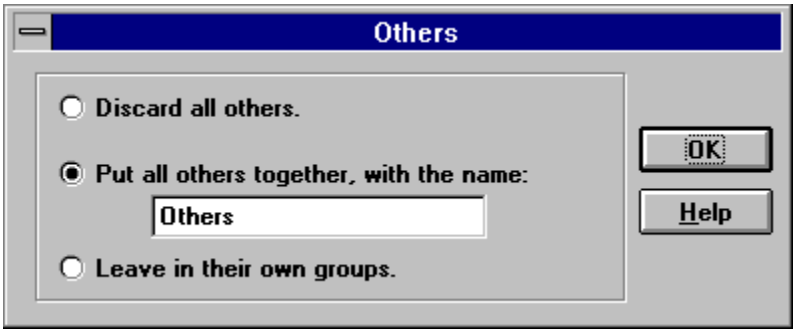
The Selection scroll box displays groups you have selected from the Drop Down edit selection box. When you have added more groups than will fit in the box, a scroll bar appears. Also, there is an up arrow and a down arrow on the right side of the selection scroll box. To change the position of any group in the list, click the group to highlight it and then use these arrows to move it up or down. When a group name is highlighted, you can delete it by clicking the Delete Button, or edit the group value by clicking the Edit Named Group button.

The Arrow Buttons

There is an up arrow and a down arrow on the right side of the selection scroll box. To change the position of any group in the list, Click the group to highlight it and then use these arrows to move it up or down.

Others dialog box

When you Click the Others button, the Others dialog box appears. The dialog box has three radio buttons and a Name edit box.



- When you select the *Discard all others* radio button, the program will not print any groups not specified in the sort order list box. If you summarize groups, values in the discarded groups will not be summarized nor will they be included in any grand total if you have one.
- When you select the *Put all others together, with the name* radio button, the program will create a group that will lump together all groups not specified in the sort order list box. This group will appear after the specified groups. If you summarize groups, the values in the group will be summarized and will be included in any grand total. You can specify the name of the group using the Name edit box.
- Use the *Name edit box* to specify the group name for the Others group. By default, the group will be named Others. To specify a group name, highlight the name Others, and enter the desired group name in the edit box.
- When you select the *Leave in their own groups* radio button, the program will print your specified groups in the order you specified them, and it will then group and print the remaining groups normally. If you summarized groups, the values in the group will be summarized and will be included in any grand total.

Click the OK button to accept the option you have selected, and return to dialog box you were in when you Clicked the Others button.

Delete button

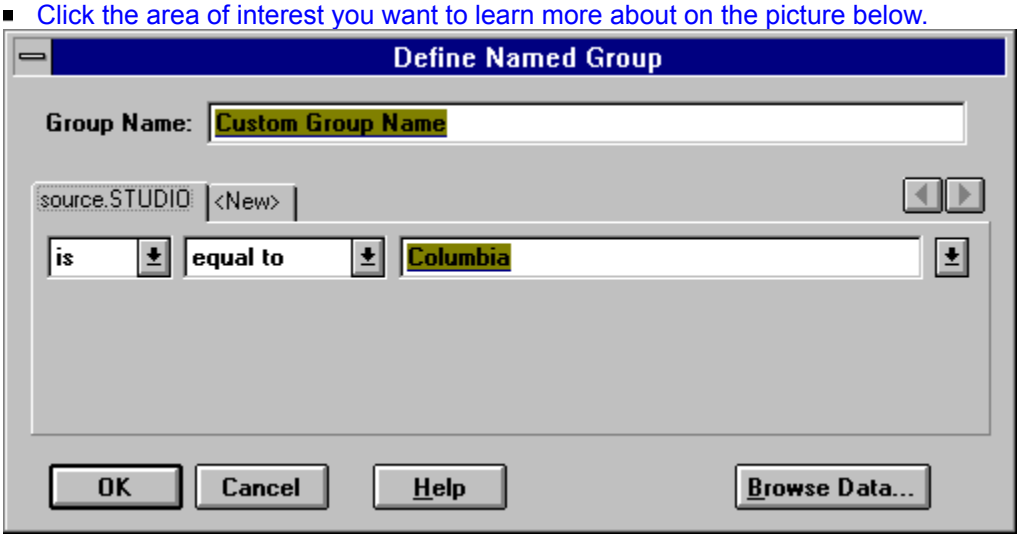
Click the Delete button to remove a group name from the Selections scroll box. The Delete button is active when a group name is selected in the Selections scroll box.

Define Named Group dialog box (for new groups)

The New Named Group button appears in four dialog boxes: Edit Group Section, Insert Group Section, Insert Summary, and Insert Subtotal. When you select the New Named Group button from one of these dialog boxes, the Define Named Group dialog box appears.

Use this dialog box to set up custom groups and custom group names. You can edit the groups later by Clicking on the Edit Named Group button.

- Click the area of interest you want to learn more about on the picture below.



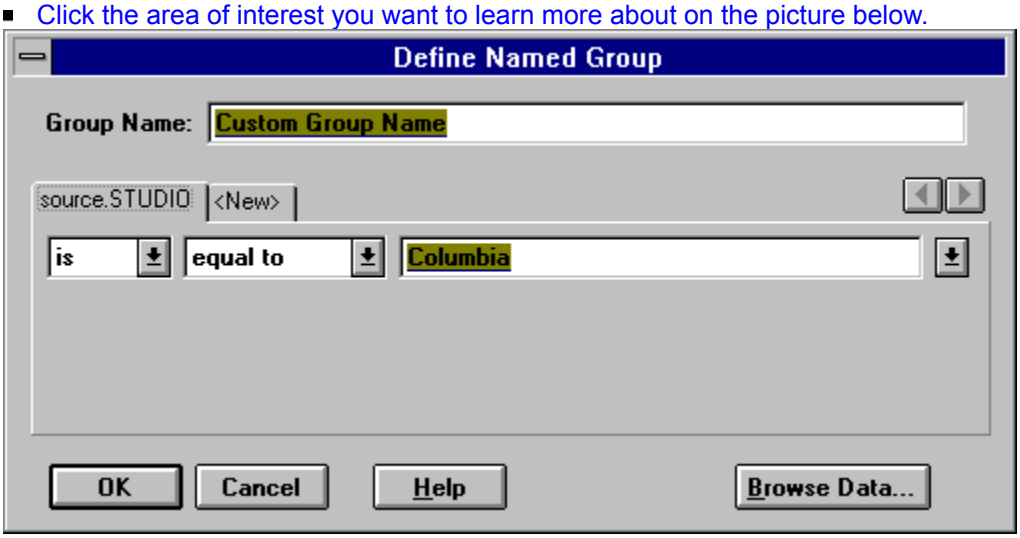
See Also

Grouping and Sorting topics index

Define Named Groups dialog box (for existing groups)

The Edit Named Group button appears in four dialog boxes: Edit Group Section, Insert Group Section, Insert Summary, and Insert Subtotal. When you select the Edit Named Group button from one of these dialog boxes, the Define Named Group dialog box appears. **You use this dialog box to edit your existing custom groups and custom group names**. To edit a group, you simply select it from the list of groups in the box to the right of the Edit Named Group button, and then Click the Edit Named Group button. The dialog box pictured below will appear.

- Click the area of interest you want to learn more about on the picture below.



See Also

Grouping and Sorting topics index

The Define Named Group dialog box

The Define Named Group dialog box appears when you click either the New Named Group button, or the Edit Named Group button. The dialog boxes appear to be identical. However, one is used for establishing new groups, and the other is used for editing the groups you have already established.

The Tabs

The tabs display the field whose values you are using to create the group. Tabs are all related to the same group, and the tabs each relate to a specific portion of the query used to create the group. When you click a tab, the appropriate portion of the query appears in the dialog box. If you want to expand the query, click the **New** tab.

- To edit a named group, click the tab for that group.
- To create a new group, click the *New* tab.

See Also

Creating Custom Groups

The *is, is not* drop down box

You select is, or is not from this drop down list box. As an example, you could search for a value that is equal to a specified value, or that is not equal to a specified value. Use the is and is not selectors to help make your searches more specific.

The Drop Down Operators Box

Use this box to select a condition on which to base your groups. The following conditions appear whenever you use Define Named Group with groups based on values in number, currency, string, or memo fields. These conditions can also be reversed in some circumstances by selecting the **is not** condition from the drop down list box.

is any value

Include a group based on each value in the selected field.

is equal to

```
company.State is equal to CA
```

«Build a group that contains only records that have CA(lifornia) as the State field value.»

is one of

```
company.State is one of AZ, CA, or CO
```

«Build a group that contains only records that have a value in the State field that matches either AZ, CA or CO»

is less than

```
company.State is less than AZ
```

« Build a group that contains only records that have a value in the State field less than Arizona (alphabetically)»

```
company.LastYrSales is less than 100000.00
```

«Build a group that contains only records that have a value in the LastYrSales field less than 100000 (numerically)»

NOTE: If you check the or equal to box, the group will contain values that are less than or equal to.

is greater than

```
company.State is greater than AZ
```

«Build a group that contains only records that have a value in the State field greater than Arizona (alphabetically)»

```
company.LastYrSales is greater than 100000.00
```

«Build a group that contains only records that have a value in the LastYrSales field greater than 100000 (numerically)»

NOTE: If you check the or equal to box, the group will contain values that are greater than or equal to.

is between

```
company.State is between AZ and CA
```

«Build a group that contains only records that have a value in the State field that falls between AZ and CA (alphabetically) or that matches either.»

```
company.LastYrSales is between 10000 and 100000
```

«Build a group that contains only records that have a value in the LastYrSales field that falls between 10000 and 100000 (numerically) or that matches either.»

is starting with

```
company.State is starting with C
```

«Build a group that contains only records that have a value in the State field that starts with the letter "C".»

```
company.Coname is starting with Cu
```

«Build a group that contains only records that have a value in the Coname field that starts with the letters "Cu"»

like

```
company.Coname is like C?m*
```

«Build a group that contains only records that have a value in the Coname field that matches the pattern "C?m*", where "?" represents any single character and "*" represents any number of characters. This would call up "Computer" and "Camera" but not "Coombs".»

```
company.Coname is like C*m*
```

«Build a group that contains only records that have a value in the Coname field that matches the pattern "C*m*" where "*" represents any number of characters. This would call up "Computer" and "Camera" as well as "Coombs".»

Selection drop down box

Use the Drop Down edit selection box to select a value to add to the group after you have selected an operator. If you know the value you can type it in. If you do not know the value, Click the down arrow on the right side of the edit box, and then select the value from the list that drops down.

Group Name Edit Box

Use this box if you want to specify a new name for the group you are specifying or editing. For example, if you have a group that contains the earnings from Paramount, Universal, and Tristar movie studios, you may call that group **Paramount, Universal, and Tristar**. To do this, enter the new title into the Group Name Edit Box.

The Browse Data button

Click the Browse Data button to browse the field data. You can select data from the dialog box that opens and Click the Paste button to paste data to the Selection drop down box. Click the Done button to return to the Define Named Group dialog box.

Field Format (strings) dialog box

This dialog box appears whenever you click the String button in the Field Format section of the Options dialog box. Use the Field Format (strings) dialog box to set default formatting specifications for string fields that appear on your report. You use this dialog box to suppress printing of duplicate string field data and to align values within string fields. You also use it as a gateway to the Format String dialog box which contains additional formatting options.

Field Format (numbers) dialog box

This dialog box appears whenever you click the Number button in the Field Format section of the Options dialog box. Use the Field Format (numbers) dialog box to set default formatting specifications for number fields that appear on your report. Specifically, you use this dialog box:

- to suppress printing of duplicate number field data,
- to set number fields to conform to Windows' default format, and
- to call up the Format Number dialog box which contains additional formatting options.

Field Format (currency) dialog box

This dialog box appears whenever you click the Currency button in the Field Format section of the Options dialog box. Use the Field Format (currency) dialog box to set default formatting specifications for currency fields that appear on your report. Specifically, you use this dialog box:

- to suppress printing of duplicate currency field data,
- to set currency fields to conform to Windows' default format, and
- to call up the Format Currency dialog box which contains additional formatting options.

Field Format (date) dialog box

This dialog box appears whenever you click the Date button in the Field Format section of the Options dialog box. Use the Field Format (date) dialog box to set default formatting specifications for date fields that appear on your report. Specifically, you use this dialog box:

- to suppress printing of duplicate date field data,
- to set date fields to conform to Windows' default format, and
- to call up the Format Date dialog box which contains additional formatting options.

Field Format (Boolean) dialog box

This dialog box appears whenever you click the Boolean button in the Field Format section of the Options dialog box. Use the Field Format (Boolean) dialog box to set default formatting specifications for Boolean fields that appear on your report. Specifically, you use this dialog box to suppress printing of duplicate Boolean field data and to align Boolean fields. You also use it as a gateway to the Format Boolean dialog box which contains additional formatting options.

Delete Section dialog box

The Delete Section dialog box appears when you select the Edit|Delete Section command. Use this dialog box to select the group section you wish to delete.

Show/Hide Sections dialog box

The Show/Hide Sections dialog box is available through two methods, the right mouse button menu and the Edit|Show/Hide Sections command:

- When you click a report section in the gray area at the left of the Design Window and then click the right mouse button, the Show/Hide Sections option appears on the pop up menu. Select this option and the Show/Hide Sections dialog box appears, or
- From the Edit menu, select the Show/Hide Sections command. The Show/Hide Sections dialog box appears.

NOTE: The Status Bar displays the name of the section the cursor is in when the cursor is in the gray area to the left of the Design Window. This includes hidden sections.

Select Records Expert dialog box

NOTE: Although this dialog box is called* Select Records Expert*, it is used for selecting groups when you are basing your selection on a group field.

When you select Report|Select Records Expert, the Choose Field dialog box appears. When you select a group field from this dialog box and Click OK, the Select Records Expert dialog box appears.

The dialog box enables you to select the groups you want to include in your report (if you don't want them all included) without the need to understand Crystal Reports formula language. When you set up your selection criteria using this dialog box, Crystal Reports automatically generates a group selection formula based on your responses to dialog box questions.

NOTE: The Select Groups option allows you to set up reasonably complex selection criteria, but it does not have the flexibility of Report|Edit Group Selection Formula.

Compile Report (program item) dialog box

This dialog box enables you to create an icon (if you wish) for your executable (compiled) report and to create or specify an existing program group (or folder in Windows 95) for that icon. It also enables you to call up the Distribute Reports expert after you have compiled the reports so you can make certain you distribute all the necessary files with your report.

If you want Crystal Reports to create a program item (icon), Click the Yes button under **Do you want to create a program item for the report in the program manager?**. If you don't want the program to create a program item, Click the No button.

If you want the program item to appear in a program group other than the default group, type the name of the new group over the existing program group name in the **Program Group** edit box.

- If the program group you specify already exists, the program will add the icon to that group.
- If the program group you specify does not exist, Crystal Reports will create it for you.

NOTE: This process differs in Windows 95. The icon will appear as a shortcut in the Taskbar | Start option, in the folder you choose.

If you want the program to call up the Distribute Reports expert when it's done compiling, Click the Yes button under **Do you want to distribute the report right after compilation?** If you don't want the Distribute Reports expert to appear, Click the No button.

Change Position command ▪ right mouse button menu

This command takes you to the Graphic Position dialog box. You can use that dialog box to set the absolute position of the selected bitmapped graphic in your report.

Cross-Tab Group Options dialog box

The Cross-Tab Group Option dialog box enables you to modify group options for any grouped field used in the cross

-tab report.

The dialog box displays the name of the sort and group by field in the following format:

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

In addition to the text, the dialog box contains a sort order scroll box, a condition scroll box, and, if your sort and group by field is a date field, a date used scroll box.

Insert From View dialog box

The Insert From View dialog box appears along with the Design Window when you select Dictionaries as your data source and then you select the View that you want to work with. For your convenience, this dialog box remains on screen until you Click the Done button.

The Insert From View dialog box is a comprehensive data source. Properly structured, it should contain all the tables and graphics you need for building your reports. You select the type of data you want using the buttons along the left side of the dialog box. Then you select the specific data that you want from the list that appears in the list box and place it on your report. Using Dictionaries, you can build your reports in a very short time.

NOTE: For information on building new reports using report templates stored in Dictionaries, see Report Templates.

Select View dialog box

This dialog box appears when you select a data dictionary as your data source for a new report and the dictionary contains multiple views. The dialog box lists all of the views in the selected dictionary.

- Highlight the view you want and click OK to select it.

Create New Report dialog box (the Report Gallery)

The Report Gallery is a dialog box that lets you select graphically:

- the kind of report you want to create, and
- the data source you want to use when creating the report.

NOTE: Choose one of the Expert report types to get step by step instructions on creating the specified report. When the appropriate Expert dialog box appears, follow the steps outlined in the dialog box.

If you want to simply build a new report based upon one that already exists, Click the Another Report button. The program will make a duplicate of the original report, which you can modify however you please to create your new report.

Select the Another Report button if you want to use an existing report and data as a template for a new report.

If you want to build a report from scratch, Click the Custom button in the Report Gallery. Several Report Type and Data source buttons will appear at the bottom of the Report Gallery; Click the buttons that are appropriate to your needs.

Report Gallery options

The Choose Report Type and Data Type options box contains a number of buttons:

Choose Report Type buttons

Select the Custom Report button if you want to create a standard report in a row/column or data block format. Use this button to create any kind of report other than a Cross-tab report or mailing labels.

Select the Custom Cross-Tab button if you want to create a Cross-Tab report.

Select the Custom Mail Label button if you want to create mailing labels or label type items.

Choose Data Source buttons

Select the Data File button if you want to activate one or more databases for use in your report.

The SQL Table button lets you open any of the wide variety of SQL data sources that the program supports.

Select the Dictionary button if you want to get your data from an existing Data Dictionary view.

The program will take you to a dialog box allowing you to select the data you need.

NOTE: The Report Gallery will not appear if you use File|New to create your report. The Report Gallery will only appear when you Click on the New Report button and you have the Use Report Gallery for new reports check box toggled on under File|Options on the New Report tab.

NOTE: You can toggle the Report Gallery off via File|Options on the New Report tab. If you toggle it off, you select your report type and data source using non-graphical dialog boxes.

Select Report Template dialog box

The Select Report Template dialog box allows you to create a new report using a report template that has been included with a data dictionary file.

Cross-Tab dialog box

The Cross-Tab dialog box appears when you select File|New|Custom Cross-Tab. This dialog box lets you create a Cross-Tab report quickly and easily. Cross-Tab are reports that present summarized data in a row versus column structure similar to that of a spreadsheet. These reports are easy to read, and they are particularly useful for making rapid comparisons or identifying trends.

Show SQL Query dialog box

The Show SQL Query dialog box displays the SQL query Crystal Reports is sending to your SQL server. This dialog box can be used to edit the SQL Query.

Choose SQL Table dialog box

The Choose SQL Table dialog box works in a similar manner to the Choose Database File dialog box. You use the Choose SQL Table dialog box to select the table you want to activate for use in your report.

Log On Server dialog box

This dialog box enables you to select the type of SQL server you want to log on to. The Server Type box lists the various SQL server types that are available to your system. Select the server type that you want to log on to and Click OK when finished.

See Also

Logging onto a SQL server

Log On Server Command (Database menu)

Log Off Server dialog box

The Log Off Server dialog box enables you to select the server(s) from which you want to log off. The Opened SQL Servers box lists the SQL servers that are currently activated.

See Also

Log Off Server command (Database menu)

SQL Options dialog box

Use the SQL Options dialog box to set up the SQL configuration that best matches your needs.

Paste Special dialog box

The Paste Special dialog box allows you to paste an OLE object into your report from the Clipboard. If you have already copied an OLE object to the clipboard, use the following options to paste the object in your report:

- If you want to embed the object in your report, Click the Paste option button.
- If you want to insert a link to an object, Click the Paste Link option button.
- In the As type list box, select the type of object you want to link or embed. For example:
- if you want to paste the ARCHES bitmap object from the Paintbrush application, select Paintbrush Picture Object.
- if you want to paste a picture of your object (a picture that can't be edited via OLE), select Metafile or Bitmap.
- If you want the program to use an icon to represent the object in your report, click the Display As Icon check box to turn it on.

When you are finished selecting your options in the Paste Special dialog box, Click OK and a placement rectangle appears in the Design Window. Using the mouse, position the rectangle where you want the object to appear in your report and Click the left mouse button to place it. The object is now embedded (or linked) in your report.

See Also

Edit|Object

Edit|Links

Insert|Object

Print Date dialog box

The Print Date dialog box lets you can change the date that appears in a PrintDate field that appears in your report. This allows you, for example, to post-date a report that you need to print prior to the date it will be reviewed in a meeting or conference.

Crystal Library dialog box

Crystal Services maintains a bulletin board for technical support. Using the bulletin board you can find information on a wide range of topics including updated technical information, fixes for minor problems that users have reported, and comments and advice from Crystal support technicians. We're constantly adding new files to help keep Crystal Reports the most efficient, sophisticated, but easy to use report writer available. You can download any of the bulletin board files for your own use.

Crystal Library boosts your efficiency when using the Crystal bulletin board. It allows you to download a list of files available on the bulletin board so you can review them at your leisure, offline, and select the files of interest. When you're finished, use the Crystal Library command to call the bulletin board and download the files automatically.

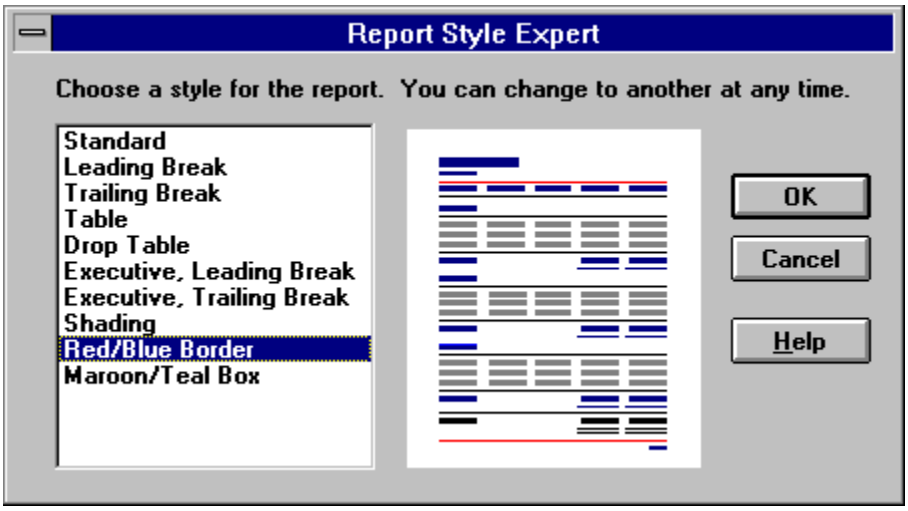
Report Style dialog box

This dialog box appears when you select Report Style Expert from the Format menu. You use this dialog box to select a report style and apply it to your report.

When you select a report style, the preview box on the right side of the dialog box changes to show you what the report style will look like.

In the picture below, the **Red/Blue Border** style is selected, and an example of that style is displayed on the right side of the dialog box.

Simply select the report style and Click OK. You can change the report style at any time by selecting Report Style Expert from the Format menu while the report is open and changing the style in the dialog box. Click OK to apply the new style



Formulas in Action Index

The following is a listing of complex formulas created to illustrate the use of various operators and functions. Each formula topic includes:

- a report scenario that describes a real
world reporting need,
- a formula that fills the need described in the scenario, and
- a dissection of the formula so you can understand the role of each of the functions and operators.

The names of the operators and functions used in the formula follow each numbered formula topic.

Click on a topic from the list below to review the formula:

Formula 1 (sales management, determining reps closest to hitting quota)

Abs(x), (■) Subtract, (/) Divide, (*) Multiply, ToText, Concatenate (+)

Formula 2 (form letter/extracting first purchase date from customer number, used date to calculate # of years as customer, and use result to customize letter)

If■Then
■Else, Subscript [], Less Than (<), Concatenate (+), Make range (to), ToNumber, Subtract (
■), ToText, Parentheses

Formula 3 (inventory analysis based on extracting inventory data from codes imbedded in item numbers)

Concatenate, ToText, ToNumber, Multiply(*), Make Range, Subscript, Parentheses ()

Formula 4 (sales compensation, calculating commissions, flag commissions that exceed certain amount)

Nested formulas, If■Then
■Else, (
■) Subtract, (*) Multiply, (>) Greater than, (>=) Greater than or equal, Sum, Parentheses

Formula 5 (form letter, soliciting orders against available credit line)

Nested formulas, If■Then
■Else, Subtract (
■), Not equal to (<>), Less than (<), Concatenate (+), ToText, To dollar ($), Negate (
■()), Parentheses

Formula 6 (calculating one value as percent of another, flag percentages outside range, disregard statistically insignificant percentages)

If■Then
■Else, Greater than (>), Percentage (%), Greater than or equal (>=), Boolean Operator And, Boolean Operator Or, Parentheses

Formula 7 (sales compensation, selecting fixed bonus or calculated commission, whichever is higher)

If■Then
■Else, Subtract (
■), Greater than (>), Maximum, Multiply (*), Parentheses

Formula 8 (purchasing, determining quantity to order based on average sales during rolling quarter)

If■Then
■Else, Less than (<), Negate (
■), Add (+), Round, Average([array]), Parentheses

Formula 9 (retail, calculating mail order sales tax based on customer ZIP or Postal code)

Nested If■Then
■Else expressions, NumericText, Subscript, Make Range, Equal to (=), ToNumber, In Range, Parentheses

Formula 10 (staff scheduling, flagging weekend incoming calls)

If■Then

■Else, Not, DayOfWeek, Make range (to), In range (in), Parentheses ()

Formula 11 (calculating a contribution based on face value of invoice, and then selecting calculated value or agreed upon minimum)

If■Then

■Else, Subscript [], Not equal to (<>), Maximum([array]), Multiply (*), Parentheses

Formula 12 (determining monthly compensation based on percent of dollars saved, and comparing result to negotiated maximum)

If■Then

■Else, Average([array]), Subtract (
■), Greater than (>), Minimum([array]), Multiply (*), Parentheses

Formula 13 (converting one unit of measure to another)

ToText, Truncate, Division (/), Concatenate (+), Remainder

Formula 14 (customer service, determining and identifying warranty plan based on length of product serial number)

If■Then

■Else, Length, TrimLeft, Less than or equal (<+), Parentheses ()

Formula 15 (form letter, personalizing salutation based on degree and sex of recipient

Nested If■Then

■Else operators, Not equal (<>), Boolean operator (and), Equal (=), Concatenate (+), Parentheses

Formula 16 (shipping, calculating discounted value of shipment and adding calculated freight charge to orders that don't meet "free freight" criteria)

If■Then

■Else, Add (+), Less than (<), Multiply (*)

Formula 17 (form letter, splitting a mailing list in half and sending a different offer to each half of the list)

If■Then

■Else, Remainder, ToNumber, Equal, Parentheses

Formulas in action

A formula is a symbolic statement of the manipulations you want performed on certain data before it is printed on your report. Formulas are used:

- to calculate information you can't obtain directly from database data fields,
- to compare data,
- to join text with data,
- to convert data from one form to another,
- to enhance the formatting options with text strings, and
- to do a number of other things to customize your reports.

These formulas have been developed to demonstrate the use of multiple functions and operators in coordination with one another. The formulas have been created to illustrate concepts; they do not represent the only way, or necessarily the best way, to achieve the desired effects.

See Also

Formulas in Action Index

Formula 1

Functions/Operators Used:

Abs(x), (■) Subtract, (/) Divide, (*) Multiply, ToText, Concatenate

Formula Purpose

As a sales manager with a large sales force, you want to identify those sales reps who are the most consistent performers with regard to quota. You want to find those who come closest to hitting quota, regardless of whether they are slightly over or slightly under. It doesn't matter whether the variation is over or under quota; all that matters is the percent of variation from the mark.

Formula

```
ToText(Abs({file.quota}■{file.sales})/
{file.quota} * 100) + "%"
```

Result

Sales	Quota	Quota ■ Sales	% Variation
8, 000	10, 000	2000	20%
11, 000	10, 000	1000■	10%

Explanation

- The formula uses the Subtract operator to subtract {file.sales} from {file.quota}. This gives the dollar variation from {file.quota} (+ or ■).
- The Abs(x) function converts the dollar variation to an absolute number, ignoring any + or ■ signs.
- It then uses the Divide operator to divide that result by {file.quota}. This gives the variation expressed as a decimal fraction.
- The formula then uses the Multiply operator to multiply the result by 100 in order to calculate the final result in the form of a percentage.
- ToText is used to convert the calculated percent to text that can then be joined with other text.
- The Concatenate operator is used to join the percentage, once converted to text, to the percent sign character.

Return to formula index

Formula 2

Operators/Functions Used

If■Then

■Else, Subscript [], Less Than (<), Concatenate (+), Make range (to), ToNumber, Subtract (
■), ToText, Parentheses

Formula Purpose

You have the date of first purchase coded as the fourth and fifth characters of the customer number (for example, 1971 as the 71 in ABC7101234 , 1988 coded as the 88 in ABC880544, etc.) and you want to customize a letter to thank customers for the number of years they've done business with you. You want the following sentence to appear in your letter:

```
"You have been a valued customer for" [x] "years."
```

«where *x* is the number of years.»

Formula

```
If{file.Customer number}[4 to 5] < "90" Then
      "You have been a valued customer for " +
      ToText(91 ■ ToNumber({file.Customer number}
      [4 to 5])) + " years."
Else
      "You are one of our newer customers,
      and we want you to know how valuable you are to us."
```

Result

Customer #	Resulting Sentence
ABC7801234	"You have been a valued customer for 13 years."
ABD890337	"You have been a valued customer for 2 years."
ABD904331	"You are one of our newer customers, and we want you to know how valuable you are to us."

Explanation

- The If

■Then

■Else expression says, "If the 4th and 5th elements of the customer number, expressed as numbers, are less than 90, print a sentence including the date of first purchase, otherwise print the 'newer customer' sentence."

- The formula above uses the Subscript [ ] operator to extract the 4th and 5th characters (your date code) from the customer numbers which are stored as text in character fields. The Make Range operator (to) is used to establish the range 4 to 5.
- In the first example (ABC7801234) the 4th and 5th digits are *78* representing the year of first purchase as 1978.
- In the second example (ABD8903337), the 4th and 5th digits are *89* representing the year of first purchase as 1989.
- If the extracted characters are less than "90" (*Then*), Crystal Reports prints a concatenated text string (a sentence) that is customized to indicate the number of years the individual has been a customer. The text string says, "You have been a valued customer for (*calculated number, expressed as text*) years."

The calculation of the *number of years as a customer* involves several steps:

- As was done earlier, the Subscript [ ] operator extracts the 4th and 5th characters (your date code) from the customer numbers which are stored as text in character fields. The Make Range operator (to) is

used to establish the range 4 to 5.

- ToNumber converts the extracted date code to a number so it can be used in the calculation 91 x «where *x* = the date code expressed as a number».
- 91 x subtracts the year of first purchase from 91 (the current year) to get the number of years the individual has been a customer.
- ToText then converts the result of that calculation back to text so it can be used in the expression "You have been a valued customer for (x) years."
- If the characters are "90" or more (*Else*), Crystal Reports prints the fixed text string "You are one of our newer customers, and we want you to know how valuable you are to us."
- The Parentheses control the order of calculation of the formula.

Return to formula index

Formula 3

Operators/Functions Used

Concatenate, ToText, ToNumber, Multiply(*), Make Range, Subscript, Parentheses ()

Formula Purpose

A sail maker, as part of his loan agreement with his bank, has to submit a detailed inventory analysis monthly. The analysis must include the cost of fabric in raw material inventory and the cost of fabric by item number for each item in finished product inventory.

The company uses one fabric for all of the sails it produces, and it uses the 5th and 6th characters in the item number for each product to represent the number of meters of material (rounded to the nearest meter) necessary to make that item.

In the form letter the manager sends to his banker each month, he wants the computer to automatically insert the quantity on hand, the item number, and the dollar value of the fabric for each item number.

Formula

```
ToText({file.Quantity}) + " each, Item " + {file.Item} + ", $ "
+ ToText({file.Quantity} * ToNumber({file.Item}[5 to 6])
* {file.FabricCost})
```

Result

With a fabric cost of $14.88/meter, the formula delivers the following result:

Quantity	Item	Letter Text
46	4423141006	"46 each, Item 4423141006, $ 9582.72"
27	4423081009	"27 each, Item 4423081009, $ 3214.08"

Explanation

- This equation uses ToText to convert the value of *{file.Quantity}* (a numeric field) to text so it can be used as part of a concatenated text string.
- It then uses the Concatenate operator (+) to join the text version of *{file.Quantity}* with the string " each, Item "
- It again uses the Concatenate operator (+) to join the resulting string with the item number, stored as the value of the text field *{file.Item}*.
- It uses the Concatenate operator (+) one final time to join the resulting text string to the calculated fabric cost (expressed as text).

To calculate fabric cost:

- The formula uses the Subscript operator [] to extract the 5th and 6th elements (meters of material used) of *{file.Item}* (a text field).
- The Make Range operator (to) is used to establish the range 5 to 6.
- The ToNumber function converts those elements to a number so it can be used in a numeric calculation.
- It uses the Multiply operator (*) to multiply that number by the value of *{file.Fabric Cost}* by the number of units in inventory (*{file.Quantity}*) to arrive at the total price.
- Finally, it uses the ToText function to convert the total price (a number) into text that can be used in the concatenated string.

Return to formula index

Formula 4

Operators/Functions Used

Nested formulas, If■Then
■Else, (
■) Subtract, (*) Multiply, (>) Greater than, (>=) Greater than or equal, Sum, Parentheses

Formula Purpose

A computer store sells hardware, software, and books. It pays its sales reps 7% of all hardware sales (monthly) over $5000, 10% of all software sales (monthly) over $10, 000, and 5% of all book sales (monthly) over $1000. The sales manager wants to calculate the commission in each category for each rep, total the commission due each rep, and flag those reps who are entitled to more than $5000 in total commission for the month. We will do this using nested formulas, that is, using one formula as one of the elements in another formula.

Formula(s)

@HARDWARE

```
      If (({file.HardSales}■5000)> 0) Then
      .07 * ({file.HardSales}
■5000)
Else
      0
```

@SOFTWARE

```
      If (({file.SoftSales}■10000)> 0) Then
      .10 * ({file.SoftSales}
■10000)
Else
      0
```

@BOOKS

```
      If (({file.BookSales}■1000)> 0) Then
      .05 * ({file.BookSales}
■1000)
Else
      0
```

@TOTCOMM

```
      Sum([{@HARDWARE}, {@SOFTWARE}, {@BOOKS}])
```

@FLAG

```
      If {@TOTCOMM} >= 5000 Then
      " **** "
      Else
      ""
```

Result

Given this data:

Salesrep	HardSales	SoftSales	BookSales
Salesrep A	4500	21000	985
Salesrep B	31427	41222	4470

Salesrep C	22000	4687	4250
Salesrep D	14000	15678	2200

The formulas return the following results:

Salesrep	Hardware	Software	Books	Total	Flag
SalesrepA	0.00	1100.00	0.00	1100.00	
SalesrepB	3122.20	1849.89	173.50	5145.59	****
SalesrepC	1190.00	0.00	162.50	1352.50	
SalesrepD	630.00	567.80	60.00	1257.80	

Explanation

The three formulas (@HARDWARE, @SOFTWARE, and @BOOKS) work in the same manner. They use the If▪Then

▪Else operator to test for a condition, do one thing if the condition is true, do another thing if the condition is false.
▪They take total sales in the category, use the Subtract operator (
▪) to subtract the amount of sales on which no commission is to be paid, and test to see if the remaining amount is a positive number (>0).
▪If it is (if >0), they calculate the commissions using the Multiply operator (*) to multiply the appropriate commission percentage (expressed as a decimal: .07, .10, .05) times the commissionable amount (*{file.HardSales*
▪*5000}*, etc.).
▪If the remaining amount is a negative number (not >0), no commission is computed and the formula prints a zero amount (0.00).

- **@TOTCOMM** uses the *SUM* function to total the commissions due. Instead of including the calculations for each formula (a duplication of time and effort) it substitutes the formula name instead of the calculations. Crystal Reports knows that when it encounters the formula name it is to use the underlying calculations from the formula(s) referenced.
- **@FLAG** uses the If

▪Then
▪Else operator to evaluate the values calculated by @TOTCOM and to flag (****) those values of $5000 or more (>= 5000). @FLAG effectively nests two levels of other formulas: it nests @TOTCOM which is a formula that itself nests three other formulas, @HARDWARE, @SOFTWARE, and @BOOKS. When Crystal Reports sees @TOTCOM, it performs all of the underlying calculations referenced by that formula and the formulas that are used in @TOTCOM.

- Parentheses () are used throughout the formulas to control the order of calculation.

Return to formula index

Formula 5

Operators/Functions Used

Nested formulas, If▪Then

▪Else, Subtract (
▪), Not equal to (<>), Less than (<), Concatenate (+), ToText, To dollar ($), Negate (
▪()), Parentheses

Formula Purpose

A company wants to stimulate new credit sales for those customers with open credit limits and to make customers who are over their limits aware of their over▪limit balances. To do this the company wants to include a brief "P.S." to its already customized (company name, contact name, etc.) sales letter to each customer.

Formula

```
      If{file.Credit Limit}▪{file.Balance}<>0 Then
      If{file.Credit Limit}
▪{file.Balance}<0 Then
      "Your account is currently $" + ToText
      ($ (
▪{file.Credit Limit}

▪ {file.Balance}))
      + " over limit.
      Please contact us if you would like
      to discuss an increase in your credit limit."
Else
      "Your account has $" +
      ToText($({file.Credit Limit}

▪{file.Balance}))
      + " available credit.
      You can order today with no additional paperwork!"
Else
      ""
```

Result

CreditLimit	Balance	Result
$5000	$2250	"Your account has $2750.00 available credit. You can order today with no additional paperwork!
$3000	$3457	"Your account is currently $457.00 over limit. Please contact us if you would like to discuss an increase in your credit line.
$7500	$7500	«no message»

Explanation

The formula uses two If▪Then

▪Else operators, one nested inside the other.

▪ The first begins with the first word "If" and doesn't end until the "Else" at the very end of the expression. This expression says, essentially, "If the credit limit less the balance is less than zero, then

print the message based on the If
- Then
- Else expression that follows (the expression inside the parentheses). If the credit limit less the balance equals zero, print nothing. Think of this first If
- Then
- Else expression in this way: If *{file.Credit Limit}*
- *{file.Balance}* <> 0 Then (do what is in the parentheses) Else ""
- The second If
- Then
- Else expression begins "If *{file.Credit Limit}*" and ends after the word "paperwork". This expression says: "if the credit limit less the balance is less than zero, Then print the *over limit* message, otherwise (in those cases where the result is greater than zero) print the *available credit* message.
- The *Then* expression ToText($(
- *{file.Credit Limit}*
- *{file.Balance}))* means
- first use the Subtract operator (
-) to subtract *{file.Balance}* from *{file.Credit Limit}*.
- Since this part of the expression will only be activated if the result is a negative number (<0), the formula uses the Negate operator (
- ()) to multiply the result by
- 1 to return a positive number.
- The To Dollar operator ($) assures that the result will be printed in a dollar and cents format with two decimal places.
- The ToText function takes the resulting number and converts it to text characters so it can be used in the *over limit* message.
- The *Else* expression ToText($(*{file.Credit Limit}*
- *{file.Balance}*)) differs only slightly from the then expression above. This expression does not use the Negate operator (
-) because this expression *{file.Credit Limit}*
- *{file.Balance}* will be used only in those cases where the result is a positive number (>0).
- The Concatenate operator (+) joins the text strings (enclosed in quotation marks) with the number (converted to text using the ToText function) to produce the appropriate message (the *Then* message[over limit] or the *Else* message [available credit]).
- Many sets of parentheses are used to control the order of calculation of this formula.
- The characters "" at the very end of the formula are the representation of an empty string. This means, "print nothing."

NOTE: The leading space in the string "over limit." NOTE too, the leading space in the string "available credit." These spaces are purposely entered into the formula so a space occurs between the dollar figure and the words in each message.

Return to formula index

Formula 6

Operators/Functions Used

If■Then

■Else, Greater than (>), Percentage (%), Greater than or equal (>=), Boolean Operator And, Boolean Operator Or, Parentheses

Formula Purpose

The manager of a minor league baseball team wants a column on the statistics report that flags all batting averages of .300 or better and all averages below .200. The manager doesn't want the batting average flagged for any player who has batted fewer than 100 times because the manager doesn't feel that such a batting average is statistically significant.

Formula

```
If({file.AtBat}>=100) and
({file.Hits} % {file.AtBat}>=30) or
({file.Hits} % {file.AtBat}<20) Then

"####"
Else
""
```

Result

Hits	AtBat	Average	Flag	Explanation
31	98	.316		AtBat <100(not significant)
31	101	.307	####	Average > 30 (.300)
43	216	.199	####	Average < 20 (.200)
19	99	.192		AtBat < 100(not significant)

Explanation

The formula uses the If■Then

■Else operator to test for specific conditions.

- Condition A (*{file.TimesAtBat}* >= 100) uses the Greater than or equal operator (>=) to make certain that the batter has batted at least one hundred times. If he has batted 100 or more times, this condition is true, otherwise it is false.
- Condition B (*{file.Hits}* % *{file.TimesAtBat}* >30) uses the Percentage operator to calculate *{file.Hits}* as a percentage of *{file.TimesAtBat}*. It then uses the Greater than or equal operator (>=) to test the resulting percentage to see if it is 30% (.300 batting average) or more. If the percentage is greater than or equal to 30, this condition is true. Otherwise it is false.
- Condition C (*{file.Hits}* % *{file.TimesAtBat}* < 20) again uses the Percentage operator to calculate *{file.Hits}* as a percentage of *{file.TimesAtBat}*. It then uses the Less than operator (<)to test the resulting percentage to see if it is less than 20% (.200 batting average). If the percentage is less than 20, this condition is true. Otherwise it is false.

The formula uses the boolean operators *And* and *Or* to evaluate conditions A, B, and C. The logical relationship required is Condition A and either Condition B or Condition C. Thus, the *If* part of the formula is satisfied if Condition A is TRUE, and *either* Condition B or Condition C is TRUE.

- If the *If* part of the formula is satisfied (*Then*), it flags the batting average by printing four pound signs "####".
- If those conditions are not true (*Else*), it prints nothing (as indicated by the empty text string "".

Return to formula index

Formula 7

Operators/Functions Used

If■Then

■Else, Subtract (
■), Greater than (>), Maximum, Multiply (*), Parentheses

Formula Purpose

A company has the following bonus/commission structure for its sales force: when a sales rep sells at or over quota, the rep earns a $250 bonus or a 15% commission (on the amount over quota), whichever is highest. The company pays no bonus or commission on sales less than quota. The sales manager wants bonus/commission calculated and included on a report.

Formula

```
      If ({file.Sales}■{file.Quota})>=0 Then
      Maximum([250,.15*({file.Sales}

■{file.Quota})])
Else
      0
```

Result

Sales	Quota	15%	Amount Paid
10,000	8000	300	540 (calculated commission)
8000	8000	0	250 (guaranteed bonus)
7999	8000	0	0 (sales not >= Quota)

Explanation

This formula uses the If■Then

■Else operator to test sales to see if they are greater than or equal to quota.

- If they are (*Then*) it calculates the commission and pays the higher of commission or guaranteed bonus.
- If they are not (*Else*), it does nothing

To test sales to see if they are at least equal to quota, the formula uses the Subtract operator (■) to subtract *{file.Quota}* from *{file.Sales}*.

- If *{file.Sales}* is greater than or equal to *{file.Quota}*, this subtraction will produce the result of zero or a positive number (>=0). This will trigger the *Then* part of the formula.
- If *{file.Sales}* is less than *{file.Quota}*, this subtraction will produce a negative result. This will trigger the *Else* part of the formula.

The *Then* part of the formula uses the Multiply operator (*) to compute a 15% commission on the amount of sales in excess of quota: .15 * (*{file.Sales}* ■ *{file.Quota}*).

- It then uses the Maximum([*array*]) function to determine if the calculated commission or the guaranteed bonus of $250 offers the highest payout Maximum([commission, 250]), and it returns this highest value.
- Calculating this highest payout is the bottom line of the *Then* part of the formula

The *Else* part of the formula makes no calculations and returns 0.

Return to formula index

Formula 8

Operators/Functions Used

If■Then

■Else, Less than (<), Negate (
■), Add (+), Round, Average([array]), Parentheses

Formula Purpose

To better control her purchasing and inventory, a store owner wants the inventory report to include a QtyToOrder column. She wants that column to tell her what quantity she needs to order to keep on hand the average quantity sold monthly during the previous three months.

- If there is stock on hand, she'll want to order the average quantity sold less the stock on hand.
- If there is no stock on hand but no backorders, she'll want to order the average quantity sold.
- If there is no stock on hand and backorders, she'll want to order enough to clear the backorders plus the average quantity sold.

Formula

```
      If {file.OnHand} < 0 Then
      (( ■({file.OnHand}))
      + Round ((Average([{file.Month1}{file.Month2},
      {file.Month3}])))
Else
      (Round((Average([{file.Month1}{file.Month2},
      {file.Month3}]))■{file.OnHand}))
```

Result

OnHand	Month1	Month2	Month3	Avg.	Order
2	16	6	30	17	15
5	■7	3	19	10	15
0	9	8	18	12	12

Explanation

The formula uses the If■Then

■Else operator to set up one set of calculations if there are backorders (*{file.OnHand}* < 0), and a different set of calculations if there are no backorders.

- The *If* part of the formula uses the Less than operator (<) to see if the quantity on hand is less than zero. This indicates that, not only is all stock gone but there are unfilled orders (a backorder situation).

If there is a backorder, the *Then* part of the formula calculates the amount to order as the average monthly sales over a specified three month period plus the amount backordered (expressed as a positive number).

- It uses the Negate operator (
■) to convert the negative value of *{file.OnHand}* to a positive value.
- It uses the Average([*array*]) function to average the product sales for *{file.Month1}*, *{file.Month2}*, and *{file.Month3}* (the three months used to determine a typical sales pattern for the product).
- It uses the Round function to round the average product sales to the nearest whole number.
- It uses the Add operator (+) to add the quantity on hand (now expressed as a positive) to the rounded average monthly sales figure.
- The result is the amount to order.

If there is no backorder, the *Else* part of the formula calculates the amount to order as the average monthly sales over the specified three month period less the quantity already on hand.

- It uses the Average function to average the product sales for *{file.Month1}*, *{file.Month2}*, and *{file.Month3}* (the three months used to determine a typical sales pattern for the product).
- It uses the Round function to round the average product sales to the nearest whole number.
- It uses the Subtract operator to subtract the quantity on hand from the rounded, average monthly sales figure.
- The result is the amount to order.

Return to formula index

Formula 9

Operators/Functions Used

Nested If▪Then

▪Else expressions, NumericText, Subscript, Make Range, Equal to (=), ToNumber, In Range, Parentheses

Formula Purpose

A retailer has a customer list that includes both customers living inside the state and customers living in Canada. For customers inside the state, some have 5 digit ZIP codes, others have 9 digit ZIP codes. Canadian customers have Postal Codes beginning with a letter.

The company wants a column on the list that shows the sales tax that needs to be levied on orders from each customer. Local customers (those within the greater metropolitan area) get assigned a 7.5% sales tax (6.5% state and 1% local); customers in the state but outside the metropolitan area are assigned a 6.5% sales tax (state only); and Canadian customers are assigned no sales tax.

Formula

```
If NumericText({file.ZIP/Postal}[1 to 5])Then
If ToNumber({file.ZIP/Postal}[1 to 5]) in 92200 to 92399 Then
"7.5"
Else
"6.5"
Else
"0"
```

Result

ZIP/Postal	Tax	Explanation
91134	6.5	Inside state/outside metro area
92305	7.5	Inside state/inside metro area
92288▪5423	7.5	Inside state/inside metro area (ZIP + 4)
96544▪2333	6.5	Inside state, outside metro area (ZIP + 4)
T5A 9S2	0.0	Canadian customer

Explanation

This formula uses nested If▪Then

▪Else operators.

The first of these If▪Then

▪Else expressions begins with the initial *If* and ends with the final "0". It says:

- If the first five characters in the *{file.ZIP/Postal}* field are all numbers, then compute and print the sales tax using the methodology specified in the second If

▪Then

▪Else expression.

- If the first five characters are not all numbers, print "0".

The second of the If▪Then

▪Else expressions begins immediately *after* the initial *Then* and ends immediately *before* the final words *Else "0"*. It says, now that we've already tested and know that the first five characters of *{file.ZIP/Postal}* are numbers:

- If the first five characters of *{file.ZIP/Postal}*, converted to numbers, fall in the range of 92200 to 92399, print "7.5" (the amount of state and local sales taxes combined).

- If the first five characters of *{file.ZIP/Postal}*, converted to numbers, fall outside that range, print "6.5" (the amount of state sales tax only).

if NumericText(*{file.ZIP/Postal}* [1 to 5]) uses the If■Then

■Else operator to test for the condition in parentheses.

- *({file.ZIP/Postal}* [1 to 5]) uses the Subscript operator [] and Make Range operator *to* to extract characters 1 to 5 in the ZIP/Postal field.

The NumericText function evaluates the extracted characters to determine if they are all numbers or not.

If they are all numbers (YES) the first *If* condition is satisfied and the formula moves to the first *Then* consequence.

If they are not all numbers, the first *If* condition is not satisfied, and the formula moves to the final *Else* consequence.

- Then if ToNumber(*{file.ZIP/Postal}* [1 to 5]) in 92200 to 92399 Then "7.5" Else "6.5" else "0" shows what is to take place if the original *If* conditions have been satisfied.
- (*{file.ZIP/Postal}* [1 to 5]) uses the Subscript operator [ ] and Make Range operator *to* to extract characters 1 to 5 in the *{file.ZIP/Postal}* field.
- ToNumber converts these extracted characters to a number that can then be used in a numeric expression.
- *in 92200 to 92399* checks to see if the ZIP/Postal code (now converted to a number) falls anywhere within the range 92200 to 92399.

If it does, then the *If* condition (inside the parentheses) is satisfied and Crystal Reports performs the then consequence by printing the text string "7.5".

If it does not, then the *If* condition is not satisfied, and Crystal Reports perform the else consequence by printing the text string "6.5".

Comments

These tax figures, rules, and ZIP/Postal codes are for illustration only; they are not meant to accurately represent state and local taxing regulations.

Return to formula index

Formula 10

Operators/Functions Used

If■Then

■Else, Not, DayOfWeek, Make range (to), In range (in), Parentheses ()

Formula Purpose

A seven day a week customer service department logs all calls and stores the records in a database. As an aid in scheduling weekend staff, management wants to flag the weekend calls so they stand out in the call report.

Formula

```
If Not(DayOfWeek({file.CallDate}) in 2 to 6) Then
"Weekend"
Else
""
```

Result

Day of Week	#	Flag
Sunday	1	Weekend
Monday	2	
Tuesday	3	
Wednesday	4	
Thursday	5	
Friday	6	
Saturday	7	Weekend

Explanation

The formula uses the If■Then

■Else operator to say, "If the day of the week isn't a weekday, then print "Weekend", otherwise (else) print nothing (as indicated by the empty text string "").
The *If* part of the formula sets up the condition, "If the day of the week isn't a weekday.

- The DayOfWeek function evaluates *{file.Call Date}* and returns a number from 1 to 7 (Sunday being 1, Saturday being 7) to indicate the day of the week on which a call was made.
- *in 2 to 6* uses the In Range function (in) and the Make Range function (to) to determine if the day of the week the call was made was in the range 2 to 6 (Monday to Friday).
- Not negates the expression that follows.
- Without the Not function preceding the expression, the *If* expression reads "If the day of the week number indicates the call date *was* a weekday."
- With the Not function, the *If* expression reads, "If the day of the week number indicates the call date *was not* a weekday."
- If the call date is not a weekday, the *If* expression is satisfied, thus triggering the *Then* consequence.
- "Weekend" tells Crystal Reports that when the *If* condition is satisfied, print the word "Weekend"
- If the call date is a weekday, the *If* condition is not satisfied, thus triggering the *Else* consequence.
- "" tells Crystal Reports to print nothing if the *If* condition is not satisfied.

Return to formula index

Formula 11

Operators/Functions Used

If■Then

■Else, Subscript [], Not equal to (<>), Maximum([array]), Multiply (*), Parentheses

Formula Purpose

As part of a charity fund raiser, a company agreed to donate 1% of the invoice amount (or $1.00, whichever is larger) for each invoice it cut during the previous quarter. The only invoices that it has exempted are Credit Invoices (identified with the letter "C" as the 6th character in the invoice number) and invoices for non■profit agencies (identified with the letter "N" as the 6th character in the invoice number). Management wants a *Contribution* column to appear on the quarterly detail sales report.

Formula

```
If    {file.Inv#}[6]<> "C" and
{file.Inv#}[6]<>"N" Then
$(Maximum([.01 * {file.Amt}), 1])
Else
0
```

Result

Invoice #	Amount	Amt. * 1%	To Charity	Reason
21523R	143.27	1.43	$1.43	
21524C	223.46	N/A	0	«credit»
21538R	47.15	.47	$1.00	
21575N	1312.49	N/A	0	«nonprofit»

Explanation

The *If* part of the expression tests for two conditions: that the invoice is not a credit invoice and that it is not a non■profit invoice. Both conditions must be true to trigger the *Then* consequence.

- Condition A: *{file.Inv#}* [6] <> "C" uses the Subscript operator [] to extract the 6th character of the value stored in *{file.Inv#}*. It then compares that character to "C". If the character is not C (not a credit invoice), Condition A is satisfied.
- Condition B: *{file.Inv#}* [6] <> "N" uses the Subscript operator [] to extract the 6th character of the value stored in *{file.Inv#}*. It then compares that character to "N". If the character is not *N* (not a non■profit invoice), Condition B is satisfied.

The boolean operator *and* indicates that both Condition A and Condition B must be TRUE in order to satisfy the *If* part of the expression.

- If Condition A and Condition B are both TRUE, the *If* part of the expression is satisfied, thus triggering the *Then* consequences.
- .01 * *{file.Amt}* uses the Multiply operator (*) to multiply the amount of the invoice by 1% (.01).
- Max([.01 * *{file.Amt}*, 1]) returns the highest contribution for a given invoice: either 1% of the invoice amount or $1.00.
- The To Dollar operator converts the amount to a dollar value
- If either Condition A or Condition B is FALSE, or both are FALSE, the *If* part of the expression is not satisfied, thus triggering the *Else* consequences.
- If the *If* part of the expression is not satisfied, the formula returns the value 0.

Comments

An alternative rendering follows:

```
If {file.Inv#}[6] in ["C", "N"] Then
0
Else
Maximum([.01*{file.Amt}, 1])
```

Return to formula index

Formula 12

Operators/Functions Used

If■Then

■Else, Average([array]), Subtract (
■), Greater than (>), Minimum([array]), Multiply (*), Parentheses

Formula Purpose

A consultant has contracts that pay him a percentage of the amount he saves his clients with a cap or ceiling on the amount he can earn. His contracts pay him 25% of the monthly savings up to a maximum of $10, 000 per month. If his percentage amounts to more than $10, 000, he collects the $10, 000 maximum; if the percentage is less than $10, 000, he collects the actual percentage. The base period against which savings are calculated is the average of three typical months in a prior year.

Formula

```
      If (Average([{file.Mon1}, {file.Mon2}, {file.Mon3}])
      ■ {file.CurrentMon})> 0 Then
      Minimum([.25*((Average([{file.Mon1}, {file.Mon2},{file.Mon3})

■ {file.CurrentMon}]), 10000])
Else
      0
```

Result

Average	CurMon	Savings	25%	Amt.Due
90, 000	65, 000	25, 000	6250	6250
120, 000	60, 000	60, 000	15, 000	10, 000
75, 000	77, 000	N/A	N/A	0

Explanation

The formula uses the If■Then

■Else operator to say: if the average outlay during the three months specified was higher than the current month, then return 25% of the difference, up to a maximum of $10, 000. If the average outlay was equal to or less than the outlay during the current month, return 0.
The formula uses the Average function to calculate the average outlay during *{file.Mon1}*, *{file.Mon2}*, and *{file.Mon3}* ■ the baseline period.

It uses the Subtract operator (■) to subtract the outlay in the current month from the average during the baseline period, thus giving the difference between the current month and the baseline.

It uses the Greater than operator (>) to compare that difference to 0.

- If the difference is greater than 0 (there actually was a savings), the *If* part of the expression is satisfied, thus triggering the *Then* consequences.
- If the *Then* consequences are triggered:
- The formula uses the Average function to calculate the average outlay during *{file.Mon1}*, *{file.Mon2}*, and *{file.Mon3}*
■ the baseline period.
- It uses the Subtract operator (
■) to subtract the outlay in the current month from the average during the baseline period.
- It uses the Multiply operator (*) to calculate 25% of the difference (.25 *...).
- It uses the Minimum([array]) function to return the smaller amount: either 25% of the difference or $10, 000. Using the Minimum function in this way effectively puts a ceiling on the result of the calculation.
- If the difference is equal to or less than 0 (there was no savings), the *If* part of the expression is not

satisfied, thus triggering the *Else* consequences.

- The *Else* consequence (else 0) simply returns the value 0.

Return to formula index

Formula 13

Operators/Functions Used

ToText, Truncate, Division (/), Concatenate (+), Remainder

Formula Purpose

Bulk grains and nuts are inventoried by the ounce, but management wants to see a breakdown in pounds and ounces on the inventory report.

Formula

```
ToText((Truncate({file.Oz}/16)),0) + " pounds, " + ToText((Remainder
({file.Oz}, 16)),0) + " ounces "
```

Result

Ounces	Resulting Text
433	"27 pounds, 1 ounces"
278	"17 pounds, 6 ounces"
1455	"90 pounds, 15 ounces"

Explanation

The formula uses the Divide operator (/) to divide *{file.Oz}* by 16, thus calculating the number of pounds of the item in inventory. Unless *{file.Oz}* is perfectly divisible by 16, the quotient will be a whole number with decimal places.

- The Truncate function returns only the whole number (integer) part of the quotient.
- The ToText function converts the number to text so it can be used in a string with other text, and formats the number to zero decimal places (the second parameter)
- The formula uses the Concatenate operator (+) to connect the number of pounds with the text string " pounds" to give the new string " n pounds" «where *n* is the calculated number of pounds».

The Remainder function determines the remainder that results from the division *{file.Oz}*/16 (*{file.Oz}* the numerator, 16 the denominator). The remainder is a number of ounces less than 16 (less than a full pound).

- The ToText function converts the remainder to text so it can be included in the final text string, and formats the number to zero decimal places.
- The Concatenate operator (+) connects the number of ounces (now converted to text, to the string before ("n pounds, ") and to the string after (" ounces") to produce the final text string "n pounds, y ounces" «where *y* is the remainder».

Return to formula index

Formula 14

Operators/Functions Used

If■Then

■Else, Length, TrimLeft, Less than or equal (<+), Parentheses ()

Formula Purpose

Five and a half years earlier, a company had changed from a lifetime warranty for its products to a five year warranty. The first products that had been manufactured with the shorter warranty were now out of warranty, but the repair department was slow to adapt. It continued to repair all products for free, as if they were all covered by the original lifetime warranty.

All products manufactured under the new, shorter warranty had issued eight(8) character serial numbers instead of the 5, 6, or 7 character serial numbers that had been issued to lifetime■warranted products.

To begin controlling the situation, management has called for a column on its repair report that identifies products serviced as either lifetime warranty or five year warranty products.

Formula

```
If Length(TrimLeft({file.Serial#}))<= 7 Then
"Lifetime Warranty"
Else
"5 Year Warranty"
```

Result

Serial Number	# Characters	Flag
BP10001	7	"Lifetime Warranty"
BP1000	6	"Lifetime Warranty"
BP999	5	"Lifetime Warranty"
BP100001	8	"5 Year Warranty"

Explanation

The formula uses the If■Then

■Else operator to say, "If the serial number is 7 characters long or less (*Then*), print "Lifetime Warranty", otherwise (*Else*) print "5 Year Warranty".
TrimLeft(*{file.Serial#}*) removes all of the blank spaces stored to the left of the actual serial number in the right justified *{file.Serial#}* field. The formula uses TrimLeft to eliminate spaces because Length counts spaces as characters if they are present.

Length counts the characters in the actual serial number.

The formula uses the Less than or equal to operator (<=) to make certain that the serial number is 7 characters long (or less).

- If the serial number is less than or equal to 7 characters, the *If* part of the expression is satisfied, thus triggering the *Then* consequences.
- If the *If* part of the expression is satisfied, the formula prints the text string "Lifetime Warranty"
- If the serial number is greater than 7 characters, the *If* part of the expression is not satisfied, thus triggering the *Else* consequences.
- If the *If* part of the expression is not satisfied, the formula prints the text string "5 Year Warranty".

Return to formula index

Formula 15

Operators/Functions Used

Nested If■Then

■Else operators, Not equal (<>), Boolean operator (and), Equal (=), Concatenate (+), Parentheses

Formula Purpose

The director of a non■profit health care facility wants to automate the salutation in a letter he is sending to his staff. The staff consists of men and women, and professional and non

■professional employees. He wants to be certain that all doctors on staff (M.D., Ph.D., and D.D.S.) are given the form of address "Dr." and all non

■doctors are given the form of address appropriate to their sex.

Formula

```
If {file.Degree} <> "Ph.D." and
{file.Degree} <>"M.D." and
{file.Degree} <>"D.D.S" Then
If {file.Sex} = "M" Then
"Dear Mr. " + {file.lname}
Else
"Dear Ms. " + {file.lname}
Else
"Dear Dr. " + {file.lname}
```

Result

Degree	Sex	Last Name	Salutation
B.A.	F	Jones	"Dear Ms. Jones"
Ph.D.	F	Smith	"Dear Dr. Smith"
M.D.	M	Jackson	"Dear Dr. Jackson"
M.S.	M	Miller	"Dear Mr. Miller"
D.D.S.	F	Johnson	"Dear Dr. Johnson"

Explanation

This formula uses nested If■Then

■Else operators.

- The first If

■Then

■Else expression begins with the first if and continues to else "Dear Dr. " + {file.lname} at the end of the formula. It says, "If the degree listed is not a doctoral degree (then), follow the Then consequences which themselves contain another If

■Then

■Else expression. If the degree listed is a doctoral degree (else), print a doctoral salutation.

- The *If* part of the expression tests for three different conditions.
- Condition A uses the Not equal operator <> to make certain that the employee's degree is not Ph.D.
- Condition B uses the Not equal operator <> to make certain that the employee's degree is not M.D.
- Condition C uses the Not equal operator <> to make certain that the employee's degree is not D.D.S.
- The two And operators indicate that all three conditions must be met to satisfy the *If* part of the expression.
- If all three conditions are met, the *If* part of the expression is satisfied, thus triggering the *Then* consequences.

- If any one of the three conditions is not met (or 2 or all three), the *If* part of the expression is not satisfied, thus triggering the *Else* consequences.
- The second If
- Then
- Else statement begins with if *{file.Sex}* = and ends with else "Dear Ms. " + {file.lname}. It says, "If the employee is male (*Then*), print a male salutation. If the employee is anything but male (*Else*), print a female salutation. This If
- Then
- Else expression determines what actually happens if the *If* part of the first expression is satisfied.
- If the sex is male (determined by using the Equal operator =), the *If* part of the second If
- Then
- Else expression is satisfied, thus triggering the *Then* consequence (printing the salutation "Dear Mr. " + *{file.lname}*.
- If the sex is not male, the *If* part of the second If
- Then
- Else expression is not satisfied, thus triggering the *Else* consequence (printing the salutation "Dear Ms. " + *{file.lname}*.
- The Concatenate operator (+) connects the "Dear x" part of the salutation with the last name (as stored in the *{file.lname}* field).

Return to formula index

Formula 16

Operators/Functions Used

If■Then

■Else, Add (+), Less than (<), Multiply (*)

Formula Purpose

A manufacturer of lawn and garden products grants a 10■5

■5 discount on Category A products and a 15

■10

■5 discount on Category B products. It also pays the freight on all orders of $5, 000 or more (before discount). It charges a flat 4% freight on all orders of less than $5000. Management wants a Total Including Freight figure to appear on the daily sales report.

Formula

```
If ({file.CatA} + {file.CatB}) < 5000 Then
1.04 * .95 *.95 *.90 *{file.CatA} + .95 *.90 *.85
*{file.CatB}
Else
.95 *.95 *.90 *{file.CatA} + .95 *.90 *.85
*{file.CatB}
```

Result

CatA	CatB	CatA+CatB	TotInclFrt
9524	1344	10868	8712.62
3424	1344	4768	3908.21

Explanation

This formula uses the If■Then

■Else operator to say, if the sum of *{file.CatA}* and *{file.CatB}* is lower than $5, 000 (then), multiply the discounted price by 104% (1.04) to arrive at the price plus freight. If the sum of *{file.CatA}* and *{file.CatB}* is higher than $5000 (else), simply calculate the discounted price (no freight, since the order is bigger than $5000.

- The if part of the expression uses the Add operator (+) to calculate the undiscounted value of *{file.CatA}* and *{file.CatB}*.
- It uses the Less than operator (<) to determine if the sum of *{file.CatA}* and *{file.CatB}* is less than $5000.
- If the sum is less than $5000, the if part of the expression is satisfied, thus triggering the *Then* consequence.
- The formula uses the Multiply operator (*) several times, first to multiply the value of {file.CatA} by .90 (10% discount), to multiply that result by .95 (5% discount), and to multiply that result by .95 (5%) discount to arrive at the discounted amount for *{file.CatA}*.
- It performs the same set of calculations on *{file.CatB}* to determine the discounted amount for that category.
- It uses the Add operator (+) to add the discounted amounts of *{file.CatA}* and *{file.CatB}* to arrive at the discounted total (before freight).
- Finally, it uses the Multiply operator (*) to multiply the discounted total by 1.04 (100% + 4% freight) to arrive at the invoice total including freight.
- If the sum is $5000 or more, the *If* part of the expression is not satisfied, thus triggering the *Else* consequence.
- The formula uses the Multiply operator (*) several times, first to multiply the value of *{file.CatA}* by .90 (10% discount), to multiply that result by .95 (5% discount), and to multiply that result by .95 (5%)

discount to determine the discounted amount for *{file.CatA}* (freight free).

- It performs the same set of calculations on *{file.CatB}* to determine the discounted amount for that category.
- Finally, it uses the Add operator (+) to add the discounted amounts of *{file.CatA}* and *{file.CatB}* to arrive at the final invoice total (freight free).

Return to formula index

Formula 17

Operators/Functions Used

If■Then

■Else, Remainder, ToNumber, Equal, Parentheses

Formula Purpose

A retailer wants to test two different direct mail offers to see which one has the best "pulling" power. He wants to send one offer to all customers on his mailing list with even customer numbers and the second offer to all customers with odd customer numbers.

Formula

```
If Remainder(ToNumber({file.CustNum}), 2 )= 0 Then
{file.Offer1}
Else
{file.Offer2}
```

Results

Cust #	Remainder	Result
203104	0	Offer 1 ■ "You're entitled to a 10% discount."
203105	1	Offer 2 ■ "You're entitled to a free gift."

Explanation

- The ToNumber function converts the customer number (stored as text) to a number.
- The Remainder function takes that number, divides it by 2, and returns a remainder.
- The Equal operator tests the remainder to see if it is equal to 0.
- If the remainder is equal to zero, the customer number is divisible by two and thus an even number triggering the printing of *{file.Offer1}*.
- If the remainder is not equal to zero, the customer number is not evenly divisible by two and thus an odd number, triggering the printing of *{file.Offer2}*.

Return to formula index

INDEX TO FORMULA TOPICS

Formulas allow you to manipulate field data by performing calculations on it, controlling how it is displayed, controlling what data is displayed (through selection formulas), and even creating new data not directly available from a database. The following topics will help you design powerful, effective formulas that can produce more informative and more interesting reports.

Click on a topic below or search for the corresponding information:

Formulas ■ an overview

Using variables in formulas

Including comments with formulas

Deleting formulas from your report

If■Then

■Else formulas

Multi■condition If

■Then

■Else formulas

Copying formulas from Crystal Reports Help

Variables in formulas ▪ an overview

Crystal Reports allows you to use variables in formulas. While programmers need no introduction to variables, non▪programmers may find the following discussion helpful.

Variables are a special kind of value that you can use in a formula. Unlike a constant value which is fixed and unchanging, a variable holds a value that may change from time to time

A variable is like a container that can hold one value at a time. You assign a value to a variable, and the variable maintains that value until you later assign a new value. Then the variable maintains the new value until you later assign a newer value, etc.

When you use a variable in a formula, the formula looks to the variable and uses its current value in calculating the formula result. If the variable value changes, the formula looks to the new value and uses that to calculate a new result.

If you have a digital alarm clock, you're already familiar with a practical use of variables, even though you may not know what a variable is or that it is involved in the wake▪up process. You set a time to get up, and the alarm goes off at the time set. You set a new time to get up and the alarm goes off at the new time. The time the alarm goes off depends on the value assigned to a variable in the clock's programming code. Here's how it works.

You set a time to get up, and the alarm clock stores that time in a variable we'll call WakeUpTime. If you set the alarm to get up tomorrow at 7:00, for example, the clock stores 7:00 in WakeUpTime. The alarm goes off whenever the time of day (a variable we'll call TimeOfDay) matches the variable value. Thus, at 7:00, TimeOfDay matches the value stored in WakeUpTime and the alarm goes off.

A simple formula that shows this action is as follows:

```
If TimeOfDay = WakeUpTime Then
Alarm"
Else
""
```

This formula uses two variables because the time of day is changing constantly, and the wake up time can change to meet the user's needs.

Now if the alarm goes off and you decide to sleep a little longer, you may reset the alarm to seven thirty. When you do this, you are, in effect, assigning a new value to the variable WakeUpTime. The clock replaces the WakeUpTime value 7:00 with the new value 7:30. This time the alarm goes off when the time of day matches the new variable value.

The alarm clock uses a variable to store the wake up time because the manufacturer knows that the wake up time will vary from person to person, and for a given person it may vary from day to day. Were the manufacturer to have used a constant for the wake up time instead of a variable, the alarm would go off at the same time every day for every person using it, a most inflexible situation.

Uses for variables

Variables can be used to solve many formula problems, but they have two primary uses:

- streamlining formulas, and
- expanding formula capabilities.

Using variables to streamline formulas

Variables allow you to write certain formulas much more efficiently than you can without using variables. For example, without using variables it takes this kind of formula to evaluate the {customer.Telephone} field to determine if the area code is for Washington (206, 509) or British Columbia, Canada (604):

```
If {customer.Telephone}[1 to 3] = "604" Then
"BC"
Else
```

```
If{customer.Telephone}[1 to 3] = "206" or
{customer.Telephone}[1 to 3] = "509"
Then
"WA"
Else
""
```

You have to write out the instructions for extracting the area code from the telephone number field ({customer.Telephone} [1 to 3}) every time you want the formula to use the area code from the current record.

But using a variable (we'll call it AreaCode), you write those instructions one time. Crystal Reports uses those instructions, automatically extracts the area code from the *{customer.Telephone}* field, and stores it in the variable each time it reads a new record. You simply reference the variable (AreaCode) whenever you want to use the area code from the current record in your formula. Here's an example of the formula using a variable:

```
StringVar AreaCode := {customer.Telephone}[1 to 3];

If AreaCode = "604" Then
"BC"
Else
If AreaCode = "206" or AreaCode = "509" Then
"WA"
Else
""
```

Not only does the streamlined version take less time to write, but it takes less time to process by Crystal Reports as well, so your report prints more quickly.

A word about semicolons in formulas:

In a formula with multiple statements, the result of the final statement is the result that is returned (gets printed). When you have multiple statements in a formula, you must separate the statements using a semicolon so that Crystal Reports knows where one statement ends and the next begins. Without semicolons, Crystal Reports treats the entire formula as a single statement. In a multiple statement formula, this can result in an incorrect result or an error message.

The general rule that best describes the use of semicolons in multiple statement formulas is this:

- Every statement needs to end in a semicolon with two exceptions:
- the last statement in a formula doesn't need one, and
- the last statement before an "else" (when there are multiple statements before an else) doesn't need one.

Using variables to expand formula capabilities

Besides their impact on streamlining formulas, variables allow you to expand your formula writing capabilities. Before discussing the specifics of using variables in formulas, it is important to understand some things about the way the Formula Editor reads formulas.

How the Formula Editor reads formulas

Crystal Reports uses a sophisticated programming language in the Formula Editor. Because you can build formulas by picking fields, operators, and functions from lists of options, and because Crystal Reports supplies the required syntax elements when you build a formula by picking items from the list, that language is almost transparent for the most part (that is, there is not a lot of the language you have to remember). Because of their special capabilities, however, variables:

- come with a set of special requirements, and
- require you to have a better understanding of the programming language than you might need if you didn't use variables in your formulas.

Special requirements for using variables in formulas

Up till now, we've worked with formula elements that were pretty narrowly defined:

- a given operator only works in certain situations and with certain kinds of text and/or data,
- a function only works with a specific number of arguments, and each argument must be a specific data type, and
- If
- Then
- Else formulas work only if the data type of the *Else* part of the formula matches the data type of the *Then* part.

Such narrow definition allows you to create formulas, in many cases, simply by filling in the blanks, with the formula checker pointing out any mistakes you make.

Variables, however, are not narrowly defined. They are extremely flexible; you make them what you want them to be. You create them at will, and you include them in formulas as needed.

Because of this flexibility, it is necessary for you to define (declare) the variables before you use them so that Crystal Reports:

- is aware of them,
- understands how you intend to use them, and
- can set aside and set up the memory space they require.

You also need to assign values to the variables so Crystal Reports knows what value they are to return.

Crystal Reports knows only what you tell it about the variables. The fail-safe formula

- checker routines that work automatically with the other formula elements work with variables only after you define them.

To use a variable in a formula, you must do three things:

- declare the variable,
- set the value of the variable, and
- enter the variable in the formula.

Declaring the Variable

Crystal Reports requires you to declare all variables prior to using them. When you declare a variable, you tell the program:

- the name you intend to use for the variable, and
- the type of data you want the variable to hold.

The program uses this information to set aside a piece of memory for receiving and storing the values that are assigned to the variable.

NOTE: If you declare a variable with the same name and data type in two or more formulas, the formulas share the same variable. Thus, if one formula sets the value of the variable, the variable in the second (and additional formulas) reflects the change.

Naming the Variable

- the variable name must not exceed 254 characters, and
- it can't have the same name as a Crystal Reports operator or built-in function.

NOTE: As a general rule, you're probably better off if you keep the variable name short, easy to remember, and unique (not so close to the name of another variable as to cause confusion).

Data type

The data type of a variable determines the type of data that can be stored as a value in that variable. With Crystal Reports, you can create a variable with one of five data types:

- number
- currency
- Boolean
- date
- string

The data types correspond to the data types used throughout Crystal Reports.

If▪Then
▪Else formulas

If▪Then

▪Else formulas are conditional formulas: if a condition is met, then a certain consequence, an action, takes place. If the condition is not met, some other action takes place. For example:

- If a sales rep has already earned the maximum allowable bonus, print the amount of the maximum bonus allowed; if he has not yet earned the maximum, calculate the bonus actually earned and print it.
- If the value in the title field is "Mr.", print "Dear Mr." as the beginning of the salutation; if it is not "Mr.", print "Dear Ms." as the salutation.
- If the quantity on hand of a part is equal to or less than the reorder amount, reorder according to the reorder instructions; if the quantity is greater than the reorder amount, do nothing.

These are just a few of the kinds of conditional formulas you can create using the If▪Then

▪Else operator.

When using the If▪Then

▪Else operator, remember that there must be three separate parts to any If
▪Then
▪Else formula:

- the ***If*** part sets the condition.
- the ***Then*** part sets the action that takes place if the *If* condition is met.
- the ***Else*** part sets the action that takes place if the *If* condition is not met.

The data types (text, number, dollar amount, date, or Boolean) for the *Then* part and the *Else* part must be the same.

Thus, if the action that takes place if the condition is met (*Then*) is to print a text string, the action that takes place if the condition is not met (*Else*) must also be to print a text string, even if that text string is empty.

Multi-condition If-Then-Else formulas

Crystal Reports allows you to create powerful multi-condition formulas using the If-Then-Else operator.

Multi-condition and nested If-Then-Else formulas can be set up in this general pattern: *If* the red (first) condition is met, *Then*, go to the blue (second) condition. *If* the blue condition is met, *Then* perform the blue action; if the blue condition is not met (*Else*), perform the blue alternative. If the red condition is not met (*Else*), perform the red alternative. Thus:

```
If {file.Color1} = "Red" Then
If {file.Color2} = "Blue" Then
"Blue Action"
Else
"Blue Alternative"
Else
"Red Alternative"
```

This formula checks the field *{file.Color1}*.

- If the value of that field is "Red," it then checks the field *{file.Color2}*.
- If the value of that field is "Blue," it prints "Blue Action."
- If the value of *{file.Color2}* is not "Blue," it prints "Blue Alternative."
- If the value of *{file.Color1}* is not "Red," it prints "Red Alternative."

While multi-condition formulas look complex at first, after you've worked through one or two you'll find that they are not as intimidating as they seem, especially given the work they perform.

Including comments with formulas

Comments are notes that you include with a formula to explain its design and operation. They can be invaluable for understanding the interaction of the various formula components when memory of your initial inspiration fades, or when someone else is editing the formula in your absence.

Comments do not print and they do not affect the formula, but they appear with the formula whenever you call the formula up for editing. It is always a good idea to include comments with complex formulas, especially those formulas that will be used again and again over time.

To include comments with formulas

Type your comments in the Formula Text box in the Formula Editor, the same box you use for creating the formula. The comment can be above or below the formula, or it can even follow the formula on the same line if it is preceded by two slashes. Any of the following placements are acceptable:

```
//This is an acceptable
//position for a comment.
//Note that when we force
//the line break, we have to
//begin each new line with
//double slashes.

If ({file.Amount} in (100.00 to 250.00)) Then //This too is an
acceptable position
(.10 * {file.Amount})//for a comment.

Else
0

//This is also an acceptable position for a comment.
```

Comment considerations

The following are considerations when including comments with formulas:

- The proper syntax for a comment is two slashes // followed by the comment. When Crystal Reports sees the two slashes, it realizes that the text that follows is comment only and not to be included as part of the formula itself.
- Crystal Reports treats everything that follows the slashes on the same line as the slashes as a comment.
- If your comment is long and automatically wraps to the next line, no additional slashes are necessary; Crystal Reports treats it as one continuous comment.
- If you break your comment into two or more lines using the Enter key, you must begin each new line with two slashes. If you don't, Crystal Reports treats each unslashed line as part of the formula itself and displays an error message when you check the formula syntax.@@@

Formulas ▪ an overview

A formula is a symbolic statement of the manipulations you want performed on certain data before it is printed on your report. If your report is to contain a *{file.Sales}* field and a *{file.Cost}* field, for example, you may want to create an *@GrossProfit* field (@ designates that the field is a formula) and designate its value as *{file.Sales}* ▪ *{file.Cost}*. *{file.Sales}*

▪ *{file.Cost}* is a simple formula that tells Crystal Reports to subtract the value of the *{file.Cost}* field from the value of the *{file.Sales}* field and then to print the result. You can use formulas to calculate numeric values, compare one value to another and select alternative actions based on the comparison, join multiple text strings into a single string, and for a multitude of other purposes.

Formula examples

Here are some additional examples showing just a few of the things that you can do with formulas:

```
.80* {file.Price}
```

«multiplies the value in the *{file.Price}* field by .80 (80%). You can use a formula like this, for example, to calculate the discounted price (20% discount) for a valued customer.»

```
"Dear " + TrimRight({file.FNAME})+ ":"
```

«concatenates (joins) the text "Dear " with the value in the first name field *{file.FNAME}* and the additional text ":" (a trailing colon) to create a custom salutation for a form letter (Dear Bill:, Dear Mary:, etc.) The TrimRight function eliminates trailing blanks in the *{file.FNAME}* field so that the colon appears directly after the last letter in the name, regardless of the length of the name.»

```
{file.QTY} * {file.SELPRICE}
```

«calculates the extended price for a line item by multiplying the quantity ordered(*{file.QTY}*) by the selling price of the item(*{file.SELPRICE}*).»

```
If ({file.ONHAND}) < 5 Then
"Reorder"
Else
""
```

«using the word "Reorder," Crystal Reports flags those items that show a quantity on hand *{file.ONHAND}* less than 5, and puts no flag (as designated by the empty string "") with those items that show a quantity on hand of 5 or more.»

How to create a formula

Creating a formula in Crystal Reports is much like creating one in your favorite spreadsheet. You can use:

- fields (*{file.LNAME}*, *{file.LstYrSales}*, etc.),
- numbers (1,2, 3.1416),
- text ("Quantity" "," ":")
- operators (+,
- , etc.),
- functions (TrimRight(x), Length(x), etc.),
- group field values [Average(field, condField), Sum(field, condField, "condition"), etc.],or
- other formulas (@GrossProfit, @QUOTA, etc.)

You combine these components into working formulas using the Formula Editor. The Formula Editor allows you to type the components into the formula directly, or, in the case of fields, functions, operators, group field values, and other formulas, to select them from lists of those items that are available.

The Formula Editor requires you to enter the various components according to a specific set of rules called syntax: text enclosed in quotation marks, arguments enclosed in parentheses (where applicable), referenced formulas identified with a leading @ sign, etc. The Formula Editor checks the syntax and helps you debug (fix) problems before you enter the formula into the report.

NOTE: For an index to the various formula topics discussed in this section, see Formula Topics Index

Deleting Formulas from your report

When you create a formula and enter it in your report, Crystal Reports does two things:

- it stores the specification for creating that report, using the name you assigned it, and
- it places a working copy of that specification at the point you specify in the report.

In order to delete formulas, you can delete:

- individual working copies of the formula.
- the specification and all working copies of the formula, or

NOTE: You cannot delete the specification without deleting all working copies of the formula.

To delete individual working copies of the formula

- To delete individual working copies of the formula, select the formula copy you want to delete and then:
- Click the right mouse button and select Delete Field from the pop
- up menu that appears,
- Press the Delete key on your computer's numeric keypad, or
- Select Edit|Clear.

NOTE: Even after you have deleted all of the working copies of a formula from your report, the formula specification remains unchanged. The specification is listed in the Insert Formula dialog box, and it is available for immediate use should you wish to enter the formula again in your report. To delete the formula specification, follow the steps below.

To delete the specification (after you have deleted all working copies of the formula)

1. Once you have deleted all working copies of the formula, select Insert|Formula. The Insert Formula dialog box appears.
2. Select the formula specification you want to delete from the Formula name list. Crystal Reports activates the Delete button.
3. Click the Delete button, and Crystal Reports deletes the formula specification.

NOTE: If you haven't deleted all working copies of the selected formula, the program displays the following message: Please delete all uses of the formula in the report first.

Copying formulas from Crystal Reports Help

Windows allows you to copy text from help topics to the Clipboard. You can then paste this text wherever it is needed. Since the formulas you develop using the Formula Editor are simply text, you can save yourself a lot of time by copying useful formulas directly into the Formula Editor and then modifying it to fit your needs..

See Also

Formulas in Action

KStacking Formulas And Fields

When you use an If■Then

■Else formula, the data type of the *Then* consequence must match the data type of the *Else* consequence. In other words, you cannot have a formula like this:

```
If {file.AMOUNT}<>0 Then
{file.AMOUNT}
Else
"Zero"
```

«where *file.Amount* is a numeric field.»

- The data type of the then consequence file.Amount is a number.
- The data type of the else consequence field must be a number as well.
- But in this formula, the else consequence is a string, "Zero".

Since the data types don't match, the program will not allow the formula and you will get an error message.

The formula above is trying to solve a real world reporting problem, i.e., to print a string instead of the number 0 to call particular attention to zero values. Fortunately, there is a way to get the results wanted, and that is by stacking fields and formulas (multiple fields and/or formula fields stacked on top of one another).

NOTE: A stack of formulas is not a vertical stack of formula fields (vertical on the screen). When you stack formulas, the stack seems to build out from the screen, towards you (theoretically, not visually). A stack of formulas looks essentially like one formula, but the text inside the top formula box is garbled because the text from formula boxes lower in the stack is showing through as well.

Reproducing the effect using a stack

Creating the effect from the formula above using stacked fields and/or formula fields follows this process:

1. Insert the *file.Amount* field in the Details section of your report.
2. Format the file.Amount field so that nothing prints if the value in the field is zero. To do this, you format the field using the Suppress if Zero option in the Format Number dialog box. At this point, the value in the *file.Amount* field will print *only* if it is something *other* than zero.
3. Create an If■Then

■Else formula that prints the word "Zero" if the value in the *file.Amount* field is zero and that prints nothing (an empty string) if the value in that field is something other than zero.

```
If file.AMOUNT = 0 Then
     "Zero"
   Else
     ""
```

4. Stack the formula field directly on top of the file.Amount field.

- Both stacked items (the *file.Amount* field and the formula field) are mutually exclusive.
- Each one prints something only when the other one doesn't print.
- By restricting our stack to mutually exclusive fields and formulas, we never run into a situation where two or more values print at the same time, thus putting unreadable and unsightly printing in your report.
- As long as we restrict our stack to mutually exclusive items, we can put many items in the stack and still get only one value printing at a time.

With this understanding of stacking theory and some careful planning and experimenting, you can create some very intelligent reports.

NOTE: When fields are stacked in such a way that you can select more than one field, the field selected becomes the top field in the stack.

Functions Index (By Function Type)

Functions are built-in procedures or subroutines used to evaluate, make calculations on, or transform data. When you specify a function, Crystal Reports performs the set of operations built into the function without you having to specify each operation separately. In this way, a function is a kind of shorthand that makes it easier and less time consuming for you to create reports.

Click on the function of interest from the choices below. For convenience some functions are listed in more than one place:

Functions (Alphabetical, by function name)

Arithmetic Functions

"Grand Total" Functions

Functions To Duplicate Group Fields

String functions

Date Functions

Date Range Functions

Array Functions

Special Fields Functions

Evaluation Time Functions

Example report using evaluation time function

Summary Functions

Other Functions

Additional Functions

ARITHMETIC FUNCTIONS INDEX

Arithmetic functions are used for a variety of arithmetic-oriented calculations and operations.

Click on the function of your choice for more information:

Abs(x)

Average(field)

Average(field, condField)

Average(field, condField, "condition")

Average([array])

Count(field)

Count(field, condField)

Count(field, condField, "condition")

Count([array])

DistinctCount(field)

DistinctCount(field, condField)

DistinctCount(field, condField, "condition")

DistinctCount([array])

Maximum(field)

Maximum(field, condField)

Maximum(field, condField, "condition")

Maximum([array])

Minimum(field)

Minimum(field, condField)

Minimum(field, condField, "condition")

Minimum([array])

PopulationStdDev([array])

PopulationStdDev(field, condField)

PopulationStdDev(field, condField, "condition")

PopulationStdDev(field)

PopulationVariance([array])

PopulationVariance(field, condField)

PopulationVariance(field, condField, "condition")

PopulationVariance(field)

Remainder(numerator, denominator)

Round(x)

Round(x, # places)

StdDev([array])

StdDev(field, condField)

StdDev(field, condField, "condition")

StdDev(field)

Sum(field)

Sum(field, condField)

Sum(field, condField, "condition")

Sum([array])

Truncate

Variance([array])

Variance(field, condField)

Variance(field, condField, "condition")

Variance(field)

"GRAND TOTAL" FUNCTIONS INDEX

Use these functions whenever you want Crystal Reports to evaluate all the values in a given field for the entire report and return a calculated value (in effect, a "grand total" average, a "grand total" count, etc.).

Click on the function of your choice for more information:

Average(field)

Count(field)

DistinctCount(field)

Maximum(field)

Minimum(field)

PopulationStdDev(field)

PopulationVariance(field)

StdDev(field)

Sum(field)

Variance(field)

FUNCTIONS TO DUPLICATE GROUP FIELDS

Use these functions whenever you want to duplicate a group field in a formula.

NOTE: In order to use these functions, you must have already entered a group field in your report with identical parameters to those you plan to reproduce in the formula, i.e., same field, same sort and group by field, same date/Boolean condition (if applicable), and same action (average, count, etc.).

Click on the function of your choice for more information:

Sort and group by field = character, number, or dollar value.

Average(field, condField)

Count(field, condField)

DistinctCount(field, condField)

Maximum(field, condField)

Minimum(field, condField)

PopulationStdDev(field, condField)

PopulationVariance(field, condField)

StdDev(field, condField)

Sum(field, condField)

Variance(field, condField)

Sort and group by field = date or Boolean.

Average(field, condField, "condition")

Count(field, condField, "condition")

DistinctCount(field, condField, "condition")

Maximum(field, condField, "condition")

Minimum(field, condField, "condition")

PopulationStdDev(field, condField, "condition")

PopulationVariance(field, condField, "condition")

StdDev(field, condField, "condition")

Sum(field, condField, "condition")

Variance(field, condField, "condition")

STRING FUNCTIONS INDEX

String functions are used for the evaluation, manipulation and conversion of text strings.

Click on the function of your choice for more information:

Length(x)

LowerCase (x)

NumericText(fieldname)

ReplicateString(x, n)

ToNumber (x)

ToText (x)

ToText (x, # places)

ToWords(x)

ToWords(x, # places)

TrimLeft (x)

TrimRight (x)

UpperCase (x)

DATE FUNCTIONS INDEX

Date functions allow you to convert numbers to dates (which you can then format to display as you wish) and to convert dates to numbers.

Click on the function of your choice for more information:

Date (yyyy, mm, dd)

Day (x)

DayOfWeek (x)

Month (x)

Today

Year (x)

DATE RANGE FUNCTIONS

Date range functions are preset date ranges.

Aged...Days

Aged as of today the number of days specified

Aged0To30Days

Aged as of today 30 days.

Aged31To60Days

Aged as of today 31 to 60 days.

Aged61To90Days

Aged as of today 61 to 90 days.

Over90Days

Aged as of today over 90 days.

AllDatesFromToday

From today to any future date

AllDatesFromTomorrow

From tomorrow to any future date

AllDatesToToday

From the first date entered to today.

AllDatesToYesterday

From the first date entered to yesterday.

Calendar1stHalf/2ndHalf

All included dates in the first or second half of the calendar year

Calendar1stHalf

All included dates in the first half of the calendar year.

Calendar2ndHalf

All included dates in the second half of the calendar year.

Calendar1stQtr...4thQtr

All included dates in the 1st to 4th quarters of the fiscal year

Calendar1stQtr

All included dates in the 1st quarter of the fiscal year.

Calendar2ndQtr

All included dates in the 2nd quarter of the fiscal year.

Calendar3rdQtr

All included dates in the 3rd quarter of the fiscal year.

Calendar4thQtr

All included dates in the 4th quarter of the fiscal year.

Last4WeeksToSun

The four weeks previous to last Sunday. Begins on a Monday and ends on a Sunday. If today is Sunday, begins four weeks ago on a Monday and ends today, including today. If today is not a Sunday, begins four weeks before the previous Sunday and ends on that Sunday. For example, January, 1995, begins on a Sunday.

{bmc bullet.bmp} If today is January 22 (a Sunday), Last4WeeksToSun begins on January 2 (a Monday) and ends on January 22 (today, a Sunday).

{bmc bullet.bmp} If today is January 27 (a Saturday), Last4WeeksToSun begins on January 2 (a Monday) and ends on January 22 (the previous Sunday).

Last7Days

From seven days ago to today

LastFullMonth

From the first to last day of the previous month

LastFullWeek

From Sunday to Saturday of the last full week

LastYearMTD

All dates in the current month last year, up to the current date last year

LastYearYTD

All dates in the last year, up to the current date last year.

MonthToDate

From the first day of the month to today.

Next...Days

Dates in the period specified starting from today.

WeekToDateFromSun

From last Sunday to Today.

YearToDate

From the first day of the calendar year to today.

ARRAY FUNCTIONS INDEX

These functions calculate a value based on an array of values.

Click on a function below for more information:

Average([array])

Count([array])

DistinctCount([array])

Maximum([array])

Minimum([array])

PopulationStdDev([array])

PopulationVariance([array])

StdDev([array])

Sum([array])

Variance([array])

SPECIAL FIELD FUNCTIONS INDEX

Special Field Functions can be used to insert special fields in formulas.

Click on a function below for more information:

GroupNumber

PageNumber

PrintDate

RecordNumber

EVALUATION TIME FUNCTIONS INDEX

Evaluation Time Functions can control how data is presented by controlling when it is calculated or presented.

Click on a function below for more information:

BeforeReadingRecords

WhilePrintingRecords

WhileReadingRecords

OTHER FUNCTIONS INDEX

Click on a function below for more information:

GroupNumber

IsNull

Next

NextIsNull

PageNumber

Previous

PreviousIsNull

PrintDate

RecordNumber

Summary Functions Index

Summary functions can be designed to perform operations on group data.

Click on a function below for more information:

Average([array])

Average(field, condField)

Average(field, condField, "condition")

Average(field)

Count([array])

Count(field, condField)

Count(field, condField, "condition")

Count(field)

DistinctCount([array])

DistinctCount(field, condField)

DistinctCount(field, condField, "condition")

DistinctCount(field)

Maximum([array])

Maximum(field, condField)

Maximum(field, condField, "condition")

Maximum(field)

Minimum([array])

Minimum(field, condField)

Minimum(field, condField, "condition")

Minimum(field)

PopulationStdDev([array])

PopulationStdDev(field, condField)

PopulationStdDev(field, condField, "condition")

PopulationStdDev(field)

PopulationVariance([array])

PopulationVariance(field, condField)

PopulationVariance(field, condField, "condition")

PopulationVariance(field)

StdDev([array])

StdDev(field, condField)

StdDev(field, condField, "condition")

StdDev(field)

Sum([array])

Sum(field, condField)

Sum(field, condField, "condition")

Sum(field)

Variance([array])

Variance(field, condField)

Variance(field, condField, "condition")

Variance(field)

Abs

Format

Abs(x)

«where x is the number for which you want the absolute value returned.»

Action

Abs returns the value of x, ignoring any + or ■ signs.

Example(s)

```
      Abs(1.50) = 1.50

      Abs(10 ■ 7) = 3
If Abs(37 ■ {file.Temp}) > 1 Then
      "Maintenance, Temperature Check"
Else
      ""
```

«Flags instances where a laboratory heat block has a temperature variation greater than +/■ 1 degree C.»

Expanded example(s) using this function

Formula 1

Average(field)

Format

Average(field)

«where *field* is a number or dollar value field or formula.»

Action

Average calculates the average (mean) value for the *field* for the entire report. It is, in effect, a "grand total" average.

Typical use(s)

Use any time you need to calculate the average value for a column of number or dollar values, or use the average value in a calculation or comparison.

Example(s)

```
Average({file.Amount}) =
```

«Calculates the average of all values in the *{file.Amount}* field.»

```
If {file.Amount} > Average({file.Amount}) Then
"Above Average"
Else
""
```

«Compares each value in the *{file.Amount}* field to the average of all values in the *{file.Amount}* field. If the value is above average, it flags it "Above Average;" if it is average or below average, it prints nothing.»

Average(field, condField)

Format

Average(field, condField)

- *field* is the name of the number or dollar value field for which Crystal Reports generates a summary field, and
- *condField* is the name of the character, number, or dollar value field that triggers the summary field to print whenever its value changes.

Action

Averages the values in each group that results from the specified subtotal or summary field. This is a group average, the average of a group of values from a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that averages the values in a group, and
- that uses a character, number, or dollar value field as the sort and group by field.

Example(s)

```
Average({file.Orders},{file.Customer}) =
```

«Calculates the average order in each group of customer orders. The orders are separated into groups whenever the value in the *{file.Customer}* field changes.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (average).

Average(field, condField, "condition")

Format

Average(field, condField, "condition")

- *field* is the name of the number or dollar value field for which Crystal Reports generates the summary field,
- *condField* is the name of the date or Boolean field that triggers the summary field to be calculated whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met.

Action

Averages the values in each group that results from the specified summary field. This is a group average, the average of a group of values in a given field. This function works just like Average(*field*, *condField*), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a subtotal or summary field:

- that averages the values in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
Average({file.Order},{file.Date}, "monthly") =
```

«Calculates the average order in each group of orders in the *{file.Orders}* field (the average order for each month). The orders are separated into groups whenever the value in the *{file.Date}* field changes to a new month.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (average).

Average([array])

Format

Average([*x*,...])

> «where *x* is an array of numeric values (constants, field values, formula results) separated by commas.»

Action

Average([*array*]) returns the average (mean) value for an array of constants, data field values, or formulas (a*b, c/d, etc.) separated by commas.

Typical use(s)

Use any time you're more interested in the typical or normal value in a group of values than in the actual values themselves. Typical uses include such things as calculating average salaries, average sales, average rainfall, average inventory turns, etc.

Example(s)

```
Average([25,50,75,100]) = 250/4 = 62.50
```

Expanded example(s) using this function

Formula 8

Formula 12

BeforeReadingRecords

Format

BeforeReadingRecords

Action

Specifies that the formula is to be evaluated before the database records are read.

Typical use(s)

Formulas are normally evaluated at the following times:

- If no database or group field is included in the formula, the formula is evaluated before the program reads database records.
- If a database is included in the formula, the formula is evaluated while the program reads database records.
- If a group field, page # field, subtotal, etc. is included in the formula, the formula is evaluated after database records are read and while the data from the records is being printed in the report.

BeforeReadingRecords forces the formula to be evaluated before the program reads database records. When this function is used in a formula, the Formula Checker returns an error message if you attempt to include elements in the formula (database fields, groups, etc.) that must be evaluated at a later time (while reading or while printing records).

Example(s)

```
BeforeReadingRecords;
ToNumber ("12345")
```

«Forces the formula (which contains no database fields or groups), to be evaluated at the time it is normally evaluate (before reading records).»

NOTE: If you try to include a database field or group in this formula, you get an error message.

Count(field)

Format

Count(field)

«where *field* is the field in which you want the number of values counted»

Action

Counts the number of values in the *field* for the entire report. It creates, in effect, a "grand total" count.

Typical use(s)

Use Count(*field*) any time you need to print the count (number of values in a field) in a report or use the number in a calculation or comparison.

Example(s)

```
If Count({file.Orders}) >= 100 Then
"Congratulations on meeting your quota!"
Else
""
```

«prints the congratulatory message if the number of orders is 100 or more, and prints nothing if the number of orders is less than 100.»

Count(field, condField)

Format

Count(field, condField)

- *field* is the name of the field for which Crystal Reports generates the summary field value, and
- *condField* is the name of the character, number, or dollar value field that triggers the summary field to print whenever its value changes.

Action

Counts the values in each group that results from the specified summary field. This is a group count, the count of a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that counts the values in a group, and
- that uses a character, number, or dollar value field as the sort and group by field.

Example(s)

```
Count({file.Orders},{file.Customer}) =
```

«This formula counts the number of orders in each group of orders in the *{file.Orders}* field (the total orders for each month). The orders are separated into groups whenever the value in the *{file.Customer}* field changes.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (count).

Count(field, condField, "condition")

Format

Count(field, condField, "condition″)

- *field* is the name of the field for which Crystal Reports generates the summary field value,
- *condField* is the name of the date or Boolean field that triggers the summary field to print whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met .

Action

Counts the values in each group that results from the specified summary field. This is a group count, the count of a group of values in a given field. This function works just like Count(field, condField), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that counts the values in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
Count({file.Order},{file.Date}, "monthly")
```

«Counts the number of orders in each group of orders in the *{file.Orders}* field (the total orders for each month). The orders are separated into groups whenever the value in the *{file.Date}* field changes to a new month.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (count).

Count([array])

Format

Count([x,...])

«where *x* is an array of data elements separated by commas»

Action

Count counts the number of data elements in an array.

Typical use(s)

Use this whenever you need to condition a response on the count of items in an array.

Example(s)

Count([1, 2, 3, 4, 5]) = 5

Date

Format

Date (YYYY, MM, DD)

Action

Date returns a date from the year, month, and day numbers entered.

- The year must be four digits.
- The month must be a number from 1 to 12.
- The day must be a number from 1 to 31.

Typical use(s)

This function is used so that the formula can differentiate between a date and a division equation such as 1999/10/1.

Example(s)

```
Date(1990,10,1) = Oct 01 90
```

```
If PageNumber = 1 Then
PrintDate
Else
Date(0,0,0)
```

«prints the print date on the first page, and prints nothing [as designated by the empty date [Date(0,0,0)] on the remaining pages. Date(0,0,0) is the designation for an empty or non-printing date. It satisfies the need for a date data type in certain If

-Then
-Else formulas without printing anything.»

Comments

- Dates specified must be within the range 0000 and 9999.
- To create an empty (null) date (for example, to use in an If

-Then
-Else formula) use Date(0,0,0). Empty dates don't print. The empty date is the date equivalent of the empty string.

- If you want to convert a date to text, you must use a formula like this:

```
ToText(Year(Date)) + "/" + ToText(Month(Date)) + "/" + ToText(Day(Date))
```

Day

Format

Day(x)

«where *x* is a date or a field that has a date as a value.»

Action

Day extracts the day from a date and returns a number.

Typical use(s)

Use any time you need only the day component of a date. For example, if you're tracking payments that fall within a given month, you're interested only in the day they arrive; month and year information would be redundant. Also, if you need to use the day of the month in a numeric calculation (Day(*{file.OctPmt}*) ■ Day (*{file.SepPmt}*), for example) use the Day function to extract the day of the month and convert it to a number.

Example(s)

```
If Day({file.LastPaymentOn}) < 15 Then
"Past Due"
Else
""
```

«if the day of the last payment was less than 15, print "Past Due", otherwise print nothing»

DayOfWeek

Format

DayOfWeek(x)

«where *x* is a date»

Action

DayOfWeek:

- extracts the day component of a date,
- determines the day of the week the date falls on, and
- converts the day of the week to a number (1 to 7) where Sunday is the first day of the week.

Typical use(s)

Use this function any time you need to use the day of the week as a number in a numeric expression.

Example(s)

```
(DayOfWeek(Date(1990,10,1) = 2
```

«where October 1, 1990 is a Monday».

```
If DayOfWeek({file.orderdate}) = 7 Then
"Saturday"
Else
""
```

Comments

If you want to get the day of the week spelled out, use this formula:

```
["Sun", "Mon", "Tues",...] [DayOfWeek(Date)]
```

«sets up an array (["Sun",...]) and uses the number of the day of the week (Sun = 1, Sat = 7) to select the desired date name from the array.»

Expanded example(s) using this function

Formula 10

DistinctCount(field)

Format

DistinctCount(field)

«where *field* is the field in which you want the number of distinct values counted»

Action

Counts the number of distinct values in the *field* for the entire report. Duplicate values are not counted.

Typical use(s)

Use DistinctCount(*field*) any time you need to print the number of distinct (unique) values in a field in a report or use the number in a calculation or comparison.

Example(s)

```
DistinctCount({file.Amount})
```

«counts the number of distinct values in the *{file.Amount}* field.»

```
if DistinctCount({file.Customer}) >= 100 Then
"Congratulations on meeting your quota!"
Else
""
```

«prints the congratulatory message if the number of new customers a sales representative brings in is 100 or more, and prints nothing if the number of orders is less than 100.»

DistinctCount(field, condField)

Format

DistinctCount(field, condField)

- *field* is the name of the field for which Crystal Reports generates the summary field value, and
- *condField* is the name of the character, number, or dollar value field that triggers the summary field to print whenever its value changes.

Action

Counts the number of distinct (unique) values in each group that results from the specified summary field. This is a group count, the count of a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that counts the number of distinct values in a group, and
- that uses a character, number, or dollar value field as the sort and group by field.

Example(s)

```
DistinctCount({file.Customer},{file.Rep}) =
```

«This formula counts the number of different customers that each Sales Representative has sold to. The customers are separated into groups whenever the value in the *{file.Rep}* field changes, and the number of distinct values in the *{file.Customer}* field is counted for each group.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (DistinctCount).

DistinctCount(field, condField, "condition")

Format

DistinctCount(field, condField, "condition")

- *field* is the name of the field for which Crystal Reports generates the summary field value,
- *condField* is the name of the date or Boolean field that triggers the summary field to print whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met.

Action

Counts the number of distinct (unique) values in each group that results from the specified summary field. This is a group count, the count of a group of distinct values in a given field. This function works just like DistinctCount(field, condField), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that counts the number of distinct values in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
DistinctCount({file.Customer},{file.Date}, "monthly") =
```

«Counts the number of different customers in each group of customers in the *{file.Customers}* field (customers making more than one order are ignored). The orders are separated into groups whenever the value in the *{file.Date}* field changes to a new month.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (DistinctCount).

DistinctCount([array])

Format

DistinctCount([x,...])

«where *x* is an array of data elements separated by commas»

Action

DistinctCount counts the number of distinct (unique) data elements in an array.

Typical use(s)

Use this whenever you need to condition a response on the distinct count of items in an array.

Example(s)

```
DistinctCount([1, 2, 3, 2, 5, 1, 4, 3, 2]) = 5
```

GroupNumber

Format

GroupNumber

Action

Inserts the current group number in a field in a formula.

Typical use(s)

You can use this function in group selection formulas to print some groups and exclude others.

Example(s)

```
GroupNumber
```

«creates a field containing the current group number that you can place on your report. Using this formula is the equivalent of using the Insert|Special Field|Group Number Field command.»

IsNull

Format

IsNull(fld)

«where *fld* is any field in the report.»

Action

Evaluates the field specified in the current record and returns TRUE if the field contains a null value.

Typical use(s)

You can use this function in a record selection formula to limit the report to records that have something other than a null value in the field specified. You can also use it to have Crystal Reports take some action whenever it encounters a null value.

Example(s)

```
Not IsNull({file.AMT})
```

«when used as a record selection formula, includes in the report only those records that have something other than a null value in the *{file.AMT}* field.»

Length

Format

Length(x)

«where *x* is a text string, or a field containing a text string»

Action

Length returns the number of characters in a text string that you enter into the formula, or in a text string stored as a value in a data field.

NOTE: Text strings must be enclosed in either double or single quotation marks (" " or ` '). Crystal Reports includes any blank spaces as part of the character count.

Typical use(s)

Use this function any time you have a manipulation, comparison, or calculation that is dependent on the length of a text string.

Example(s)

```
Length("Account") = 7
Length(ToText({file.AMOUNT})) = 8
«where {file.AMOUNT} = 14,233.08»
```

NOTE: ToText does not convert commas (or other thousands separators) to text. In the present example, it converts 14,233.08 (the number) to "14233.08" (text) and then counts the number of characters in the text to arrive at the length (8)

```
Length("SMITH") = 5
Length("BOB SMITH") = 9
```

«the blank space between BOB and SMITH is counted as a character»

```
Length({file.Field1}) =
```

«the length of the text string stored as a value in *{file.Field1}*.»

```
{file.ItemNum}[Length({file.ItemNum})■2]
```

«Returns the third character from the right in an item number. This can be used, for example, if item numbers are of different length but the character that represents the color is always the third from the last character. The formula determines the length of the item number, subtracts 2 from it, and uses the result to identify the character in the item number that represents the color. Thus:

- «if the item number is ICADER34, the Length is 8, 8
■2 = 6, and the formula returns the 6th character, R.»
- «if the item number is ICPLB21, the Length is 7, 7
■2 = 5, and the formula returns the 5th character, B.»

Expanded example(s) using this function

Formula 14

LowerCase

Format

LowerCase(x)

«where x is a text value in a data field or a text string.»

Action

LowerCase prints the text string or text value in the data field in all lower case letters.

Typical use(s)

A good use of this function is when a field contains both uppercase and lowercase letters and you want to convert all values to lowercase for consistency.

Example(s)

```
LowerCase("Description") = "description"
LowerCase("BrEaD "  +  "AND "  +  "bUtTeR") =  "bread and butter"
```

Comments

Numbers that are part of the text are not affected by the LowerCase function.

Maximum(field)

Format

Maximum(field)

«where (*field*) is a field or formula.»

Action

Maximum(field) returns the highest value in the *field* for the entire report. It is, in effect, a "grand total" maximum.

Typical use(s)

Use Maximum(field) any time you need to print the highest value in a field or use the number in a calculation or comparison.

Example(s)

```
Maximum({file.Amount})
```

«returns the highest value in the *{file.Amount}* field.»

Maximum(field, condField)

Format

Maximum(field, condField)

- *field* is the name of the field for which Crystal Reports generates the summary field, and
- *condField* is the name of the character, number, or dollar value field that triggers the summary field to print whenever its value changes.

Action

Calculates the highest value in each group that results from the specified summary field. This is a group maximum, the maximum value in a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field that:

- returns (finds and reports) the highest (maximum) value in a group, and
- that uses a character, number, or dollar value field as the sort and group by field.

Example(s)

```
Maximum({file.Orders},{file.Customer}) =
```

«Identifies the largest order in each group of orders in the *{file.Orders}* field (the largest order for each month). The orders are separated into groups whenever the value in the *{file.Customer}* field changes.»

```
Maximum({file.Amount}, {file.State}) % Maximum({file.Amount})
```

«Groups values in the *{file.Amount}* field by state, and calculates the highest value for each state group as a percentage of the highest value for the entire report.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (maximum).

Maximum(field, condField, "condition")

Format

Maximum(field, condField, "condition")

- *field* is the name of the field for which Crystal Reports generates the summary field value,
- *condField* is the name of the date or Boolean field that triggers the summary field to print whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met.

Action

Calculates the highest value in each group that results from the specified summary field. This is a group maximum, the maximum value in a group of values in a given field. This function works just like Maximum(field, condField), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that returns (finds and reports) the highest (maximum) value in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
Maximum({file.Order},{file.Date}, "monthly") =
```

«Identifies the largest order in each group of orders in the *{file.Orders}* field (the largest order for each month). The orders are separated into groups whenever the value in the *{file.Date}* field changes to a new month.»

```
Maximum({file.Amount},{file.date}, "monthly") % Maximum({file.Amount})
```

«Groups values in the *{file.Amount}* field by month, and calculates the highest value for each month group as a percentage of the highest value for the entire report.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (maximum).

Maximum([array])

Format

Maximum([x,...])

«where *x* is an array of values separated by commas.»

Action

Maximum evaluates an array of constants, data field values, or formula results and returns the highest value in the array.

Typical use(s)

Use Maximum([array]) any time you have an array of values and need to use the highest of those values in a calculation or comparison.

Maximum([array]) also allows you to set a floor limit, a limit beneath which a calculation must not go.

Example(s)

```
Maximum([25, 50, 75,100]) = 100
```

```
Maximum([{file.Period 8}, {file.Period 9}, {file.Period 10}]) = 100.00
```

You might use this example to return the maximum account balance during three specific periods. Maximum allows you to set a floor on a calculation. For example:

```
Maximum([{file.Balance}, 500])
```

This formula sets a floor of 500 on the calculation. The expression will always return the balance unless the balance drops below 500. Then it will return the floor amount of 500. Thus:

```
Maximum([{file.Profits}, 500]) = {file.Profits}
```

«where {file.Profits} > 500»

and

```
Maximum([{file.Profits}, 500]) = 500
```

«where *{file.Profits}* < 500»

This kind of formula could be used if an employee recreation hall receives as its monthly budget, $500 or the profits from employee vending machines, whichever is higher.

Expanded example(s) using this function

Formula 7

Formula 11

Minimum(field)

Format

Minimum(field)

«where *field* is a database field or a formula.»

Action

Minimum(field) returns the lowest value in the field for the entire report. It is, in effect, a "grand total" minimum.

Typical use(s)

Use Minimum(field) any time you need to print the lowest value in a field or use the number in a calculation or comparison.

Example(s)

```
Minimum({file.Amount}) =
```

«returns the lowest value in the *{file.Amount field}*.»

Minimum(field, condField)

Format

Minimum(field, condField)

- *field* is the name of the database field or formula for which Crystal Reports generates the summary field value, and
- *condField* is the name of the character, number, or dollar value field or formula that triggers the summary field to print whenever its value changes.

Action

Minimum(field, condField) calculates the lowest value in each group that results from the specified summary field. This is a group minimum, the minimum value in a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that returns (finds and reports) the lowest (minimum) value in a group, and
- that uses a character, number, or dollar value field as the sort and group by field.

Example(s)

```
Minimum({file.Orders}, {file.Customer}) =
```

«Identifies the smallest order in each group of orders in the *{file.Orders}* field (the smallest order for each month). The orders are separated into groups whenever the value in the *{file.Customer}* field changes.»

```
(Minimum({file.Amount}, {file.State})) % (Minimum({file.Amount}))
```

«Groups values in the *{file.Amount}* field by state, and calculates the minimum value for each state group as a percentage of the minimum value for the entire report.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (minimum).

Expanded example(s) using this function

Formula 12

Minimum(field, condField, "condition")

Format

Minimum(field, condField, "condition")

- *field* is the name of the database field or formula for which Crystal Reports generates the summary field value,
- *condField* is the name of the date or Boolean field that triggers the summary field to print whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met.

Action

Minimum(field, condField, "condition") calculates the lowest value in each group that results from the specified summary field. This is a group minimum, the minimum value in a group of values in a given field. This function works just like Minimum(field, condField), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that returns (finds and reports) the lowest (minimum) value in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
Minimum({file.Order},{file.Date}, "monthly") =
```

«Identifies the smallest order in each group of orders in the *{file.Orders}* field (the smallest order for each month). The orders are separated into groups whenever the value in the *{file.Date}* field changes to a new month.»

```
Minimum({file.Amount}, {file.date}, "monthly") % Minimum({file.Amount})
```

«Groups values in the *{file.Amount}* field by month, and calculates the lowest value for each month group as a percentage of the lowest value for the entire report.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (minimum).

Minimum([array])

Format

Minimum([array])

«where *array* is an array of values separated by commas.»

Action

Minimum([array]) evaluates a array of constants, data field values, or formula results and returns the lowest value in the array.

Typical use(s)

Minimum also allows you to set a ceiling limit, a limit above which a calculation must not go.

Example(s)

```
Minimum([25, 50, 75, 100]) = 25
```

```
Minimum([{file.Period 8}, {file.Period 9}, {file.Period 10}]) = 45.00
```

«where *{file.Period 8}* = 45.00, *{file.Period 9}* = 100.00, *{file.Period 10}* = 50.00»

You might use this example to return the minimum account balance during three specific periods.

Minimum allows you to set a ceiling on a calculation. For example:

```
Minimum([{file.Commission}, 2500])
```

returns the commission accrued up to a cap or ceiling of 2500. Once accrued commission passes the $2500 mark, this expression returns 2500. Thus:

```
Minimum([{file.Commission}, 2500]) = 1575
```

«where *{file.Commission}* = 1575»

and

```
Minimum([{file.Commission}, 2500]) = 2500
```

«where *{file.Commission}* = 3250 (or any number in excess of 2500)»

Sales managers often use commission plans like this to keep commissions within a controllable range.

Month

Format

Month(x)

«where x is a date or a field that has a date as a value.»

Action

Month extracts the month component of a date and converts it to a number.

Typical use(s)

Use any time you need only the month component of a date. For example, if you're tracking payments that fall within a given year, you're interested only in the month they arrive; day and year information would be redundant. Also, if you need to use the number of the month in a numeric calculation (Month(*{file.OctPmt}*) ■ Month(*{file.SepPmt}*), for example) use the Month function to extract the month component of the date and convert it to a number.

Example(s)

```
Month({file.Last invoice on}) = 10
```

«where the last invoice date was in October.»

```
Month({file.PaymentDate}) ■ Month({file.InvoiceDate}) =
```

«calculates the number of months between the invoice and the payment.»

```
If Month(*{file.LastPurchase}*) < 6 Then
"No doubt you'll see some changes since
 you last shopped our store."
Else
"As you have certainly noticed, we've been
making many changes in our store recently."
```

«Prints "No doubt you'll see some changes since you last shopped our store." for those customers whose last purchase was in May or earlier, and prints "As you have certainly noticed, we've been making many changes in our store recently." for customers who have purchased more recently than May.»

Next

Format

Next (fld)

«where *fld* is any field or formula field in the report»

Action

Next(fld) returns the value of the specified field for the next record.

Typical use(s)

You can use Next to identify the first or last record in a range, to identify the starting point for a new group, or to test for duplicate values.

Example(s)

```
If Next({file.QTY})<>0 Then
{file.QTY}/2
Else
{file.QTY}
```

«tests the next value in the *{file.QTY}* field to see if it is a zero value. If it is not, it divides the value by two. If it is a zero value, it returns the value itself.»

```
If Remainder(Next ({file.SerialNum}), 300) = 0
Then
"End of block"
Else
""
```

«divides the next value in the *{file.SerialNum}* field by 300. If there is no remainder, it flags the current value as "End of block". If there is a remainder (else) it prints no flag. (This formula divides serial numbers into blocks of 300)»

NextIsNull

Format

NextIsNull(fld)

«where *fld* is any field or formula field in the report.»

Action

Evaluates the field specified in the next record and returns a TRUE if the field contains a null value.

Typical use(s)

You can use this function to test for the last item in a list and to take some action when you identify that last item.

Example(s)

```
If NextIsNull ({file.EmpNo}) Then
"Last employee of record"

Else
""
```

«in an employee database with no null values in the employee number field, flags the last employee on the list»

NumericText

Format

NumericText(x)

«where x is a text field you wish to test to see if it contains a number stored as text.»

Action

NumericText tests to see if the content of a text field is a number.

- If the entire content of the field is a number
- or if the characters extracted via the subscript operators are entirely a number
- the expression returns a YES value.
- If any part of the content of the field
- or of the characters extracted
- are not a number, the expression returns the value NO.

Typical use(s)

If you store numbers (like weight) in a text field, you would use NumericText to check the value of each record to make sure it is OK to convert using ToNumber.

Example(s)

```
NumericText({file.Reference}) = NO
```

«where *{file.Reference}* = "ABCDEFG"»

```
NumericText({file.IDNUM}) = YES
```

«where *{file.IDNUM}* = "12345"»

```
NumericText ({file.IDNUM} [1 to 5]) = YES
```

«where *{file.IDNUM}* = "12345■443"»

```
NumericText({file.IDNUM}) = NO
```

«where *{file.IDNUM}* = "12345T"»

NOTE: You can use this function in combination with ToNumber to test for a number in the Reference field, then print the string as a number or print 0 if the text string is not a number.

```
If NumericText({file.Reference}) Then
ToNumber({file.Reference})
Else
0
```

Expanded example(s) using this function

Formula 9

PageNumber

Format

PageNumber

«no arguments required»

Action

PageNumber inserts the current page number as a field in a formula.

Typical use(s)

Use this function any time you want Crystal Reports to determine the page number, at print time, and use that number in calculating formula results. For example, you can use this function when you want Crystal Reports to print something on certain pages and not on others.

Example(s)

```
PageNumber
```

«creates a field containing the current page number that you can place anywhere on the page. Using this formula is the equivalent of using the Insert|Special Field|Page Number Field command.»

```
If PageNumber > 1 Then
PageNumber
Else
0
```

«creates a field that prints the page number on every page but the first. By formatting this formula field using Suppress if Zero = active, the formula will print nothing on the first page instead of a zero.»

NOTE: You don't have to have a page number field in your report to use the PageNumber function in a report formula.

PopulationStdDev(fld)

Format

PopulationStdDev(*fld*)

«where *fld* is a number or currency field or formula».

Action

PopulationStdDev(*fld*) calculates the population standard deviation of the number or dollar values in the field for the entire report. It calculates the grand total population standard deviation for that field.

Typical use(s)

Use any time you need to calculate the population standard deviation of the values in a number or currency field, or use the population standard deviation in a calculation or comparison.

Example(s)

```
PopulationStdDev({file.Quantity})
```

«Calculates the grand total population standard deviation for all values in the field *Quantity*».

Comment

Crystal Reports uses *N* when calculating population standard deviation, *N*−*1* when calculating standard deviation.

PopulationStdDev(fld, condFld)

Format

PopulationStdDev(*fld, condFld*)

- *fld* is the name of the number or currency field for which Crystal Reports generates a summary field, and
- *condFld* is the name of the string, number, or currency field that triggers the summary field to print whenever its value changes

Action

PopulationStdDev(*fld, condFld*) calculates the population standard deviation for the values in each group that results from the specified summary field. This is a group population standard deviation, the population standard deviation of a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that calculates the population standard deviation for the values in a group, and
- that uses a string, number, or currency field as the sort and group by field.

Example(s)

```
PopulationStdDev({file.Amount}, {file.State}) %
PopulationStdDev({file.Amount})
```

«Groups values in the *Amount* field by state, and calculates the population standard deviation for each state group as a percentage of the population standard deviation for the values for the entire report».

Comments

- Crystal Reports uses *N* when calculating population standard deviation, *N*
1 when calculating standard deviation.
- In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (population standard deviation).

PopulationStdDev(fld, condFld, "condition")

Format

PopulationStdDev(*fld, condFld, "condition"*)

- *fld* is the name of the number or currency field for which Crystal Reports generates subtotals/summary fields,
- *condFld* is the name of the date or Boolean field that triggers the subtotals/summary fields to print whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met.

Action

PopulationStdDev(*fld, condFld, "condition"*) calculates the population standard deviation for the values in each group that results from the specified summary field. This is a group population standard deviation, the population standard deviation of a group of values in a given field). This function works just like PopulationStdDev(*fld*, *condFld*), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that calculates the population standard deviation for the values in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
PopulationStdDev({file.Scores}, {file.naturalized}, "any change")
```

«Groups values in the *Scores* field based on whether or not the test taker is a naturalized citizen, and calculates the population standard deviation for each group of scores.»

```
PopulationStdDev({file.Scores}, {file.Date}, "weekly") %
PopulationStdDev({file.Scores})
```

«Groups values in the *Scores* field by week, and calculates the population standard deviation for each group of scores as a percentage of the population standard deviation of the scores for the entire report (for all dates listed)».

Comments

- Crystal Reports uses *N* when calculating population standard deviation, *N*
- *1* when calculating standard deviation.
- In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (population standard deviation).

PopulationStdDev([array])

Format

PopulationStdDev(*[array]*)

«where *array* is an array of numeric values, separated by commas».

Action

PopulationStdDev(*[array]*) calculates the population standard deviation for an array of numeric constants, data field values, or formulas (*a*b, c/d*, etc.), separated by commas.

Typical use(s)

Use PopulationStdDev(*[array]*) any time you need to use the population standard deviation of values in an array in a calculation or comparison.

Example(s)

```
PopulationStdDev([2,4,6,8,10]) = 2.83
```

```
PopulationStdDev([({file.Qty1} * {file.Price1}), ({file.Qty2} *
{file.Price2}), ({file.Qty3}) * ({file.Price3}), ({file.Qty4} *
{file.Price4})]) = 31.70
```

«where Qty1 = 2, Price1 = 10.00, Qty2 = 2, Price2 = 2.00, Qty3 = 10, Price3 = 3.00, and Qty4 =8 , Price4 = 11.00»

Comment

Crystal Reports uses *N* when calculating population standard deviation, *N■1* when calculating standard deviation.

PopulationVariance(fld)

Format

PopulationVariance(*fld*)

«where *fld* is a number or currency field or formula».

Action

PopulationVariance(*fld*) calculates the population variance of the number or currency values in the field for the entire report. It calculates the grand total population variance for that field.

Typical use(s)

Use any time you need to calculate the variance of the values in a number or currency field, or use the variance in a calculation or comparison.

Example(s)

```
PopulationVariance({field.Quantity})
```

«Calculates the grand total population variance for all values in the field *Quantity*».

Comment

Crystal Reports uses *N* when calculating population variance, *N*−*1* when calculating variance.

PopulationVariance(fld, condFld)

Format

PopulationVariance(*fld, condFld*)

- *fld* is the name of the number or currency field for which Crystal Reports generates subtotals/summary fields, and
- *condFld* is the name of the string, number, or currency field that triggers the subtotals/summary fields to print whenever its value changes.

Action

PopulationVariance(*fld, condFld*) calculates the population variance for the values in each group that results from the specified summary field. This is a group population variance, the population variance of a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that calculates the population variance for the values in a group, and
- that uses a string, number, or currency field as the sort and group by field.

Example(s)

```
PopulationVariance({file.Results},{file.Procedure})
```

«Calculates the population variation for each group of laboratory test results in the *Results* field. The test results are separated into groups whenever the value in the *Procedure* field changes».

Comments

- Crystal Reports uses N when calculating population variance, N
1 when calculating variance.
- In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (population variance).

PopulationVariance(fld, condFld, "condition")

Format

PopulationVariance(*fld, condFld, "condition"*)

- *fld* is the name of the number or currency field for which Crystal Reports generates subtotals/summary fields,
- *condFld* is the name of the date or Boolean field that triggers the subtotals/summary fields to print whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met.

Action

PopulationVariance(*fld, condFld, "condition"*) calculates the population variance for the values in each group that results from the specified summary field. This is a group population variance, the population variance of a group of values in a given field. This function works just like PopulationVariance(*fld*, *condFld*), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that calculates the population variance for the values in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
PopulationVariance({file.Results},{file.Date}, "daily") =
```

«Calculates the population variation for each group of laboratory test results in the *Results* field. The test results are separated into groups whenever the value in the *Date* field changes to a new day».

```
PopulationVariance({file.Scores}, {file.Date}, "weekly") %
PopulationVariance({file.Scores})
```

«Groups values in the *Scores* field by week, and calculates the population variance for each group of scores as a percentage of the population variance of the scores for the entire report (for all dates listed)».

Comments

- Crystal Reports uses *N* when calculating population variance, *N*
- *1* when calculating variance.
- In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (population variance).

PopulationVariance([array])

Format

PopulationVariance(*[array]*)

«where array is an array of numeric values, separated by commas».

Action

PopulationVariance(*[array]*) calculates the statistical population variance for an array of numeric constants, data field values, or formulas (*a*b, c/d*, etc.), separated by commas.

Typical use(s)

Use PopulationVariance(*[array]*) any time you need to use the population variance of values in an array in a calculation or comparison.

Example(s)

```
PopulationVariance([2,4,6,8,10]) = 8.00
```

```
PopulationVariance([({file.Qty1}* {file.Price1}), ({file.Qty2}*
{file.Price2}), ({file.Qty3}* {file.Price3}), {file.Qty4}*
{file.Price4})]) = 1004.75
```

«where Qty1 = 2, Price1 = 10.00, Qty2 = 2, Price2 = 2.00, Qty3 = 10, Price3 = 3.00, and Qty4 =8, Price4 = 11.00»

Comment

Crystal Reports uses *N* when calculating population variance, *N■1* when calculating variance.

Previous

Format

Previous (fld)

«where fld is any field or formula field in the report»

Action

Previous(fld) returns the value of the specified field for the previous record.

Typical use(s)

You can use previous to identify the last record in a previous range or the last record occurring before a new range begins, or to test for duplicate values.

Example(s)

```
If Previous({file.QTY})<>0 Then
{file.QTY}/2
Else
{file.QTY}
```

«tests the previous value in the *{file.QTY}* field to see if it is a zero value. If it is not, it divides the value by two. If it is a zero value, it returns the value itself.»

```
If Previous ({file.CustNum}) = {file.CustNum} Then
"Repeated Value"
Else
""
```

«flags repeated values in the *{file.CustNum}* field»

```
If Remainder(Previous ({file.SerialNum}), 300) = 0 Then
"Beginning, new block"
Else
""
```

«divides the previous value in the *{file.SerialNum}* field by 300. If there is no remainder, it flags the current value as "Beginning, new block". If there is a remainder (else) it prints no flag. (This formula divides serial numbers into blocks of 300)»

PreviousIsNull

Format

PreviousIsNull(fld)

«where *fld* is any numeric field or numeric formula field in the report.»

Action

Evaluates the field specified in the previous record and returns a TRUE if the field contains a null value.

Typical use(s)

You can use this function to test for the first item in a list and to take some action when that first item is identified.

Example(s)

```
If PreviousIsNull ({file.EmpNo}) Then
"First employee of record"
Else
""
```

«in an employee database with no null values in the employee number field, flags the first employee on the list»

PrintDate

Format

PrintDate

«no arguments required»

Action

PrintDate inserts the date the report is printed as a field in a formula.

Typical use(s)

You can use this function any time you want to print the date of printing on the report, or to condition something in the report on the print date.

Example(s)

```
PrintDate
```

«creates a field containing the date the report is printed that you can place on your report. Using this formula is the equivalent of using the Insert|Special Field|Print Date Field command.»

```
If PrintDate >= Date(1992,01,01) Then
"Please excuse the delayed report."
Else
""
```

«prints the message apologizing for the delay if the report prints after January 1, 1992, and prints nothing if the report prints before that date.»

RecordNumber

Format

Record number

«no arguments required»

Action

RecordNumber inserts the current record number as a field in a formula.

Typical use(s)

You can use this function in creating record selection formulas to print some records and exclude others.

Example(s)

```
RecordNumber
```

«creates a field containing the current record number that you can place on your report. Using this formula is the equivalent of using the Insert|Special Field|Record Number Field command.»

Remainder

Format

Remainder(numerator, denominator)

Action

Remainder returns the remainder after the numerator (dividend) has been divided by the denominator (divisor). In a typical division situation, Crystal Reports expresses a quotient as a whole number (if any) and up to six decimal places. When using Remainder, however, Crystal Reports performs the division internally, determines the whole number quotient and the remainder, and returns only the remainder.

Typical use(s)

You can use this function in making conversions (feet to miles, units to grosses, etc.) You can also use it to select every *n*th item out of an array.

Example(s)

```
Remainder(12,5) = 2
```

```
If Remainder({file.Exam#}, 7) = 0 Then
"*****"
Else
""
```

«flags every 7th exam for grading by a second party»

```
ToText(Truncate({file.Days}/7)) + " week(s), "  +
ToText(Remainder({file.Days},7)) + " day(s)"  =
"9 week(s), 1 day(s)"
```

«(if *{file.Days}* = 64)converts days to weeks and days»

Expanded example(s) using this function

Formula 13

Formula 17

ReplicateString(x, n)

Format

ReplicateString(*x, n*)

«where *x* is a string and *n* is an integer».

Action

Prints string *x*, *n* number of times.

Typical use(s)

You can use this function to insert a line of characters any time they are needed. Some typical uses are:

- to flag critical data on your report,
- to build simple bar graphs,
- to split your report into visible sections, and
- to highlight totals, subtotals, and other summary data.

Example(s)

```
If {file.Sales} << {file.Quota} Then
ReplicateString("*",10)
Else
""
```

«Prints the string "*" ten times as a flag».

```
{file.Name}+ "  " + ReplicateString("*", {file.Score})
```

«prints a simple bar graph showing test results. The formula prints an asterisk for each point in a test score (the value in the file.Score field).

Comments

You must enclose the string in quotation marks.

Round

Format

Round(x)

«where x is the number ■ or arithmetic expression yielding a number ■ to be rounded»

Action

Round rounds to the nearest whole number. If the value to the right of the decimal point is .499 or below, Crystal Reports rounds to the next lowest number. If the value to the right of the decimal point is .5 or above, Crystal Reports rounds to the next highest number.

Typical use(s)

Use Round any time an approximate integer value will suffice instead of a value with many decimal places.

Example(s)

```
Round(1.23456) = 1
Round(1.499) = 1
Round({file.Amount}) = 1854.00
```

«where *{file.Amount}* = 1854.49»

```
Round({file.Amount}) = 1854.00
```

«where *{file.Amount}* = 1854.51)»

```
Round((A*B)/C) = 11
```

«where A= 25, B = 3, and C = 7»

Expanded example(s) using this function

Formula 8

Round(x, # places)

Format

Round(x, n)

> «where *x* is a number or dollar amount and *n* is the number of places to which you want the value rounded.»

Action

Round(x, n) rounds a number *x* to the number of decimal places specified by *n*. The number *n* may be positive, negative, or zero (0).

Typical use(s)

Crystal Reports' Round function rounds to the closest whole number. Use the Round(x, n) function any time you need greater precision in your rounding.

Example(s)

```
Round(2345.23456,4) = 2345.2346

Round(2345.23456,■1) = 2350
Round(2345.23456,■2) = 2300
Round(2345.23456,■3) = 2000
Round(1.234499,3) = 1.234
Round({file.Amount},1) = 1854.5
```

> «where *{file.Amount}* = 1854.49»
>
> ```
> Round({file.Wage} * {file.Hours worked}),2) = $146.63
> ```
>
> «where *{file.Wage}* = $5.75 and *{file.Hours worked}* = 25.5: 5.75 * 25.5 = 146.625 = 146.63 (when rounded to two decimal places)»

StdDev(fld)

Format

StdDev(*fld*)

«where *fld* is a number or currency field or formula».

Action

Standard Deviation(*fld*) calculates the standard deviation of the number or dollar values in the field for the entire report. It calculates the grand total standard deviation for that field.

Typical use(s)

Use any time you need to calculate the standard deviation of the values in a number or currency field, or use the standard deviation in a calculation or comparison.

Example(s)

```
StdDev({file.Quantity})
```

«Calculates the grand total standard deviation for all values in the field *Quantity*».

Comment

Crystal Reports uses *N*■*1* when calculating standard deviation, *N* when calculating population standard deviation.

StdDev(fld, condFld)

Format

StdDev(*fld, condFld*)

- *fld* is the name of the number or currency field for which Crystal Reports generates subtotals/summary fields, and
- *condFld* is the name of the string, number, or currency field that triggers the subtotals/summary fields to print whenever its value changes.

Action

Standard Deviation(*fld, condFld*) calculates the standard deviation for the values in each group that results from the specified summary field. This is a group standard deviation, the standard deviation of a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that calculates the standard deviation for the values in a group, and
- that uses a string, number, or currency field as the sort and group by field.

Example(s)

```
Standard Deviation({file.Amount}, {file.State}) % Standard
Deviation({file.Amount})
```

«Groups values in the Amount field by state, and calculates the standard deviation for each state group as a percentage of the standard deviation for the values for the entire report».

Comments

- Crystal Reports uses *N*
1 when calculating standard deviation, *N* when calculating population standard deviation.
- In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (standard deviation).

StdDev(fld, condFld, "condition")

Format

StdDev(*fld, condFld, "condition"*)

- *fld* is the name of the number or currency field for which Crystal Reports generates subtotals/summary fields,
- *condFld* is the name of the date or Boolean field that triggers the subtotals/summary fields to print whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met

Action

Standard Deviation(*fld, condFld, "condition"*) calculates the standard deviation for the values in each group that results from the specified summary field. This is a group standard deviation, the standard deviation of a group of values in a given field). This function works just like standard deviation(*fld*, *condFld*), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that calculates the standard deviation for the values in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
Standard Deviation({file.Scores}, {file.Date}, "weekly") % Standard
Deviation({file.Scores})
```

«Groups values in the *Scores* field by week, and calculates the standard deviation for each group of scores as a percentage of the standard deviation of the scores for the entire report (for all dates listed)».

Comments

- Crystal Reports uses *N*
1 when calculating standard deviation, *N* when calculating population standard deviation.
- In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (standard deviation).

StdDev([array])

Format

StdDev(*[array]*)

«where *array* is an array of numeric values, separated by commas».

Action

Standard deviation(*[array]*) calculates the standard deviation for an array of numeric constants, data field values, or formulas (*a*b, c/d*, etc.), separated by commas.

Typical use(s)

Use Standard deviation(*[array]*) any time you need to use the standard deviation of values in an array in a calculation or comparison.

Example(s)

```
Standard deviation([2,4,6,8,10]) = 3.16
```

```
Standard deviation([{file.Qty1}, {file.Qty2}, {file.Qty3}, {file.Qty4}])
= 4.12
```

«where Qty1 = 2, Qty2 = 2, Qty3 = 10, and Qty4 = 8»

```
Standard deviation([({file.Qty1} * {file.Price1}),({file.Qty2}
*{file.Price2}), ({file.Qty3}*{file.Price3}), ({file.Qty4} *
{file.Price4})]) = 36.60
```

«where Qty1 = 2, Price1 = 10.00, Qty2 = 2, Price2 = 2.00, Qty3 = 10, Price3 = 3.00, and Qty4 =8, Price4 = 11.00»

Comment

Crystal Reports uses *N■1* when calculating standard deviation, *N* when calculating population standard deviation.

Sum(field)

Format

Sum(field)

«where *field* is a number or dollar value field or formula.»

Action

Sum(*field*) totals the number or dollar values in the *field* for the entire report. It calculates the grand total for that field.

Typical use(s)

Use any time you need to print the sum of the values in a number or dollar value field, or use the sum in a calculation or comparison.

Example(s)

```
Sum({file.Amount}) =
```

«calculates the sum of all values in the *{file.Amount field}*.»

Sum(field, condField)

Format

Sum(field, condField)

- *field* is the name of the number or dollar value field for which Crystal Reports generates subtotals/summary fields, and
- *condField* is the name of the character, number, or dollar value field that triggers the subtotals/summary fields to print whenever its value changes.

Action

Sum(field, condField) sums (totals) the values in each group that results from the specified subtotal or summary field. This is a group sum (a subtotal), the sum of a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a subtotal or summary field:

- that sums (totals) the values in a group, and
- that uses a character, number, or dollar value field as the sort and group by field.

Example(s)

```
Sum({file.Orders}, {file.Customer}) =
```

«Sums (totals) the orders in each group of orders in the *Orders* field. The orders are separated into groups whenever the value in the *Customer* field changes.»

```
Sum({file.Amount}, {file.State}) % Sum({file.Amount})
```

«Groups values in the *Amount* field by state, and calculates the sum of values for each state group as a percentage of the sum of values for the entire report.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a subtotal or other group field in your report with identical parameters: same field, same sort and group by field, same action (sum).

Sum(field, condField, "condition")

Format

Sum(field, condField, "condition")

- *field* is the name of the number or dollar value field for which Crystal Reports generates subtotals/summary fields,
- *condField* is the name of the date or Boolean field that triggers the subtotals/summary fields to print whenever a certain condition is met, and
- *condition* is the condition(weekly, monthly, "change to Yes", "next is No", etc.) that needs to be met.

Action

Sum(field, condField, "condition") sums (totals) the values in each group that results from the specified subtotal or summary field. This is a group sum (a subtotal, the sum of a group of values in a given field). This function works just like Sum(*field, condField*), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a subtotal or summary field:

- that sums (totals) the values in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
Sum({file.Order},{file.Date}, "monthly") =
```

«Sums (totals) the orders in each group of orders in the *{file.Orders}* field (the total of all orders for each month). The orders are separated into groups whenever the value in the *{file.Date}* field changes to a new month.»

```
Sum({file.Amount},{file.date}, "monthly") % Sum({file.Amount})
```

«Groups values in the *{file.Amount}* field by month, and calculates the sum of the values for each month group as a percentage of the sum of the values for the entire report.»

Comment

In order to use this function to insert a group field in a formula, you must have already entered a subtotal or other group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (sum).

Sum([array])

Format

Sum([array])

«where *array* is an array of numeric values, separated by commas.»

Action

Sum([array]) adds the values in an array of numeric constants, data field values, or formulas (*a*b, c/d*, etc.), separated by commas.

Typical use(s)

Use Sum([array]) any time you need to use the sum of values in an array in a calculation or comparison.

Example(s)

```
Sum([5, 5, 5, 5]) = 20

Sum([{file.ItemA Price} * {file.QuantityA}, {file.ItemB Price} *
{file.QuantityB}]) = $47.50
```

«where *{file.ItemA Price}* = 5.00, *{file.QuantityA}* = 3, *{file.ItemB Price}* = 6.50, *{file.QuantityB}* = 5 ■
(5.00 * 3) +
(6.50 * 5) = (15.00) + (32.50) = 47.50»

Expanded example(s) using this function

Formula 4

Today

Format

Today

Action

Today returns the current date.

Typical use(s)

Use Today any time you wish to insert today's date (taken from your computer's internal clock) into a formula.

Example(s)

```
Today = May 16 91
```

«where today's date is May 16, 1991)»

```
Day(Today) = 16
```

«where today's date is May 16, 1991»

Comment

If you simply want to print today's date on your report, use the PrintDate function.

ToNumber

Format

ToNumber(x)

«where *x* is a number stored as a text string»

Action

ToNumber converts a text string to a number.

In a database, some numbers are stored in numeric fields, as numbers, and some are stored in character fields, as text. You make the determination which fields are to be numeric and which are to be text when you set up the database in the first place. Numbers on which you might wish to perform arithmetic (item cost, quantity ordered, etc.) are typically stored in numeric fields; numbers on which you don't expect to perform arithmetic (customer number, telephone number, etc.) are typically stored in text fields.

ToNumber allows you to convert a number stored as text to a number on which you can perform arithmetic.

Typical use(s)

You might use this function, for example, if your item numbers contain coded product information and you want to use that information in calculations.

Example(s)

```
ToNumber("123.45") = 123.45
```

```
ToNumber({file.AcctNo}/2) = 22144
```

«where *{file.AcctNo}* = 44288

```
ToNumber({file.ZIP}) < 33333 = TRUE
```

«where *{file.ZIP}* is 21385"»

Expanded example(s) using this function

Formula 2

Formula 3

Formula 9

Formula 17

ToText

Format

ToText(x)

«where x is a number that you wish to convert to text.»

Action

ToText converts a number to a text string.

Typical use(s)

You can use this function to convert a numeric field value or the result of a numeric calculation to text so it can be used in a text string (form letter, comment on report, flag, etc.)

Example(s)

```
ToText(123.45) =  "123.45"
```

ToText is useful when you want to build a sentence by concatenating a converted number with other text strings.

```
"The base price of item #  "  + {file.Item number} +  " is $"  +
ToText({file.Base Price}) + "."
```

The above formula prints this sentence: "The base price of item A1/4520/B12 is $50.00."»

«where the *{file.Item number}* field contains A1/4520/B12" and the *{file.Base Price}* field contains 50.00, converted to text and formatted with two decimal places.»

NOTE: Use the TrimLeft and TrimRight functions to eliminate spaces before and after left-justified and right-justified text fields. For example:

```
TrimLeft({file.Item number})
```

will trim the spaces to the left of the item number, which is stored right-justified. The example for ToNumber also includes the use of the ToText function.

```
ToText({file.Amt} * {file.Quantity}) =  "44,890.20"
```

«where *{file.Amt}* = 24.45 and *{file.Quantity}* = 1836»

Expanded example(s) using this function

Formula 1

Formula 2

Formula 3

Formula 5

Formula 13

ToText (number of places)

Format

ToText(x, # places)

> «where *x* is a number you want to convert to text, and # *places* is a number specifying the number of decimal places.»

Action

ToText(x, # places) converts a number to a text string and allows you to specify the number of decimal places in the number when it is written as text.

Typical use(s)

Use ToText(x, # places) any time you want to convert a number to text and you don't want as many decimal places in the text version as in the decimal version (for example, when you're converting multiple numbers, each with a different number of decimal places, and you want a consistent number of decimal places in the text representation of the numbers).

Example(s)

```
ToText(12345.6749,2) = "12345.67"
ToText(12345.6750,2) = "12345.68"
ToText(12345.5000,0) = "12346"
```

Comments

This function does not truncate the number written as text, but rounds it to the number of decimal places specified. See Round(x, # places) for additional information.

ToWords(x)

Format

ToWords(*x*)

«where *x* is a number that you wish to convert to words (1 = one, 68 = sixty-eight, etc.».

Action

You can use this function to convert a number or currency field value or the result of a numeric calculation to words so it can be used as text.

The function treats the number as a whole, rather than as a series of individual numbers. That is, 123 is treated as the number one hundred twenty-three rather than the individual digits one, two, and three.

Negative numbers begin with the word "negative."

Typical use(s)

You can use this function to spell out the dollar amount for each check if you are using computer checks.

Example(s)

```
ToWords(12345) = twelve thousand three hundred forty-five and xx/100
ToWords(12.3499) = twelve and 35/100
ToWords(.98 * {file.Amount}) = two thousand sixteen and 84/100
```

«where Amount = 2058.00»

```
ToWords((({file.Qty1} + {file.Qty2} + {file.Qty3}) * {file.Price}) *
1.075) = one hundred two and 13/100
```

«where QTY1 = 1, Qty2 = 82, QTY3 = 12, and Price = 1.00 (sums three quantities, multiplies them times the price and adds 7.5% sales tax. In this case the numeric answer is 102.125 which is then rounded to 102.13 (the standard two decimal places) before putting into words».

Comments

- Currency field values and number field values are treated in the same way and produce identical results.
- Since the spelled out value will be much longer than the number value, you will need to increase the size of the field box to accommodate the new field length.

ToWords(x, number of places)

Format

ToWords(*x, n*)

«where *x* is a number that you want to convert to words (1 = one, 68 = sixty-eight, etc.) and *n* is the number of decimal places you want included in the resulting word version».

Action

You can use this function to convert a number or currency field value or the result of a numeric calculation to words so it can be used as text. The ability to adjust the number of decimal places can be useful when the number is the result of a calculation that may produce more decimal places than you want

Typical use(s)

You can use this function to spell out the dollar amount for each check if you are using computer checks.

Example(s)

```
ToWords(12345.6749,2) = twelve thousand three hundred forty-five and 67/100
ToWords(12345.5000,0) = twelve thousand three hundred forty six
```

«Note that the value is rounded to the number of decimal places specified».

```
ToWords((({file.Qty1} + {file.Qty2} + {file.Qty3}) * {file.Price}) * 1.075,0) = one hundred two
```

«where QTY1 = 1, Qty2 = 82, QTY3 = 12, and Price = 1.00 (sums three quantities, multiplies them times the price and adds 7.5% sales tax. In this case the numeric answer is 102.125 which is then rounded to 0 decimal places before putting into words».

Comments

- The function treats the number as a whole, rather than as a series of individual numbers. That is, 123 is treated as the number one hundred twenty-three rather than the individual digits one, two, and three.
- Negative numbers begin with the word "negative."
- Currency field values and number field values are treated in the same way and produce identical results.
- Since the spelled out value will be much longer than the number value, you will need to increase the size of the field box to accommodate the new field length.

TrimLeft

Format

TrimLeft(x)

«where x is a string or data field stored right justified."

Action

TrimLeft removes all spaces to the left of a string or data field which is stored as a right justified string in a database.

Typical use(s)

Use this function any time there are leading blanks in a text field that may interfere with an alignment of text strings, a character count, or with a calculation (if the string is eventually converted to a number).

Use whenever you are concatenating justified text strings and want to have the proper spacing between strings.

Example(s)

```
TrimLeft("  A1/4520/B12") =  "A1/4520/B12"
TrimLeft({file.Customer number}) =  "200"
```

«where the text string " 200" is right justified in the field.»

You have two right justified database fields, *{file.Food1}* and *{file.Food2}*. Each field can hold up to 15 characters. *{file.Food1}* contains the word *bread* and *{file.Food2}* contains the word *butter*. If you were to print these words, they would appear like this:

" bread"

" butter"

For each 15 character field the database includes the field value, justified right plus enough blank spaces to fill up the field. To use these words without the leading spaces (to create the expression bread and butter, for example) use the TrimLeft function in the following manner:

```
TrimLeft({file.Food1}) + " and " + TrimLeft({file.Food2}) = "bread and
butter"
```

NOTE: The spaces enclosed in the quotation marks before and after the word* and *assure that there is the correct spacing between the three words in the resulting sentence.

Expanded example(s) using this function

Formula 14

TrimRight

Format

TrimRight(x)

«where *x* is a string or data field stored left justified."

Action

TrimRight removes all spaces to the right of a string or data field which is stored left-justified in a database.

Typical use(s)

Use this function any time there are trailing blanks in a text field that may interfere with an alignment of text strings, a character count, or with a calculation (if the string is eventually converted to a number).

Example(s)

```
TrimRight("Al/4520/B12____ ") =  "A1/4520/B12"
TrimRight({file.Reference}) = "Bal Fwd."
```

«where the text string "Bal Fwd. " is left-justified in the field.»

Truncate

Format

Truncate(x)

«where x is the number you want truncated»

Action

Truncate returns a whole number (integer) by truncating the number at the decimal point.

Typical use(s)

Use this function whenever the characters to the right of the decimal are unimportant.

Example(s)

```
Truncate(1.23456) = 1
Truncate(1.499) = 1
Truncate(1.999) = 1
```

If you have 147 golf balls in stock and want to know how many dozen are available for sale, your calculation is 147/12=12.25 12.25 (truncated) = 12 dozen available for sale. If you sell balls only by the dozen, the .25 dozen you truncated is unimportant.

```
Truncate ({file.Ball inventory}/12) = 12
```

«where *{file.Ball inventory}* = 155» (155/12 = 12.92, 12.92 truncated = 12)

```
Truncate({file.Ball inventory}/12) = 13
```

«where *{file.Ball inventory}* = 157» (157/12 = 13.08, 13.08 truncated = 13)

Comments

This is *not* a rounding function; truncate simply deletes all characters to the right of the decimal point.

Expanded example(s) using this function

Formula 13

UpperCase

Format

Uppercase(x)

«where *x* is the text string you want converted to upper case.**»**

Action

UpperCase prints the text string or text value in the data field in upper case (capital letters).

Typical use(s)

A good use of this function is when a field contains both uppercase and lowercase letters and you want to convert all values to uppercase for consistency.

Example(s)

```
UpperCase("Description") =  "DESCRIPTION"
UpperCase("abc12345") = "ABC12345"
UpperCase("BrEaD "  + "AND "   +  "bUtTeR" ) =  "BREAD AND BUTTER"
```

Variance(fld)

Format

Variance(*fld*)

«where *fld* is a number or currency field or formula».

Action

Variance(*fld*) calculates the variance of the number or currency values in the field for the entire report. It calculates the grand total variance for that field.

Typical use(s)

Use any time you need to calculate the variance of the values in a number or currency field, or use the variance in a calculation or comparison.

Example(s)

```
Variance({field.Quantity})
```

«Calculates the grand total variance for all values in the field *Quantity*».

Comment

Crystal Reports uses *N*−*1* when calculating variance, *N* when calculating population variance.

Variance(fld, condFld)

Format

Variance(*fld, condFld*)

- *fld* is the name of the number or currency field for which Crystal Reports generates subtotals/summary fields, and
- *condFld* is the name of the string, number, or currency field that triggers the subtotals/summary fields to print whenever its value changes.

Action

Variance(*fld, condFld*) calculates the variance for the values in each group that results from the specified summary field. This is a group variance, the variance of a group of values in a given field.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that calculates the variance for the values in a group, and
- that uses a string, number, or currency field as the sort and group by field.

Example(s)

```
Variance({file.Results},{file.Procedure})
```

«Calculates the variation for each group of laboratory test results in the *Results* field. The test results are separated into groups whenever the value in the *Procedure* field changes».

Comments

- Crystal Reports uses *N*
1 when calculating variance, *N* when calculating population variance.
- In order to use this function to insert a group field in a formula, you must have already entered a group field in your report with identical parameters: same field, same sort and group by field, same action (variance).

Variance(fld, condFld, "condition")

Format

Variance(*fld, condFld, "condition"*)

- *fld* is the name of the number or currency field for which Crystal Reports generates subtotals/summary fields,
- *condFld* is the name of the date or Boolean field that triggers the subtotals/summary fields to print whenever a certain condition is met, and
- *condition* is the condition ("weekly", "monthly", "change to Yes", "next is No", etc.) that needs to be met.

Action

Variance(*fld, condFld, "condition"*) calculates the variance for the values in each group that results from the specified summary field. This is a group variance, the variance of a group of values in a given field). This function works just like Variance(*fld*, *condFld*), but, because it uses a date or Boolean field as a sort and group by field, it requires a condition in addition to the other arguments.

Typical use(s)

Use this function whenever you want to duplicate, in a formula, a summary field:

- that calculates the variance for the values in a group, and
- that uses a date or Boolean field as the sort and group by field.

Example(s)

```
Variance({file.Scores}, {file.Date}, "weekly") % Variance({file.Scores})
```

«Groups values in the *Scores* field by week, and calculates the variance for each group of scores as a percentage of the variance of the scores for the entire report (for all dates listed)».

Comments

- Crystal Reports uses *N*
1 when calculating variance, *N* when calculating population variance.
- In order to use this function to insert a group field in a formula, you must have already entered a subtotal or other group field in your report with identical parameters: same field, same sort and group by field, same date or Boolean condition, same action (variance).

Variance([array])

Format

Variance(*[array]*)

«where *array* is an array of numeric values, separated by commas».

Action

Variance(*[array]*) calculates the statistical variance for an array of numeric constants, data field values, or formulas (*a*b, c/d*, etc.), separated by commas.

Typical use(s)

Use Variance(*[array]*) any time you need to use the variance of values in an array in a calculation or comparison.

Example(s)

```
Variance([2,4,6,8,10]) = 10
```

«where Qty1 = 2, Qty2 = 2, Qty3 = 10, and Qty4 = 8»

```
Variance([({file.Qty1} * {file.Price1}),({file.Qty2}*{file.Price2}),
({file.Qty3}*{file.Price3}), ({file.Qty4} * {file.Price4})]) = 1339.67
```

«where Qty1 = 2, Price1 = 10.00, Qty2 = 2, Price2 = 2.00, Qty3 = 10, Price3 = 3.00, and Qty4 =8, Price4 = 11.00»

Comment

Crystal Reports uses *N■1* when calculating variance, *N* when calculating population variance.

WhileReadingRecords

Format

WhileReadingRecords

Action

Forces the program to evaluate the formula while it is reading database record data.

Typical use(s)

Formulas are normally evaluated at the following times:

- If no database or group field is included in the formula, the formula is evaluated before the program reads database records.
- If a database is included in the formula, the formula is evaluated while the program reads database records.
- If a group field, page # field, subtotal, etc. is included in the formula, the formula is evaluated after database records are read and while the data from the records is being printed in the report.

WhileReadingRecords forces the formula to be evaluated while the program reads database records. When this function is used in a formula, the Formula Checker returns an error message if you attempt to include elements in the formula (groups, page number fields, etc.) that must be evaluated at a later time (while printing records).This function can also be used to force a formula that includes no database fields and no group fields to process while reading records instead of before reading records.

Example(s)

```
WhileReadingRecords;
ToNumber({items.Qty})
```

«Forces the formula (which contains a database field) to be evaluated at its normal time (while reading records).»

NOTE: If you try to include a group in this formula, you will get an error message.

```
WhileReadingRecords;
ToNumber ("12345")
```

«Forces the formula (which contains no database fields or groups), to be evaluated later than it would normally be evaluated. In this case it causes the formula to be evaluated while reading records instead of before reading records.»

NOTE: If you try to include a group in this formula, you will get an error message.

See Also

Example report using evaluation time function

WhilePrintingRecords

Format

WhilePrintingRecords

Action

Forces the program to evaluate the formula while it is printing database record data.

Typical use(s)

Formulas are normally evaluated at the following times:

- If no database or group field is included in the formula, the formula is evaluated before the program reads database records.
- If a database is included in the formula, the formula is evaluated while the program reads database records.
- If a group field, page # field, subtotal, etc. is included in the formula, the formula is evaluated after database records are read and while the data from the records is being printed in the report.

WhilePrintingRecords forces the formula to be evaluated while the program prints database records.

This function can also force a formula that includes no database fields and no group fields to process while printing records instead of before reading records. It will also force a formula that includes database fields to process while printing records instead of while reading records.

Example(s)

```
WhilePrintingRecords;
3* Sum ({detail.Qty1}, {detail.Qty2})
```

«Forces the formula (which contains a group) to be evaluated at its normal time (while printing records)»

```
WhilePrintingRecords;
ToNumber ("12345")
```

«Forces the formula (which contains no database fields or groups), to be evaluated later than it would normally be evaluated. In this case it causes the formula to be evaluated while printing records instead of before reading records.»

See Also

Example report using evaluation time function

Year

Format

Year(x)

«where x is a date (yyyy, mm, dd) from which you want to extract the year.»

Action

Year extracts the year from a date and returns a number.

Typical use(s)

Use this function any time you need to use a year, converted to a number, in calculations or comparisons.

Example(s)

```
Year({file.Last invoice on}) = 1989
```

«where the last invoice date was in 1989.»

```
If Year({file.LastUpgrade}) < 1988 Then
"P.S. You're missing out on many of the
benefits of our new, improved version."
Else
""
```

«Prints a P.S. for those customers who last upgraded prior to 1988, and prints nothing for those customers who show a more recent upgrade.»

Example report using evaluation time function

This example report illustrates the use of an evaluation time function to take a formula that would normally be evaluated while reading records and force it instead to be evaluated while printing records.

Scenario

You want to create an order detail report that:

- shows the extended price for each line item,
- calculates a running total for each line, and
- resets itself to 0 for each new order.

Unless you disabled the *Samples and Examples* option during installation, the report, called EVALTIME.RPT, is one of the sample reports that was installed on your system. An example showing partial data from that report follows:

Order#	Item#	Qty	Price	Extension	RunTotal
Order #2203					
2203	1002	3	276.00	828.00	828.00
2203	1102	1	1,230.00	1,230.00	2,058.00
Total for order 2203	**2,058.00**				
Order # 2204					
2204	1001	7	192.00	1,344.00	1,344.00
2204	1002	3	276.00	828.00	2,172.00
2204	1003	1	484.00	484.00	2,656.00
2204	1102	3	1,230.00	3,690.00	6,346.00
Total for order 2204	**6,346.00**				

The report uses four database fields:

Order#	{detail.ORDERNUM}
Item#	{detail.ITEMNUM}
Qty	{detail.QTY}
Price	{detail.PRICE}

It uses three formulas:

```
@extend {detail.QTY} * {detail.PRICE}
```

«@extend simply calculates the extended price for a line item (quantity ordered times price per item). This provides the values in the *Extension* column.»

```
@initialize NumberVar Runtotal;

If {detail.ORDERNUM} <> Previous ({detail.ORDERNUM}) Then
Runtotal := 0
Else
Runtotal := Runtotal
```

«@initialize resets the variable *Runtotal* each time the order number changes so the running total for each order begins at 0.»

```
@runningWhilePrintingRecords;
NumberVar Runtotal;
```

```
Runtotal:= Runtotal + {@extend}
```

«@running calculates the running total for each order by adding the extended price for each line item to the existing running total for the order. This provides the values in the *Running Total* column.»

And it includes one subtotal:

```
Sum of @extend End group #1: detail.ORDER
Sum of @extend
```

«It subtotals the extended price each time the order number changes, thus providing the *Total for order nnnn* order total.»

Evaluation Times

Formulas are normally evaluated at the following times:

- If no database or group field is included in the formula, the formula is evaluated before the program reads database records.
- If a database is included in the formula, the formula is evaluated while the program reads database records.
- If a group field, page # field, subtotal, etc. is included in the formula, the formula is evaluated after database records are read and while the data from the records is being printed in the report.

The evaluation time problem which is solved with one of the evaluation time functions involves the relative evaluation time of two formulas, @initialize and @running.

- @initialize includes the Previous function which is evaluated while records are printed. Any formula that includes the Previous function, therefore, will be evaluated while the records are printing as well.
- @running, on the other hand, is normally evaluated while records are read. @running includes the formula @extend in its calculations, and @extend involves a database field. Whenever a database field is involved in a formula, the formula is evaluated while reading records (unless there is something in the formula
- a subtotal, a page # field, etc.
- that causes the formula to be evaluated later).

If we leave both formulas to be evaluated at their normal times,

- @running is evaluated first (during record read time) and it outputs running totals for each line item.
- Then, after it is finished calculating the running totals, @initialize is evaluated (during record print time). This formula initializes (sets to 0) the Runtotal variable each time the order number changes.

By this time it's too late. The running totals have already been calculated without being initialized between orders. The final printed report shows running totals getting bigger with each line item; they are not reset from order to order.

Your report comes out looking like this:

NOTE: Order detail report - @running doesn't include an evaluation time function.

Order#	Item#	Qty	Price	Extension	RunTotal
Order #2203					
2203	1002	3	276.00	828.00	828.00
2203	1102	1	1,230.00	1,230.00	2,058.00
Total for order 2203			**2,058.00**		
Order # 2204					
2204	1001	7	192.00	1,344.00	3,402.00
2204	1002	3	276.00	828.00	4,230.00
2204	1003	1	484.00	484.00	4,714.00

2204	1102	3	1,230.00	3,690.00	8,404.00

Total for order 2204 6,346.00

To solve this problem, you must make certain that @initialize is evaluated at the same time as @running. You can't force a print time evaluation formula (@initialize) to evaluate at read time (forcing it to be evaluated before the required data is available), but you can force a read time evaluation formula (@running) to evaluate later, at print time. You do this by starting the formula (as we did) with the function ***WhilePrintingRecords;***

```
WhilePrintingRecords;

NumberVar Runtotal;

Runtotal:= Runtotal + {@extend}
```

When you do this, your report comes out looking like you want it, like this:

NOTE: Order detail report @running includes evaluation time function.

Order#	Item#	Qty	Price	Extension	RunTotal
Order #2203					
2203	1002	3	276.00	828.00	828.00
2203	1102	1	1,230.00	1,230.00	2,058.00
Total for order 2203		**2,058.00**			
Order # 2204					
2204	1001	7	192.00	1,344.00	1,344.00
2204	1002	3	276.00	828.00	2,172.00
2204	1003	1	484.00	484.00	2,656.00
2204	1102	3	1,230.00	3,690.00	6,346.00
Total for order 2204		**6,346.00**			

See Also

BeforeReadingRecords

WhileReadingRecords

WhilePrintingRecords

FUNCTIONS INDEX (ALPHABETICAL)

Click on one of the buttons to find the function of your choice, or choose from the list below:

A

Abs(x)

Average([array])

Average(field, condField)

Average(field, condField, "condition")

Average(field)

B

BeforeReadingRecords

C

Count([array])

Count(field, condField)

Count(field, condField, "condition")

Count(field)

D

Date (yyyy, mm, dd)

Day (x)

DayOfWeek (x)

DistinctCount([array])

DistinctCount(field, condField)

DistinctCount(field, condField, "condition")

DistinctCount(field)

G

GroupNumber

I

IsNull

L

Length(x)

LooksLike(string, mask)

LowerCase (x)

M

Maximum([array])

Maximum(field, condField)

Maximum(field, condField, "condition")

Maximum(field)

Minimum([array])

Minimum(field, condField)

Minimum(field, condField, "condition")

Minimum(field)

Month (x)

N

Next

NextIsNull

NumericText(fieldname)

P

PageNumber

PopulationStdDev([array])

PopulationStdDev(field, condField)

PopulationStdDev(field, condField, "condition")

PopulationStdDev(field)

PopulationVariance([array])

PopulationVariance(field, condField)

PopulationVariance(field, condField, "condition")

PopulationVariance(field)

Previous

PreviousIsNull

PrintDate

R

RecordNumber

Remainder(numerator, denominator)

ReplicateString(x, n)

Round(x, # places)

Round(x)

S

StdDev([array])

StdDev(field, condField)

StdDev(field, condField, "condition")

StdDev(field)

Sum([array])

Sum(field, condField)

Sum(field, condField, "condition")

Sum(field)

T

Today

ToNumber (x)

ToText (x, # places)

ToText (x)

ToWords(x, # places)

ToWords(x)

TrimLeft (x)

TrimRight (x)

Truncate (x)

U

UpperCase (x)

V

Variance([array])

Variance(field, condField)

Variance(field, condField, "condition")

Variance(field)

W

WhilePrintingRecords

WhileReadingRecords

Y

Year (x)

OPERATORS INDEX

Operators are special symbols or words that describe an operation or an action to take place between two or more values.

Operators are used in formulas. Crystal Reports reads the operators in a formula and performs the actions specified.

Click on an operator from the list below for more information:

Arithmetic Operators

Conversion Operators

Comparison Operators

String Operators

Range Operators

Array Operators

Boolean Operators

Miscellaneous Operators

OPERATORS INDEX (BY TYPE)

Click on an operator from the list below for more information:

Arithmetic Operators

Add +

Subtract -

Multiply *

Divide /

Percentage %

Negate -()

Conversion Operators

To Currency $

Comparison Operators

Equal =

Not equal <>

Less than <

Greater than >

Greater than or equal >=

Less than or equal <=

String Operators

Concatenate +

Subscript []

In String in

Range Operators

Make range to

In range in

Boolean Operators

And and

Or or

Not not

Array Operators

Make array [,]

Subscript []

In Array in

Miscellaneous Operators

Assignment :=

Parentheses ()

If then else If■Then ■Else

Statement separator ;

Comment seperator //

Pattern Operators

Starts With

Like pattern

ARITHMETIC OPERATORS INDEX

Arithmetic operators are used to calculate **number** or **dollar** values.

Click on an operator from the list below for more information:

Add +

Subtract ▪

Multiply *

Divide /

Percentage %

Negate ▪()

CONVERSION OPERATORS INDEX

The conversion operator is used to convert one data type to another.

To Currency $

COMPARISON OPERATORS INDEX

Comparison operators are used to compare data in a data field with a constant, with the content of another data field, or with a formula result.

Click on an operator from the list below for more information:

Equal =

Not equal <>

Less than <

Greater than >

Greater than or equal >=

Less than or equal <=

STRING OPERATORS INDEX

String operators are used to Concatenate strings, to extract substrings from strings(**Subscript**), or to test for the presence of substrings in strings (**In String**).

NOTE: Crystal Reports operators are case sensitive. Thus, ABC is not equal to abc, and thus abc is not in the string ABCDEF, etc.

Click on an operator from the list below for more information:

Concatenate	+
Subscript	[]
In String	in

RANGE OPERATORS INDEX

Range operators are used to create ranges (**Make range**) and to see if a value is within the range created (**In range**). These operators test for consecutive values such as dates, text, or amounts which fall within a range.

Click on an operator from the list below for more information:

Make range	to
In range	in

BOOLEAN OPERATORS INDEX

Boolean operators are used to create conditions that require a logical relationship between two or more values. Conditions that use boolean operators are called boolean expressions.

- A **and** B means that **both** A and B must be true for the condition to be satisfied (to return a TRUE value);
- A **or** B means that **either** A or B (or both) must be true for the condition to be satisfied (to return a TRUE value), and
- A **not** B means that A **must be true** but that B **must not be true** for the condition to be satisfied (to return a TRUE value).

Click on an operator from the list below for more information:

And and

Or or

Not not

ARRAY OPERATORS INDEX

Array operators are used to build a list of data fields, constants, or text strings. These lists can then be used for checking to see if a field exists in the list, or for extracted elements by their position. Whereas the range operators are used to see if an item exists in a range of values, these operators allow you to see if an item exists in a set of non-contiguous values.

Click on an operator from the list below for more information:

Make array [,]

Subscript []

In Array in

MISCELLANEOUS OPERATORS INDEX

Miscellaneous operators are used to indicate an order in which calculations are to be performed or to build formulas setting conditions, that if met, trigger specific consequences.

Click on an operator from the list below for more information:

Assignment :=

Parentheses ()

If■Then

■Else If

■Then

■Else

Statement separator ;

Add Operator

Operator (Symbol/Word)

+

Usage

x+y

«Add values *x* and *y*»

Example(s)

```
5 + 6 = 11
```

```
{file.Qty1} + {file.Qty2} = 1487
```

«where *{file.Qty1}* = 366 and *{file.Qty 2}* = 1121»

```
{file.Amt1} + {file.Amt2} + {file.Amt3} + {file.Amt4} = 20
```

«where {file.Amt1} = 2, {file.Amt2} = 4, {file.Amt3} = 6, {file.Amt4} = 8»

```
{file.Class1} + 25 = 37
```

«where *{file.*Class1*}* = 12»

```
Date (1991, 04, 05) + 12 = Apr 17 91
```

Expanded example(s) using this operator

Formula 8

Formula 16

Subtract Operator

Operator (Symbol/Word)

■

Usage

x■y

«Subtract value *y* from value *x*»

Example(s)

```
      244 ■ 112 = 132
{file.OnHand} ■ 877 = 114
```

«where *{file.OnHand}* = 991»

```
      Date (1991, 04, 05) ■ 12 = Mar 24 91
```

Expanded example(s) using this operator

Formula 1

Formula 2

Formula 4

Formula 5

Formula 7

Formula 12

Multiply Operator

Operator (Symbol/Word)

*

Usage

x*y

«Multiply value *x* by value *y*»

Example(s)

```
2883 * 1999 = 5,763,117
{file.Exmpt} * 356.00 = 152,012
```

«where *{file.Exmpt}* = 427»

Expanded example(s) using this operator

Formula 1

Divide Operator

Operator (Symbol/Word)

/

Usage

x/y

«Divide value *x* by value *y*»

Example(s)

```
25 / 5 = 5
1 / 3 = .333333
{file.DaysDue} / 5 = 22
```

«where *{file.DaysDue}* = 114»

NOTE: If the denominator = 0, the report will be halted with a divide by zero warning from Crystal Reports. If you want to avoid this type of problem, you should put a test in. For example:

```
If {file.forecast} = 0 Then
0
Else
{file.sales} / {file.forecast}
```

Expanded example(s) using this operator

Formula 13

Percentage Operator

Operator (Symbol/Word)

%

Usage

x % y

«Calculate value *x* as a percentage of value *y* [(x/y) * 100]»

Example(s)

{file.Balance outstanding} % *{file.Credit Limit}* means the value of *{file.Balance Outstanding}* is what percent of the value of *{file.Credit Limit}?*

```
{file.Balance outstanding} % {file.Credit Limit} = 30.00
```

«where {file.Balance outstanding} = $1500 and {file.Credit Limit} = $5000»

{file.Amount % {file.*Credit Limit}* means the value of *{file.Amount}* is what percentage of the value of *{file.Credit Limit}.*

```
{file.Amount} % {file.Credit Limit} = 32.26
```

«where {file.Amount} = 2257.87 and {file.Credit Limit} = 7000»

NOTE: If the denominator = 0, the report will be halted with a divide by zero warning from Crystal Reports. If you want to avoid this type of problem, you should put a test in. For example:

```
If {file.forecast} = 0 Then
0
Else
{file.sales} / {file.forecast}
```

Expanded example(s) using this operator

Formula 6

Negate Operator

Operator (Symbol/Word)

■( )

Usage

■(x)

«Multiply the value inside the parentheses by ■1»

Example(s)

■(
■1) = 1

«Negative times negative = positive»

■(0
■14) = 14
■ (
■ (15
■18) =
■3

«15 ■ 18 =
■3,
■(
■3) = +3,
■(+3) =
■3»

Expanded example(s) using this operator

Formula 5

Formula 8

To Currency Operator

Operator (Symbol/Word)

$

Usage

$x

«Convert *x* from number to currency»

Example(s)

The following examples all assume the following format (set in the Format Number dialog box): Decimal places = (1.00), Negative sign = (345.00-), Currency symbol = (Float), and Thousands Separator = (1,000.00).

```
$12345678= $12,345,678.00
```

```
$({file.Quantity} * 3) = $42.00
```

«where *{file.Quantity}* = 14»

```
$({file.Miles} * {file.Pledge}) = $363.35
```

«where *{file.Miles}* = 13 and *{file.Pledge}* = 2.15»

NOTE: $ * $ = error. You can not multiply a currency value with a currency value.

Expanded example(s) using this operator

Formula 5

Equal Operator

Operator (Symbol/Word)

=

Usage

x = y

«*x* is equal to *y*»

The equal operator tells Crystal Reports to evaluate an expression (x=y) and return a YES (if *x* is *equal* to *y*) or NO (if *x* is *not equal* to *y*)

Example(s)

```
{file.Quantity} = 3 is
YES
```

«where *{file.Quantity}* has a value of 3»,

```
NO
```

«in all other situations».

```
{file.YTD} = {file.LastYearYTD} is
YES
```

«where the value of the field *{file.YTD}* is identical to the value of the field *{file.LastYearYTD}*»,

```
NO
```

«in all other situations».

```
({file.Sales} ■ {file.COGS}) = 22,554 is
YES
```

«where calculating the expression *{file.Sales}*■*{file.COGS}* produces the value 22,554, i.e. *{file.Sales}* = 109,986}, *{file.COGS}* = 87,332}»,

```
NO
```

«in all other situations».

```
{file.LNAME} = "Johnson" is
YES
```

«where the text string in the *{file.LNAME}* field is "Johnson"»,

```
NO
```

«in all other situations».

Comments

This operator is often used in expressions with the If■Then ■Else operator. For example:

```
If {file.Purchases} = 0 Then
"Your account had no activity this month."
Else
""
```

«which prints the "Your account..." statement if the *{file.Purchases}* field has a zero value, and prints nothing (indicated by the empty text string ""), if the *{file.Purchases}* field has something other than

a zero value.»

Expanded example(s) using this operator

Formula 9

Formula 15

Formula 17

Not Equal Operator

Operator (Symbol/Word)

<>

Usage

x<>y

«*x* is not equal to *y*»

The equal operator tells Crystal Reports to evaluate an expression (x<>y) and return a

YES (if *x* is *not equal* to *y*) or NO (if *x* is *equal* to *y*).

Example(s)

```
{file.Amount} <> 400 is:
YES
```

«where *{file.Amount}* is equal to 200 or *{file.Amount}* is equal to 401, etc.»,

```
NO
```

«where *{file.Amount}* is equal to 400».

```
{file.Day} <> "Thursday" is:
YES
```

«when *{file.Day}* = "Friday"»,

```
NO
```

«when *{file.Day}* = "Thursday"».

```
{file.OnHand} <>0 is:
YES
```

«where the value of *{file.OnHand}* is 10 or 5»,

```
NO
```

«where the value of *{file.OnHand}* is zero».

```
{file.Available} - {file.Used} <>10 is:
YES
```

«where the value of the *{file.Available}* field less the value of the *{file.Used}* field gives a result other than 10»,

```
NO
```

«where it gives a value of 10».

Comments

This operator is often used in expressions with the If-Then-Else operator. For example:

```
If {file.Sex} <> M Then
"FEMALE"
Else
"MALE"
```

«which prints the word "FEMALE" if the value in the *{file.Sex}* field is not equal to M, and which prints

the word "MALE" in all other situations».

Expanded example(s) using this operator

Formula 5

Formula 11

Formula 15

Less Than Operator

Operator (Symbol/Word)

<

Usage

x < y

«*x* is less than *y*»

The *less than* operator tells Crystal Reports to evaluate an expression (x<y) and return a

YES (if *x* is less than *y*) or NO (if *x* is equal to or greater than *y*).

Example(s)

```
{file.Weight} < 200 is:
TRUE
```

«where *{file.Weight}* = 150, or *{file.Weight}* = 199»,

```
FALSE
```

«where {file.Weight} = 200 or {file.Weight} = 400».

```
{file.Cost} < {file.Price} is:
TRUE
```

«where *{file.Cost}*=350, *{file.Price}* = 400»,

```
FALSE
```

«where *{file.Cost}* = 350 and *{file.Price}* = 350, or where *{file.Cost}* = 350 and *{file.Price}* = 325».

Expanded example(s) using this operator

Formula 2

Formula 5

Formula 8

Formula 16

Greater Than Operator

Operator (Symbol/Word)

>

Usage

x > y

«*x* is greater than *y*»

Example(s)

```
{file.Weight} > 200 is:
FALSE
```

«where {file.Weight}= 150, {file.Weight} = 199, or {file.Weight} = 200»,

```
TRUE
```

«where {file.Weight} = 400 or {file.Weight} = 201**»**.

```
{file.Cost} > {file.Price} is:
FALSE
```

«where {file.Cost =350}, {file.Price = 400}, or where {file.Cost} = 350 and {file.Price} = 350»,

```
TRUE
```

«where *{file.Cost}* = 350 and *{file.Price}* = 325».

Expanded example(s) using this operator

Formula 4

Formula 6

Formula 7

Formula 12

Greater Than Or Equal Operator

Operator (Symbol/Word)

>=

Usage

x >= y

«*x* is greater than or equal to *y*»

Example(s)

```
{file.Weight} >= 200 is
FALSE
```

«where *{file.Weight}* = 150 or *{file.Weight}* = 199»,

```
TRUE
```

«where {file.Weight} = 400, {file.Weight} = 200, or {file.Weight} = 201**»**.

```
{file.Cost} >= {file.Price} is:
FALSE
```

«where *{file.Cost}* =350, *{file.Price}* = 400»,

```
TRUE
```

«where *{file.Cost}* = 350 and *{file.Price}* = 325, or where *{file.Cost}* = 350 and *{file.Price}* = 350».

Expanded example(s) using this operator

Formula 4

Formula 6

Less Than Or Equal Operator

Operator (Symbol/Word)

<=

Usage

x <= y

«*x* is less than or equal to *y*»

Example(s)

```
{file.Weight} <= 200 is:
TRUE
```

«where {file.Weight} = 150, {file.Weight} = 200, or {file.Weight} = 199»,

```
FALSE
```

«where *{file.Weight}* = 400».

```
{file.Cost} <= {file.Price} is:
TRUE
```

«where *{file.Cost}* =350, *{file.Price}* = 400, or where *{file.Cost}* = 350 and *{file.Price}* = 350»,

```
FALSE
```

«where *{file.Cost}* = 350 and *{file.Price}* = 325»

Expanded example(s) using this operator

Formula 14

Concatenate Operator

Operator (Symbol/Word)

+

Usage

x + y

«connect string *x* to string *y*»

Concatenate joins multiple text strings to make one contiguous string.

Example(s)

```
"Bread" + " and " + "butter" = "Bread and butter"
```

```
"Your customer number is " + ({file.Customer Number}) + " and your
company contact person is " + ({file.Contact}) +  "." = "Your customer
number is 12345 and your company contact person is Bob."
```

«where *{file.Customer Number}* = 12345 and *{file.Contact}*= Bob».

Comments

You can only use this operator if all the elements you are connecting are text strings. If you want to include a value from a numeric field (for example, an account balance) , you must first convert that value to a text string using the ToText function

```
"Your account balance is " + ToText({file.Balance}) + "."
```

Expanded example(s) using this operator

Formula 2

Formula 5

Formula 13

Formula 15

Subscript Operator (string)

Operator (Symbol/Word)

[ ]

Usage

x[y]

«Extract the *y* element from string *x*» or

x[y to z]

«Extract the *y* to *z* range of elements from string *x».*

NOTE: The subscript ranges are 1 origin: they start at 1 rather than 0.

Subscript is used to extract one or more characters from a text string (an array of characters). For example:

```
{file.Item number} [4]
```

«extracts the 4th element of the item number».

```
{file.Item number} [4 to 5]
```

«extracts the 4th and 5th elements of the item number».

***NOTE: The correct expression for specifying a range of elements in a text string (array) is* x to y**

Example(s)

```
{file.LNAME} [1] = "S"
```

«where *{file.LNAME}* = Smith».

```
{file.Postal} [6] = "V"
```

«where *{file.Postal}* = T5A 9V2 (the space between A and 9 counts as an element)»**.**

```
{file.ItemNumber} [4 to 5] = 40
```

«where *{file.ItemNumber}* is A1/4020/B10».

Comments

Don't confuse Subscript with In String. While Subscript tests a target string for the presence of an element and extracts the element (if found) from the string. In String simply tests the target string for the presence of the element.

Expanded example(s) using this operator

Formula 2

Formula 3

Formula 9

Formula 11

In String Operator

Operator (Symbol/Word)

in

Usage

x in y

«Test for the presence of string *x* in string *y*»

Example(s)

```
"Elm"  in {file.Address} =
TRUE
```

«where *{file.Address}* is "1335 Elm Street"».

```
"elm" in {file.Motto} =
FALSE
```

«where {file.Motto} = "Feel more energy"».

(The "el" ending "feel" and the "m" beginning the word "more" are separated by a space which itself counts as an element.)

```
"el m" in {file.Motto} =
TRUE
```

«where *{file.Motto}* = "Feel more energy"».

(The search string "el m" this time contains the space between the "l" and the "m" which allows for a perfect match.)

```
"bread and butter" in "bread " + "and "   +  "butter" =
TRUE
```

(Crystal Reports first concatenates the string and then tests it for the presence of the string "bread and butter".)

NOTE: "In" can also be used to test for the presence of text in a text range, i.e., "V5B" in "V0A" to "V9Z". Such a range can be created using the Make Range operator.

Make Range Operator

Operator (Symbol/Word)

to

Usage

x to y

«Create the range *x* to *y*»

Example(s)

```
100.00 to 250.00
```

«the range of consecutive numeric values beginning with 100.00 and ending with 250.00, including the end values».

```
Date (1990, 09, 01) to Date (1990, 09, 20)
```

«the range of consecutive dates beginning with September 1, 1990 and ending with September 20, 1990. Both September 1 and September 20 are included in the range».

Comments

You cannot create a formula that has a range as a result. Thus, Make Range is *always* used in conjunction with other operators such as the In Range operator. The combination of Make Range and In Range produces a formula that gives a single TRUE or FALSE value, not a range.

Crystal Reports comes with a number of date ranges such as YearToDate, preset for your convenience.

Expanded example(s) using this operator

Formula 2

Formula 3

Formula 9

Formula 10

In Range Operator

Operator (Symbol/Word)

in

Usage

x in y

«Tests a range of values (*y*) to see if a value (*x*) falls within the range specified».

Example(s)

```
10 in (5 to 15) = TRUE
```

```
Today in Date(1990, 09, 01) to Date(1990, 09, 20) =
TRUE
```

«if Today = September 15, 1990»,

```
FALSE
```

«if Today = September 21, 1990».

```
{file.Qty} in {file.OnHand} to ({file.Backorder} + {file.OnOrder}) =
TRUE
```

«where *{file.Qty}* = 20, *{file.OnHand}* = 10, *{file.Backorder}* = 5, *{file.OnOrder}* = 25 (is 20 in the range that begins with 10 and ends with the sum of 5 and 25)»,

```
FALSE
```

«where *{file.Qty}* = 31, *{file.OnHand}* = 10, *{file.Backorder}* = 5, *{file.OnOrder}* = 25 (is 31 in the range that begins with 10 and ends with the sum of 5 and 25)»

Comments

The combination of Make Range and In Range operators is often used with the If▪Then ▪Else operator. For example,

```
If ({file.Amount} in (100.00 to 250.00)) Then
(.10 * {file.Amount})
Else
0
```

«The above means "See if the value of *{file.Amount}* falls within the range 100.00 to 250.00. If it does, multiply .10 times *{file.Amount}*. If it does not, return zero."**»**

Expanded example(s) using this operator

Formula 9

Formula 10

Make Array Operator

Operator (Symbol/Word)

[,]

Usage

[x,y,z,...n]

«build an array containing the elements *x*, *y*, *z*,...*n*».

Examples

```
[100,200,300,400]
[{file.QtyA}, {file.QtyB}, {file.QtyC}]
[500, ({file.Qty} / 3)]
```

Subscript Operator (array)

Operator (Symbol/Word)

[ ]

Usage

x[y]

«extract the *y* element of an array *x*»

Example(s)

```
[100, 233, 466, 998] [3] = 466
```

In Array Operator

Operator (Symbol/Word)

in

Usage

x in [y]

«is *x* in the array *y*»

Example(s)

```
{file.State} in ["CA", "HI", "AK"]
```

«Is the value of the *{file.State}* field in the array of state abbreviations listed in the brackets?»

```
{file.Color} in ["Red", "White", "Blue"]
```

«Is the value of the *{file.Color}* field in the array of colors listed in the brackets?»

```
DayofWeek({file.Date}) in [2, 4, 6]
```

«Is the value of the *{file.Date}* field, converted to a number that represents the day of the week, in the array of numbers listed in the brackets? (Sunday = 1, Saturday = 7)»

And Operator

Operator (Symbol/Word)

and

Usage

x and y

False *and* False = False

False *and* True = False

True *and* False = False

True *and* True = True

Example(s)

```
If    {file.Credit Limit} = 5000 and
{file.Salesman} = "SP" Then
      {file.Amount}
Else
      0
```

«means that if the credit limit is 5000 and the salesman is SP (*both conditions true*), then return the value in the *{file.Amount}* field, otherwise return zero».

```
A > B and B > C =

TRUE
```

«where A = 10, B = 6, and C = 3 (both conditions true)»,

```
FALSE
```

«where A=10, B=6, and C=7 (only one of the two conditions true)».

```
(A>B) and (A * C ■ D > E) and (E / D <= B) =
TRUE
```

«where A = 7, B = 5, C = 3, D = 2, E = 10 (all three of the conditions are true.)».

Expanded example(s) using this operator

Formula 6

Formula 15

Or Operator

Operator (Symbol/Word)

or

Usage

x or y

«either *x* or *y* or both is true»

False or False = False

False or True = True

True or False = True

True or True = True

Example(s)

```
If    {file.Credit Limit} = 5000 or
{file.Salesman} = "SP"  Then
      {file.Amount}
Else
0
```

«This means that if the credit limit is 5000, or, if the salesman is SP, (either of the conditions are true), then return the value in the *{file.Amount}* field, otherwise return zero».

```
A > B or B > C =

TRUE
```

«where A = 10, B = 6, and C = 3 (*both conditions true*)»,

```
TRUE
```

«where A=10, B=6, and C=7 (either one of the two conditions true)» and

```
FALSE
```

«where A=5, B=6, and C=7 (neither of the two conditions true)».

```
(A>B) or (A * C ■ D > E) or  (E / D<= B) =
TRUE
```

«where A = 5, B = 5, C = 3, D = 2, E = 12 (At least one of the three conditions is true. In this case only (A * C ■ D > E) is true.)»

Expanded example(s) using this operator

Formula 6

Not Operator

Operator (Symbol/Word)

not

Usage

not (x)

«reverses the True or False value of *x*»

not (True) = False

not (False) = True

not (not(False)) = False

not (not(True)) = True

Example(s)

```
not (A>B and B>C)
```

«If *A*=5, *B* = 4, *C* = 3, the expression (*A*>*B* and *B*>*C*) is TRUE. Both conditions tied together by the boolean operator And are TRUE, thus the entire statement has a value of TRUE. The Not operator thus changes the value of the expression to FALSE».

```
not (A>B and B>C)
```

«If *A*=3, *B* = 4, *C* = 3, the expression (*A*>*B* and *B*>*C*) is FALSE. One of the two conditions tied together by the boolean operator And is FALSE, thus the entire statement has a value of FALSE. The Not operator thus changes the value of the expression to TRUE».

```
not ({file.OnHand} ■ {file.Order} > 0) =
TRUE
```

«if {file.OnHand} = 10 and {file.Order} = 11»,

Expanded example(s) using this operator

Formula 10

Assignment Operator

Operator (Symbol/Word)

:=

Usage

x := n

«assigns the value *n* to the variable *x*»

Examples

```
Amount:= 0
```

«initializes (zero's out) the variable named Amount.»

```
Amount:= 100
```

«assigns the value 100 to the variable named Amount.»

```
Amount:= Amount + {detail.QTY}
```

«assigns the result of a calculation to the variable named Amount. The calculation adds the value of the quantity field {detail.QTY} to the current value of the Amount variable. This type of expression is useful in running total situations where each running total consists of the current amount plus an additional value.»

```
Customer:= "Westside Motors"
```

«assigns the string "Westside Motors" to the variable named Customer.»

```
Customer:= TrimRight({file.FNAME}) + {file.LNAME}
```

«trims the trailing blanks from the first name field ({file.FNAME}), concatenates that field to the last name field ({file.LNAME}), and assigns the concatenated value of both fields to the variable named Customer.»

```
Customer:= "Mr. " + {file.LNAME}
```

«concatenates the string "Mr. " with the value of the last name field {file.LNAME}, and assigns the concatenated value to the variable named Customer.

Parentheses Operator

Operator (Symbol/Word)

( )

Usage

(x + y) * z

«perform the calculations inside the parentheses first»

Parentheses are used to control the order in which Crystal Reports calculates a formula.

Example(s)

```
      8 + 6 * 3 ■ 6 / 2 = 23
(8 + 6) * 3 ■ 6 / 2 = 39
{file.Sales} ■ {file.COGS}
■ {file.T&E} * .8 = 11,800
```

«where *{file.Sales}* = 25,000, *{file.COGS}* = 12,000, and *{file.T&E}* = 1500»

```
   {file.Sales} ■ (({file.COGS}
■ {file.T&E}) * .8) = 16,600
```

«values the same as previous example»

```
   ({file.Sales} ■ {file.COGS}
■ {file.T&E}) * .8 = 9200
```

«values the same as first Sales example»

Expanded example(s) using this operator

Most formula examples use this operator. Select the formula of interest from the Formulas in Action index.

See Also

Order of Precedence

If▪Then
▪Else Operator

Operator (Symbol/Word)

if then else

Usage

if x then y else z

«If *x* is true then do *y*. If *x* is not true (else), do *z*.»

Example(s)

```
If    ToNumber({file.Item}) >= 2500 and
ToNumber({file.Item}) < 2600 Then
      "Seasonal"
Else
""
```

«If statement includes an And operator.»

```
If {file.Count} >= 25 Then
{file.Distributor} * {file.Count}
Else
{file.Dealer} * {file.Count}
```

«Quantity ordered determines price list used.»

```
If PageNumber = 1 Then
PrintDate
Else
Date(0,0,0)
```

«prints the print date (from the PrintDate function) on the first page, and prints nothing [as designated by the empty date Date(0,0,0)] on the remaining pages.»

Comments

The *If* part of the expression can include text, numbers, Boolean expressions (Cust#<"10000"), and formulas ({@Formula}), where @Formula is Boolean. The *Then* and the *Else* parts, however, must both be of the same type: (*Then* text, *Else* text; *Then* number, *Else* number). Mixing text and number actions will result in an error message.

Expanded example(s) using this operator

Most formula examples use this operator. Select the formula of interest from the Formulas in Action index.

Statement Separator

Operator (Symbol/Word)

;

Usage

1+1;"abc"

«1+1 and "abc" are two different formula statements in a multiple statement formula. The semicolon between the statements specifies where one statement ends and the next one begins. Without the semicolon the statements would be treated together as an individual statement».

Starts With Operator

Operator (Symbol/Word)

x startswith y

Usage

{fieldname} startswith "abc"

«This operator tests to see if the contents of {*fieldname}* start with a character string that you specify "abc". If the contents of the field do start with "abc", then the formula returns the value True. If the field starts with anything else, the formula returns False.».

Example(s)

```
{company.name} startswith "C"
TRUE
```

«where *{company.name}* = Carls Custom Log Cabins»,

```
{company.name} startswith "Acme"
TRUE
```

«where *{company.name}* = Acme Farming Supply Co.»,

```
FALSE
```

«where *{company.name}* = American Birding Assoc.».

Comments

The Starts With operator is useful for selecting records to include or exclude from your report.

Like pattern Operator

Operator (Symbol/Word)

x like y

Usage

{fieldname} like "c?n*"

«This operator tests to see if the contents of {*fieldname}* matches a pattern that you specify in a character string "c?n*". If the contents of the field do fit the pattern "c?n*", then the formula returns the value True. If the field starts with anything else, the formula returns False.

Use the wildcard symbols **?** and ***** to stand for variable characters. The **?** stands for a single character. The ***** symbol stands for any number of characters.».

Example(s)

```
{customer.firstname} like "D?n"
TRUE
```

«where {customer.firstname} = Dan»

```
{customer.lastname} like "*s?n*"
TRUE
```

«where *{customer.lastname}* = Johnson»,

```
FALSE
```

«where *{customer.lastname}* = Johnston»,

«where *{customer.lastname}* = Smith»

Comments

The Like operator is useful for selecting records to include or exclude from your report.

OPERATORS INDEX (ALPHABETICAL)

Click on a button on the picture below to locate the Operator of your choice, or choose from the list below:

A

Add +

And and

Assignment :=

C

Concatenate +

D

Divide /

E

Equal =

G

Greater than >

Greater than or equal >=

I

If Then

Else If

Then

Else

In range in

In Array in

In String in

L

Less than <

Less than or equal <=

M

Make range to

Make array [,]

Multiply *

N

Negate -()

Not not

Not equal <>

O

Or or

P

Parentheses ()

Percentage %

S

Statement separator ;

Subscript (array) [ ]

Subscript (string) [ ]

Subtract -

T

To Currency$

Error Messages & Formula Compiler Warnings

A) is missing.

Parentheses must be used in pairs; each opening parenthesis must be matched with a closing parenthesis. One of your opening parentheses is not matched by a closing parenthesis. Insert the missing parenthesis and recheck.

A] is missing.

Brackets must be used in pairs; each opening bracket must be matched with a closing bracket. One of your opening brackets is not matched by a closing bracket. Insert the missing bracket and recheck.

A boolean range variable is not allowed.

You have entered a Boolean range variable. Range variables are allowed in all data types other than Boolean. Either change the data type to something other than Boolean, or enter a Boolean item variable to replace the Boolean range variable.

Access denied.

DOS will not allow access to a file specified. Make certain the file is not in use by another program (or another user on a network), and/or make certain you have the right network permissions and try again.

A day number must be between 1 and the number of days in the month.

You have entered a day number that doesn't fit the month. The Formula Checker displays this warning if, for the month of January, for example, you enter a day number of zero (0) or a number 32 or greater. Change the day number to fit the month and recheck.

A field is required here.

You have entered something in your formula other than a field at a position where a field is expected. Correct the problem and recheck.

A formula cannot refer to itself, either directly or indirectly.

You cannot enter a formula that refers to itself. For example, in creating the formula @Profit, you cannot use @Profit as the argument to a function. Remove the reference and recheck.

A function is required here.

The Formula Editor is expecting a function but none was entered. Review your formula and enter the required function or correct the formula if it is in error.

A memo field cannot be used in a formula.

You have picked a memo field for use in a formula. Crystal Reports does not allow the use of memo fields in formulas. Remove the memo field from the formula and try again.

A month number must be between 1 and 12.

You have entered a month number that falls outside the allowable range. Enter a month number between 1 and 12 and recheck.

A string can be at most 254 characters long.

Crystal Reports allows strings in formulas to be up to 254 characters long. You have entered a string that exceeds that limit. Reduce the length of the string (or break it into 2 or more concatenated

strings) and recheck.

A subscript must be between 1 and the length of the string.

You have entered a subscript number that specifies a character that doesn't exist. If you enter a subscript that references the 6th or the 8th character in a five character string, for example, you will get this warning. Change the subscript to a value that exists and recheck.

A subscript must be between 1 and the size of the array.

You have entered a subscript that specifies an array item that doesn't exist. If you enter a subscript that references the 6th or 8th item in a five item array, for example, you will get this warning. Change the subscript to a value that exists and recheck.

All network licenses are in use. You will be able to run Crystal Reports when a user leaves the program. To increase the number of licensed copies of Crystal Reports on your network, contact Crystal Services at (604) 681▪3435 and ask about Network LanPaks.

Your current license specifies the maximum number of users that can use Crystal Reports on a network at any given time. The maximum number of users are currently using the program. You can increase the number of users allowed on the system at a given time by purchasing additional Network LanPaks through Crystal Services.

A subtotal condition is not allowed here.

You have entered a subtotal condition for a subtotal that uses something other than a date or Boolean field as the sort and group by field. Your subtotal does not require a condition. Delete the condition and continue.

A subtotal condition must be a string.

You have entered a subtotal condition that is not in string format. Make certain when you enter the condition in the formula that it is surrounded by single or double quotation marks.

A variable cannot be redeclared with a different type.

You have declared a variable with the same name but a different data type than a variable already declared. This is not allowed. Either change the name of the variable or change the data type so it conforms with the original data type.

A variable is required here.

You have used the assignment operator (=:) in a formula without preceding it with a variable. The program expects to see a variable immediately before (to the left of) the assignment operator. Enter a variable and try again.

A variable name is expected here.

You have declared a variable data type without declaring a variable name. You must enter a variable name to complete the declaration. Enter the variable name and continue.

Dates must be between year 1 and year 9999.

You have entered a date that falls outside the allowable range. Enter a date that falls within the range of years 1 to 9999 (including the end values), and then recheck.

Cannot allocate memory

This message typically indicates that there is not enough memory available. Close any reports that are not needed, and exit any programs that are not essential. Then try again.

Cannot reallocate memory

This message typically indicates that there is not enough memory available. Close any reports that are not needed, and exit any programs that are not essential. Then try again.

Disk full

You have attempted to save a report to a disk that is full. Either save to a different disk, or delete unnecessary files from the current disk and try again.

Division by zero.

You have entered a formula that attempts a division by zero. Crystal Reports does not allow such a division. Edit the formula so it does not attempt to divide by zero, and then recheck.

If you want to avoid this type of problem, you can use a test such as this:

```
If {file.forecast} = 0 Then
0
Else
{file.sales} / {file.forecast}
```

Error in formula code. Please contact Crystal Services.

There is something unusual about the formula that was not foreseen. Please save the formula text that produced this warning and contact the company.

Error in parse tree. Please contact Crystal Services.

In parsing your formula, the program encountered a situation that the parse tree could not process. Please save the formula text that produced this warning and contact the company.

Field still in use.

The field you are requesting is currently in use. Try again once the field becomes available.

File name already in use. Please close the window for xxx before saving under this name.

You have tried to save a file under the name of a file already in use in an open report. Close that report first, and then try again.

File not found.

The file name you specified cannot be found. Either the filename or the path is incorrect. Enter the correct filename/path and try again. In some instances the file WBTRVDEF.DLL is missing from your CRW directory. This file is required for reading Data Dictionary files along with WBTRCALL.DLL.

File permission error

You have requested a file for which you don't have permission. You must gain the necessary permission before you can activate the file.

Incorrect Borland Custom Control DLL (BWCC.DLL) installed. Version m.n or higher required.

Crystal Reports is finding and using a version of BWCC.DLL that is too old for proper program operation. Here's how this can happen:

- BWCC.DLL is installed in the CRW directory (the same directory in which CRW.EXE or CRW32.EXE resides) during program installation.
- The CRW directory is added to the end of the path statement in AUTOEXEC.BAT during installation (if

you allowed the installation program to update the path statement).

- If an older version of BWCC.DLL has been installed in the Windows directory, the Windows System directory, or a directory that appears earlier in the path than the CRW directory (the result of an earlier installation), Crystal Reports picks up that version, not the newer version in the CRW directory.

To correct this problem

The correct version of BWCC.DLL is shipped with Crystal Reports. To correct the problem, delete older versions of BWCC.DLL that reside in directories earlier in the path than CRW.

If this doesn't solve the problem, move the latest version of BWCC.DLL from the CRW directory to the Window's directory.

Internal Error: PrintDlg fail: 4100

There is no printer driver installed in the Windows Control Panel. When Crystal Reports opens a report, it looks for the printer that was saved with the report. If that printer cannot be found, it looks for the default printer. If there is no default printer set, the error message results.

Insufficient memory available

There is not enough memory available to do what you want the program to do. Free up memory and try again.

Invalid DOS version.

You are using a version of DOS earlier than Version 3.0. Install DOS Version 3.0 or higher and try again.

Invalid file handle.

You have specified a file handle that does not exist. Enter the correct file handle and continue.

No default printer selected. You may use the Control Panel to select a default printer.

Please use the Control Panel to select a printer and start Crystal Reports again.

You cannot begin using Crystal Reports unless you have a default printer selected. Trying to start the program without a default printer results in this error message.

***To select a default printer*:**

Select the Printers icon in the Windows Control Panel; the Printers dialog box appears with all installed printers listed in the Installed Printers box.

If you haven't yet installed the printer, install it first, and then Double Click its listing.

NOTE: A printer must first be given the status Active before it can be selected as the default printer.

NOTE: For additional information in installing printers and default printers, please refer to the documentation that came with Microsoft Windows.

Non unique table reference: tablename

This is usually caused if a table name contains an underscore or is more than 15 characters long, or starts with a number.

Not enough arguments have been given to this function.

The function requires more arguments than you have entered. Enter the missing argument(s) and recheck.

Not enough memory

There is not enough memory available to process the command. Close any reports that are not needed, and exit any programs that are not essential. Then try again.

Numeric overflow.

An intermediate result or the final result cannot be represented because it is too big. Restructure or subdivide the formula to create smaller results, and then recheck.

Physical database not found.

The program is unable to locate either a DLL or the database. Check to make certain that the directories that hold these files are listed in the path statement.

Please cancel printing before closing

Your print engine call is attempting to close a job while it is still in progress. Make certain that you cancel the printing before you close the print job.

Printer not available.

Crystal Reports is having difficulty connecting with the selected printer. Reselect the printer through the Windows Control Panel and try again.

Report file already exists.
Overwrite *sample.rpt*?

You are attempting to save a report under the same name as an existing report. This will overwrite the existing report and make it no longer available. Select Yes to overwrite the report, No to stop the saving process to give you a chance to select a different name.

Report has changed.
Save changes to *sample.rpt* before closing?

You are attempting to close a report window without first saving it, even though you have made changes to the report since you opened it. The changes will be lost unless you save the report before closing. Select Yes to save the changes, No to close the report without saving the changes.

Sorry, this feature is not yet implemented. Try again later.

You have attempted to use a Crystal Reports feature which has not been implemented in the current release. Wait till an upgrade that implements the feature and try again.

The formula cannot be evaluated at the time specified.

This field cannot be used because it must be evaluated later.

This formula cannot be used because it must be evaluated later.

This function cannot be used because it must be evaluated later.

You are trying to force a field, formula, or function to be evaluated earlier than is possible. Evaluation time functions can only force a later evaluation time, never an earlier one. Change the formula to accommodate the required evaluation time.

The formula is too complex. Try simplifying it.

The formula could not be evaluated because it exceeds the limit of 50 pending operations. Pending operations are operations that are on hold due to order of precedence rules; they will be performed once the operations with higher level precedence are finished.

Sometimes it is possible to rearrange the formula and calculate the same value without requiring as many pending operations. As a very simplified example, in the formula 2+3*4, the addition cannot be performed until the multiplication has been done. The addition thus becomes pending, on hold until the multiplication is complete. If the formula is written as 3*4+2 instead, the operations can be performed left to right with the same result, thus eliminating the pending operation.

Correct the formula and recheck.

The matching } for this field name is missing.

Field names must be enclosed in braces { }. You have entered one of the required braces but not the other. Insert the missing brace and recheck.

The matching ' for this string is missing.

A string that begins with a ' must end with a ' before the end of the line. You have used the ' in one of those positions but not the other. Insert the missing punctuation and recheck.

The matching " for this string is missing.

A string that begins with a " must end with a " before the end of the line. You have used the ' in one of those positions but not the other. Insert the missing punctuation and recheck.

The number of copies of the string is too large or not an integer.

Using the ReplicateString function, you have requested too many copies or you are requesting a non integer number of copies. Lower the number of copies requested or specify an integer number of copies and try again.

The number of days is too large or not an integer.

When adding days to dates, or subtracting days from dates, you can use only an integer number of days (a whole number); you cannot add or subtract non integer numbers of days (1/2 days, 3.6 days, etc.). Additionally, once you add or subtract days from a date, the resulting date must fall within the allowable (year) date range, 0000 9999. If you enter a non integer number of days or if your result falls outside the allowable range, the Formula Editor displays this warning. Correct the problem and recheck.

The number of decimal places is too large or not an integer.

The second argument to the Round(x, # places) or ToText(x, # places) functions must be a small integer (whole number). You have entered a number as the second argument (# places) that specifies too many decimal places or that is not an integer. Change the number to a small integer and recheck.

The record selection formula cannot include 'PageNumber', 'RecordNumber', 'GroupNumber', 'Previous', or 'Next'.

You cannot include PageNumber, RecordNumber, GroupNumber, Previous, or Next fields in record selection formulas. Eliminate the field(s) and recheck.

The record selection formula cannot include a summary field.

You have included a summary field in a record selection formula. Crystal Reports does not allow this. Remove the summary field and recheck.

The remaining text does not appear to be part of the formula.

You have provided a formula operand (the item on which a formula operation is to be performed)

where none is expected. Often this means that you have forgotten an operator, or an earlier part of a function, or some required syntax item. Correct the error and then recheck.

The result of a formula cannot be a range.

You have created a formula that results in a range. A formula must result in a single value. Correct the formula and recheck.

The result of a formula cannot be an array.

You have created a formula that results in an array. A formula must result in a single value. Correct the formula and recheck.

The result of the selection formula must be a boolean.

You have created a selection formula that returns something other than a Boolean value. Reconstruct the formula using comparison operators (=,, etc.) and recheck.

The string is non-numeric.

The argument to the ToNumber function must be a number stored as a string (for example, a customer number, an I.D. number, etc.). The string may be preceded by a minus sign and may contain leading and trailing blanks. You have used an argument that is non-numeric and therefore cannot be converted to a number. Change the argument to numeric and recheck.

The variable could not be created.

The variable you declared couldn't be created. Check the spelling and syntax of your declaration statement and try again.

The word 'else' is missing.

In an If-Then-Else expression, you have left out (or misplaced) the 'else' component and the formula will not function. Insert (or reposition) the 'else' component and recheck.

The word 'then' is missing.

In an If-Then-Else expression, you have left out (or misplaced) the 'then' component and the formula will not function. Insert (or reposition) the 'then' component and recheck.

There are too many characters in this field name

A field name may have at most 254 characters. You have entered a field name that exceeds that number. Enter a field name that has an allowable number of characters and try again.

There are too many characters in this string.

Crystal Reports allows strings in formulas to be up to 200 characters long. You have entered a string that exceeds that limit. Reduce the length of the string (or break it into 2 or more concatenated strings) and recheck.

There are too many digits in this number.

Crystal Reports allows numbers in formulas to have up to 25 digits before the decimal point. You have entered a number that exceeds that limit. Reduce the size of the number (or break it into 2 or more smaller numbers) and recheck.

There are too many letters and digits in this name.

A variable name can have at most 254 characters. You have entered a name that exceeds that

number. Shorten the name to conform to the limit and continue.

There is an error in this formula. Please edit it for more details.

You have tried to accept a formula (via the Accept button in the Formula Editor) that contains an uncorrected error. Correct the error that was indicated and try again.

There must be a subtotal section that matches this field.

You have entered a subtotal in a formula without there being a corresponding subtotal in the report itself. Any subtotal you enter in a formula must duplicate a subtotal already in your report. Add the required subtotal to the report and then re-enter the formula, or delete the formula, and then recheck.

The special variable field could not be created.

This message typically indicates that there is not enough memory available. Close any reports that are not needed, and exit any programs that are not essential. Then try again.

The summary field could not be created.

This message typically indicates that there is not enough memory available. Close any reports that are not needed, and exit any programs that are not essential. Then try again.

This field cannot be summarized.

You have entered a summary field that does not already exist in your report. Any summary field you enter in a formula must duplicate a summary field already in your report. Either enter the summary field in your report first and then re-enter it in the formula, or don't enter the summary field in the formula at all.

This field cannot be used as a subtotal condition field.

The field you are entering as a condition field causes the subtotal in the formula not to match any subtotal in the report. Any subtotal you enter in a formula must duplicate a subtotal already in your report. Either enter the subtotal in your report first and then re-enter it in the formula, or don't enter the subtotal in the formula at all.

This field has no previous or next value.

You have used a field for which there is no "previous" value as the argument for the Previous or PreviousIsNull function, or you have used a field for which there is no "next" value as the argument for the Next or NextIsNull function. If you want to use either of those functions, replace the argument with a field that contains the appropriate values.

This field must be in the same section as the current formula.

Since the field was put into the formula as an operand, it has been moved to a section where it is no longer a valid operand.

This field name is not known.

You have entered a field name that does not appear in any of the active databases. Correct the spelling of the field name and/or its alias, and then recheck. Or, if you want to enter a field name from a database that is not currently active, activate the database first and then re-enter the field name.

This group section cannot be printed because its condition field is non-existent or invalid.

Your report contains a group section that is based on a condition field that is either no longer in the report or changed so it is invalid for the group section. Review your grouping criteria to identify and correct the source of the problem.

This array must be subscripted. For example: Array [i].

You have entered an array without enclosing it in brackets. Enclose the array in brackets and recheck.

This subtotal condition is not known.

You have entered a subtotal condition that does not appear anywhere in your report. Any subtotal you enter in a formula must duplicate a subtotal already in your report. Change the condition and recheck.

Too many arguments have been given to this function.

You have entered an array as the argument to a non-array function. This kind of problem can occur, for example, if you forget to use brackets (the required syntax items for an array) to enclose an array. The Formula Checker sees the array values as arguments to a non-array function and displays the error message.

Too many items have been given for this array.

Crystal Reports allows up to 50 values in an array. You have exceeded this limit. Reduce the number of values in the array and recheck.

Too many open files.

You have too many open files (databases, reports) given the number of files you specified in the CONFIG.SYS FILES = statement. To prevent this error from recurring, either use fewer files or increase the number of files specified in the FILES = statement.

Missing or incorrect operand warnings

The following warnings appear when the Formula Checker expects to find a specific kind of operand (the item on which a formula operation is to be performed), and finds something different. For example, the formula **5>a** is comparing a number to text (the old comparing apples to oranges analogy). When the Formula Checker sees that the number five is being compared to something, it expects that something to be another number. If anything other than a number appears, it displays the warning: *A number is required here.*

A Boolean array is required here.

A Boolean is required here.

A currency amount is required here.

A currency array is required here.

A currency range is required here.

A date array is required here.

A date is required here.

A date range is required here.

A number array is required here.

A number array or currency array is required here.

A number, currency amount, boolean value, or string is expected here.

A number, currency amount, Boolean, date, or string is required here.

A number, currency amount, date, or string is required here.

A number, currency amount, or date is required here.

A number field or currency amount field is required here.

A number is required here.

A number or currency amount is required here.

A number range is required here.

A string array is required here.

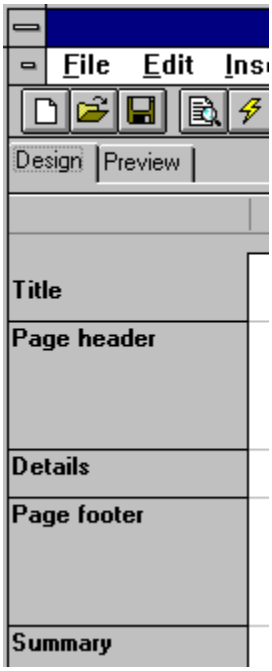
A string is required here.

A string or an array of values is required here.

An array of values is required here.

The Design Window

Once you select your database, Crystal Reports displays the Design Window. You use this window to insert and format data and to watch your report take shape.



When you open a new report, Crystal Reports automatically creates five sections in the Design Window:

- a **Title section**
- This section is generally used for the report title, and other information you want to appear at the top of the first page of your report.
- a **Page header section**
- this section is generally used for field headings, range of values included, and other information that you want to appear at the top of each page.
- a **Details section**
- this section is the body of the report. The bulk of your report data will generally appear in this section.
- a **Page footer section**
- this section usually contains the page number and any other information that you want to appear on the bottom of each page.
- a **Summary section**
- This section is generally used for including a written summary printed only on the last page of your report.

Each section is separated by a section marker which displays the section name.

- You build your report by inserting data fields, formulas, and other report elements (record counts, record numbers, etc.) in the Details section of the Designer. You use the Insert menu, in most cases, to select or create the elements you want to insert on the report. The Design Window uses rectangular field boxes to indicate the size, position, and data type of the fields and formulas you have inserted.
- You add subtotals (and other group values) by selecting a field to summarize and then telling Crystal Reports the conditions that are to generate a new summary(change of customer number, change of sales rep, etc.). Crystal Reports creates group sections as needed and places the group value in the section. Again, Crystal Reports uses rectangular field boxes to represent the group values.
- You insert grand totals in the Grand Total section. This section appears when you select the field to total and then select Insert|Grand Total or when you opt to place a summary field in the Grand Total section. In both cases, Crystal Reports uses a rectangular field box, this time to identify the field in the Grand Total section of the Design Window.
- You can add freeform text anywhere on the report by positioning the insertion point in the section in

which you want the text to appear, typing in the text, and then using the Tab key to move it into position. You can also type freeform text as a text field using the Insert|Text Field command; this allows you to avail yourself of additional formatting options (alignment, hide options, etc.).

Creating summary reports

Summary reports are reports that present only summarized values, leaving out the details used to arrive at those values. For example, in a report summarizing the sales generated by each sales rep, the total amount of sales for each rep would appear but not the individual orders making up the total.

Crystal Reports makes it easy to create summary reports. To do so you simply create a report, enter totals, subtotals, and summary values to fit your needs, and then turn off (suppress) the details using the Format|Section command. Here is the process.

1. Create the report including all the fields, formulas, subtotals, grand totals, and summaries that you want. (For help in this area, refer to the topics in the Creating a report index.)
2. Select Format|Section. The Format Section dialog box appears.
3. Select Details (to format the Details section of your report.) A second Format Section dialog box appears.
4. Select Hide Section and click OK. Crystal Reports hides the Details section when it prints your report.

NOTE: When you hide the Details section you hide everything in the Details section. You may have certain fields, formulas or text in that section that you want to appear on your final report, however. Re-enter those items in the subtotal section where they will print along with the summarized values.

NOTE: You can also create a summary report using File|New|Summary Expert.

Understanding the grid

The Design Window looks very freeform. It looks like you can place anything anywhere and hope for good results. But that is not the case. By default, the Design Window contains an invisible grid which directly affects the placement of data fields and text fields.

Think of the grid as a series of row and column coordinates. When the grid is active, Crystal Reports allows you to place fields and text only at these coordinates, not between them. In this way it makes it very easy for you to place and space data on your report, and to align text and fields as needed. If you attempt to place a report element between grid coordinates, Crystal Reports "snaps" the element to grid, that is, it moves the element automatically to the nearest coordinate.

When you place a data field in the Design Window, Crystal Reports "snaps" it to the nearest coordinate. Use the arrow keys or the mouse to move the fields, once they have been placed.

- When you use the arrow keys to move the field, each time you press the arrow the field moves **one grid position**.
- When you use the mouse to drag the field to a new location, Crystal Reports "snaps" the field to the nearest coordinate as the field moves.
- You use the Tab key to move the text; each time you press the Tab key the text moves **in quarter inch increments.**.

Using the Grid

In previous versions of Crystal Reports, a fixed size, invisible grid was present to help align your data. Now, Crystal Reports allows you to change the size of the grid, make the grid visible or invisible, and choose whether or not fields snap to the grid. You can also call up individual grid options when you Click the right mouse button when your cursor is positioned in the gray area on the Format bar or the Ruler. The following options are available on the Layout tab under File|Options:

Snapping to the Grid

When Snap to Grid is on, a grid is placed on your report with equally spaced horizontal markers. When you place a field on the report, the field will snap (attach) to the nearest grid marker. This allows you to align fields vertically with precision. You can also use Snap to Grid to space fields across a page equally or to size fields equally.

Show Grid in Design Window/Preview Window

You have the option to show the grid in the Design Window, Preview Window, or both. for example, you may want to show the grid in the Design Window for aligning fields, but have a clear view of your report in the Preview Window. In that case, you would turn Show Grid in Design Window on and turn Show Grid in Preview Window off.

NOTE: Although the grid will be invisible in the Preview Window, the fields in your report will still Snap to the Grid in both the Design Window and the Preview Window as long as you have Snap to Grid turned on.

To set Show Grid in Design Window and Show Grid in Preview Window, select the Layout tab under File| Options. You can also set these options by Clicking the right mouse button at the top of either window near the ruler. A menu will pop up displaying the show grid options. If the option has a checkmark next to it, it is turned on. If there is no checkmark, the option is inactive.

Grid Size

To control the vertical spacing of fields, use the Grid Size option. When you make a change to the grid size, the grid markers on the screen change to match. Space increments are measured in either inches or centimeters, depending on your Windows International Measurement setting. (Regional settings for Windows 95)

Selecting an alias

For a variety of reasons, database names get changed. Many times this is not a problem, but if you have created a report using fields in formulas, changing the name of the database after you have created the formulas could be a real problem. Crystal Reports would look for formula fields under the old database name, and if the database had its name changed, the program wouldn't find the fields and thus couldn't use the formulas.

Crystal Reports avoids this kind of problem by using aliases. Aliases are pointers, programming devices that tell Crystal Reports where it should currently look for a database field. Now, if you change the name or location of the database (via the Database|Set Location command), you simply reset the pointer (tell the alias about the change). The name of the alias doesn't change, so your formulas are not affected. Crystal Reports looks to the alias for a database reference as always, goes to the database referenced, and runs the formula without a problem.

NOTE: Crystal Reports comes preset to use the default alias (the database name without the extension). If you want to use something different than the default, you will need to toggle the Use Default Alias switch off in the Options dialog box. This will cause the Alias Name dialog box to appear whenever you activate a new database.

Adding, copying, deleting, and editing text

To add text

To add text, set the insertion point where you want the new text to begin, and type in your addition, or use a text field (Insert|Text Field command).

To copy text

- Select the text you want to copy by dragging the I
- beam cursor over the text.
- Select Edit|Copy. Crystal Reports moves a copy of the selected text to the Clipboard without disturbing the highlighted text.
- Set the insertion point where you want to insert the copied text and select Edit|Paste. Crystal Reports copies the text at the selected point.

To delete text

- set the insertion point where you want to begin deleting, and press the Delete key enough times to delete the unneeded text, or
- select the text you want to delete by dragging the I
- beam cursor over it, then:
- press the Delete key on your keyboard to delete it permanently, or
- select Edit|Cut (or press Shift
- Delete) to cut the text to the clipboard for later use, or
- set the insertion point at the beginning of a line of text you want to delete and press Shift
- End to select the entire line. Then:
- select Edit|Clear (or press Delete) to delete it permanently,
- or select Edit|Cut (or press Shift
- Delete) to cut the text to the clipboard for later use.

To edit text

Delete, edit, and/or add text as needed following the techniques described above.

Moving text

You can move text in Crystal Reports in two different ways:

- By pushing or pulling it to a new position using the Tab key.
- By Cutting it to the Clipboard and then Pasting it in a new position.

Using the Tab key

Set the insertion point immediately in front of the text you want to move.

- Press the Tab key to move the text to the right. All text to the right of the insertion point moves each time you press the Tab key.
- Press the Backspace key to move the text to the left. All text to the right of the insertion point moves each time you press the Backspace key.

NOTE: If you want to move several pieces of text on a given line into position (i.e., aligning titles with data fields), begin at the left. Move the leftmost text into position, reset the insertion point to the left of the second text element and move it into position, reset the insertion point to the left of the third text element, etc.

Using Cut and Paste

1. Select the text you want to move.
2. Select Edit|Cut. Crystal Reports moves the text to the Clipboard.
3. Set the insertion point to the new text position.
4. Select Edit|Paste to paste the text at the new insertion point.

Moving fields

You move a database field by dragging it to a new position with the mouse, or by selecting it and using the Arrow keys.

With a mouse

1. Select the field you want to move.
   - To select a single field, Click the field.
   - To select multiple fields, hold the Shift key down while you Click the fields. Handles appear on the selected field(s).
2. With the cursor on the field and the left mouse button depressed, move the field to its new position.
3. Release the mouse button when the field is in place.

With the Arrow keys

1. Select the field you want to move.
   - To select a single field, Click the field.
   - To select multiple fields, hold the Shift key down while you Click the fields. Handles appear on the selected field(s).
2. Use the Arrow keys to move the field to its new position. The field moves one grid position each time you press the Arrow key.

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

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

- ***grand totals cannot be moved outside the Grand Total section, and***
- ***a subtotal or summary can be moved only within its initial section or to the header portion of its initial section.***

See Also

Understanding the invisible grid

Printing your report

When you want to print your report, see what your report will look like when printed, or export the report to a disk file, you can use one of Crystal Reports' several printing options.

Click on a topic below for more information:

File|Print Preview (to review your work and fine tune your report using real data)

File|Print|Preview Sample(to review a draft of your report using a subset of data)

File|Print|Printer(for hard copy output)

File|Print|Export(for exporting to disk and changing the output format if necessary)

File|Print|Mail(for sending via E■mail and changing the output format if necessary)

File|Print|Crystal Reports Server(for connecting to the Crystal Reports Server)

When you are creating a report, you will find yourself printing to the Preview Window often, in order to check placement and formatting of the various report elements. You can print a draft report using only a subset of the data you'll use in your final report, and you can even fine tune your report in the Preview Window using real data. Then, when you want to print a final or interim copy of the report for hands■on review, you can print to the printer for hard copy output.

If you want to use your report data in another application (in a spreadsheet or word processor for example), you can export the report to a disk file in any of a variety of popular file formats. Once in a disk file, you can import the data into your other application following the importing procedures established by the receiving application. You can also export your report by E■mail using another output format if you wish.

Crystal Reports data types

The data type of a field, (text, number, dollar, date, or Boolean) determines the type of information that can be stored in that field and which will print in the report.

Fields of each data type display on your screen like this:

```
XXXXXXX
```

Text: for example, a company name, account description or customer name.

Dollar Amounts

```
$5,555,555.55
```

Dollar amount (Paradox/Btrieve files only): for example $500.00 or $50,000,000.00.

```
5,555,555.55
```

Number: for example 120 or 5555.

Dates

```
YYYY■MM
■DD
```

Date: for example Oct. 10 90.

Boolean

```
YES/NO
```

Boolean (YES/NO) data fields: for example, the result of a formula which compares a customer's credit limit to see if it is greater than $5000 and prints YES if the credit limit is more than that amount; NO if it is less than that amount.

BLOBS

Binary Large Objects, for example, bitmap images of items in a database. Bitmap fields appear as the actual bitmaps on your report. You cannot sort your report or select records based on BLOB fields.

How data types are sorted

A field's data type determines the method in which the data from the field is sorted.

Text

Text fields are sorted in the following manner:

- One character values are sorted so that blanks have the lowest value, then punctuation, then numbers, then uppercase letters, and finally lowercase letters.
- Then two character values are sorted, then three, etc., using the same rules. As a result:
- "BOB" comes before "bob",
- "123" comes before "124",
- "" (blank) comes before "a", and
- "aa" comes before "aaa"

Dollars

Dollar fields are sorted in numeric order.

Number

Number values (120, or 5555) are sorted in numeric order.

Dates

Date fields are sorted in chronological order.

Yes/No comparisons (Boolean)

Comparison fields are sorted so that false values come first, then true.

BLOBS

You can not sort your report based on Blob fields.

Databases that work with Crystal Reports

Crystal Reports can build reports using the standard data files generated by dBASE (version III and later), dBASE for Windows, Paradox(up to and including Version 4.5), Paradox for Windows, FoxPro, Clipper, and Btrieve (through the use of .ddf dictionaries). Using the ODBC drivers that ship with Crystal Reports, you can also build reports using Access (.mdb), Excel(.xls), and ASCII text(.txt) files. The Professional Edition of Crystal Reports includes drivers for many popular SQL systems as well.

Since many programs can export data in dBASE format (.dbf files), you will find that you can use Crystal Reports to create reports for virtually any database.

Btrieve files

NOTE: In the 32 versions of Crystal Reports 4.5, Btrieve files are only supported in Windows NT.

To activate a Btrieve file, you can select any .ddf file. Crystal Reports draws in all of the relevant files in the active directory so no linking is necessary.

NOTE: .ddf files are data dictionary files created by Novell's Xtrieve utility. You will need to create .ddf files using this utility before you can use Btrieve files with Crystal Reports.

.db files

NOTE: Paradox databases are not supported in the 32 bit versions of Crystal Reports.

To activate a Paradox .db file, select the file. If you select additional files using the Database|Add Database to Report command, the files will need to be linked by the Visual Linking Expert feature.

Crystal Reports works with the following indexes:

- **px** (Paradox indexes)
- **?, .y??** (Paradox secondary indexes)

.dbf files

To activate a dBASE, Clipper, or FoxPro .dbf file, select the file. If you select additional files using the Database|Add Database to Report command, the files will be linked by the Visual Linking Expert feature.

Crystal Reports works with the following indexes:

- .ndx
- .mdx
- .ntx Clipper
- .ctx FoxPro

ACT files

NOTE: ACT files are not supported in the 32 bit versions of Crystal Reports 4.5

To activate ACT files you must specify *.act in the file name edit box when the Choose Database File dialog box appears. After specifying *.act, press the Enter key. (On some systems this is called CR, or Carriage return) The file crw.act appears in the file name scroll box. Select this file, and Click OK. The Choose File for ACT dialog box appears. Use the file name scroll box to select the ACT file you wish to use and Click the OK button.

.mdb files

If you are upgrading from the version of Crystal Reports that came with Visual Basic, you can select any .mdb file directly as a standard database file. If you are not upgrading from the Visual Basic version, you will have to access .mdb files via ODBC as SQL files. In order to do this, you will have to register the files using the ODBC Admin facility that was installed when you installed the program. In either case, to

activate more than one of the tables in the file, use the Visual Linking Expert feature.

.xls files.

You access .xls files via ODBC as SQL files. In order to do this, you will have to register the files using the ODBC Admin facility that was installed when you installed the program. If you want to add another database, you do so via the Add Database to Report command. The files will need to be linked by the Visual Linking Expert feature.

.txt files

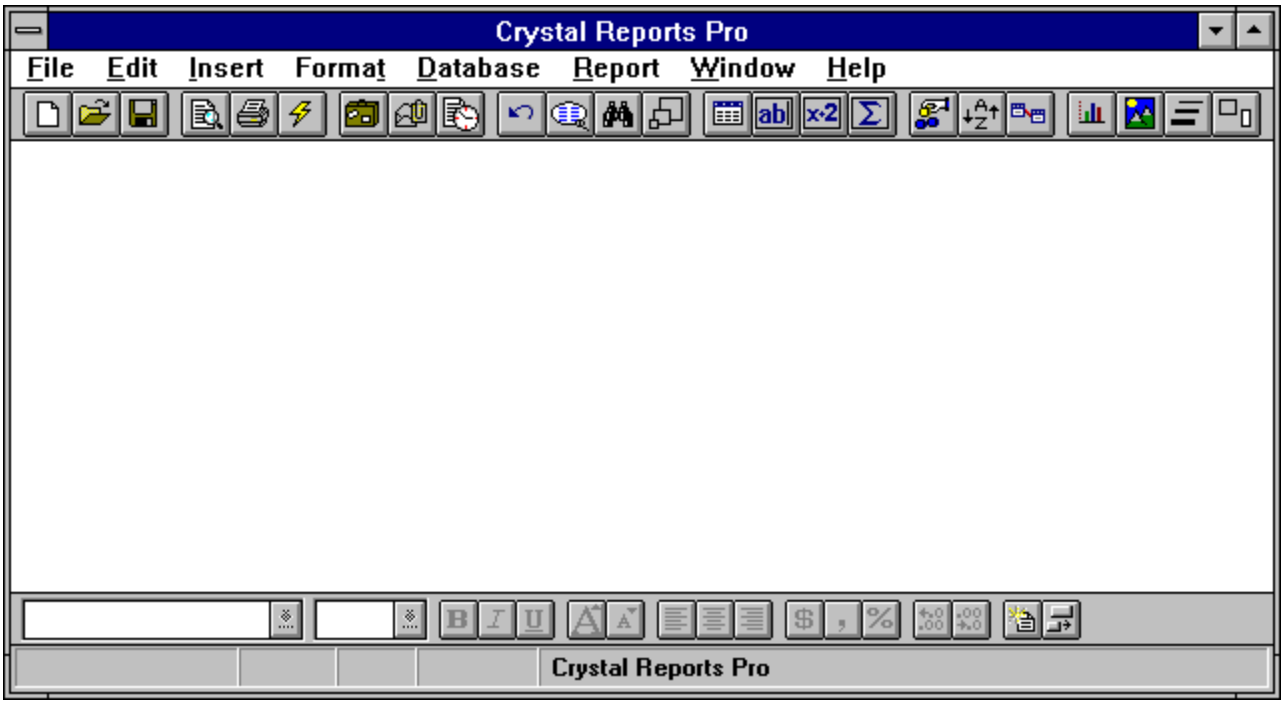
You access ASCII .txt files via ODBC as SQL files. In order to do this, you will have to register the files and define the fields using the ODBC Admin facility that was installed when you installed the program. box. If you want to add another database, you do so via the Add Database to Report command. The files will need to be linked by the Visual Linking Expert feature.

SQL files

See Logging Onto a SQL server and the SQL Features Index for information on using SQL files in your report.

The Crystal Reports Design Window

The Crystal Reports Design Window is clean and easy to understand:



- the Minimize and Maximize buttons appear in the upper right hand corner of the window,
- the Title Bar appears at the top of the window,
- the Menu Bar appears just below the Title Bar, and
- the Button Bar appears just below the Menu Bar.
- the Format Bar and Status Bar appear at the bottom of the window.

The Menu Bar

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

See Also

File Menu Commands

Edit Menu Commands

Insert Menu Commands

Format Menu Commands

Database Menu Commands

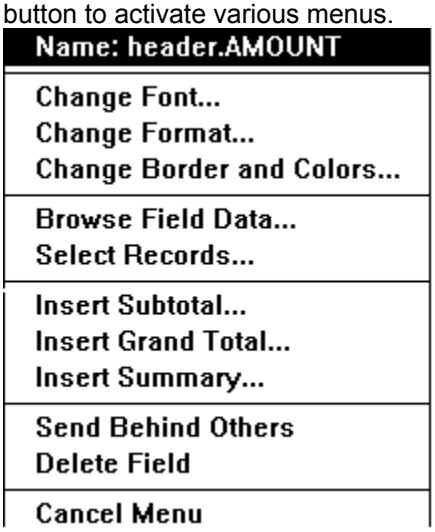
Report Menu Commands

Window Menu Commands

Help Menu Commands

Right mouse button capabilities

When you are working in the Preview Window and Design Window, you can speed up your work considerably using Crystal Reports' right mouse button capabilities. When the cursor is positioned on a report element (a field, a group field, a formula, etc.) and you Click the right mouse button, Crystal Reports displays a pop-up menu right next to the element. To see some examples of pop-up menus, move the cursor around the Design Window and Preview Window and Click the right mouse button to activate various menus.



- The **menu heading** is either the name of the object selected (line, box, etc.), the text if the selection is a text field, the field name if the selection is a field or formula, or a group identifier if the selection is a group.
- The option **Cancel Menu** appears on all menus and simply closes the pop-up menu and returns you to the report.

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

The pop-up menus are valuable because:

- they display the name and source (alias) of the element at the top of the menu so you can identify the elements on your report with a single mouse click,
- they appear right next to the selected element making them quicker and easier to access than Crystal Reports' main menus,
- they contain only the commands you need; you don't need to search for commands on a more comprehensive menu,
- they make it easier to learn Crystal Reports because:
- they eliminate the need to remember where to find a command,
- because you're dealing with only a compact list of commands, they make it easier to pick the right one, and
- they *spotlight* the things you can do with an element making it an easier system to use when you are under pressure or distracted.

NOTE: If you have swapped left/right mouse buttons via the Control Panel, the left mouse button will activate the pop-up menus.

NOTE: You can call up a Grid menu by clicking the right mouse button while in the gray area above the Design Window. You can call up a Section/Line menu by Clicking the right mouse button while in the gray area to the left of the Design Window.

Order of precedence

When entering formulas that contain different kinds of operators, it is important to consider order of precedence, the order in which Crystal Reports performs the operations in your formula.

You learned simple order of precedence in high school math: when performing calculations, do multiplication and division first, then addition and subtraction. Thus:

```
5 + 10 x 3 = 35
```

The calculation 10 x 3 is performed first to get 30. 30 is then added to 5 to arrive at the final answer.

Now if your intention is to add 5 to 10 and then multiply the sum by 3, you have to modify the order of precedence with parentheses. You can do that thus:

```
(5 + 10) x 3 = 45
```

It's clear that parentheses have a higher precedence than the add, subtract, multiply, and divide operators. They redirect the order of calculation.

You learned all of this in school and Crystal Reports follows the same rules of precedence. But Crystal Reports uses many additional operators, and it's important for you to understand the precedence Crystal Reports assigns to each so you can write your formulas to perform as expected.

- In the following list, Crystal Reports performs the top level operations first, then the second level, then the third, and so forth.
- When it encounters two or more operations that are on the same level, it performs them left to right.

Level 1

Parentheses, Array, If Then Else

Level 2

Functions, Subscript

Level 3

+ sign in front of value, Negate, Dollar, Not

Level 4

Multiply, Divide, Percent

Level 5

Add, Subtract

Level 6

To

Level 7

Less than, Greater than, Greater than or equal, Less than or equal, In

Level 8

Equal, Not equal

Level 9

And

Level 10

Or

Precedence Examples

Example 1

```
      If 24 in [(7■1) *4, 7
■1*4 ] Then
      "Hit"
Else
      "Miss"
```

This formula consists of the following components:

Order of appearance	Order of precedence
If■Then■Else operator	Parentheses
In Array operator	Make Array operator
Make Array operator	If■Then■Else operator
Parentheses	Multiply operator
Subtract operator	Subtract operator
Multiply operator	In array operator

- The calculation in the parentheses is done first. That redirects the order of calculation so that (7 ■1)*4 = 24 while 7 ■1*4 = 3.
- Crystal Reports makes the array next. The array consists of two values: 24 and 3.
- Finally, the If then else operator uses the calculated values in the array as the *If* condition.

Example 2

```
      Average([ToNumber("12345"[3 to 4]), ToNumber("2468"[2 to 3])])
```

This formula consists of the following components:

Order of appearance	Order of precedence
Average	Parentheses
Parentheses	ToNumber
ToNumber	Subscript
Subscript	Average

- This calculation uses the function ToNumber to convert the strings "12345" and "2468" to numbers.
- Once converted, it uses the Subscript operator to pull out the 3rd and 4th digits in the first number (34) and the 2nd and 3rd digits in the second (46).
- These two numbers are then averaged to arrive at the number 40 (80/2 = 40).
- The Average function is on hold to the very end because the rest of the calculations take place within parentheses which gives them primary precedence.

Example 3

```
      ToText(Abs({file.quota}■{file.sales})/{file.quota} * 100) + "%"
```

This formula consists of the following components:

Order of appearance	Order of precedence
ToText function	Parentheses
Parentheses	Abs function
Abs function	Divide operator
Divide operator	Multiply operator
Multiply operator	ToText function
Concatenate operator	Concatenate operator

- The work inside the parentheses is done first. If there are parentheses inside parentheses, the work in the innermost parentheses is done first. Thus the subtraction of {file.sales} from {file.quota} is performed first.
- The Abs function is performed next because it is inside the primary parentheses and because it takes precedence over the divide and the multiply operators that are also inside the parentheses.
- Once the absolute value of the difference between {file.quota} and {file.sales} is calculated, that value is divided by {file.quota} and the result is multiplied by 100. (The divide and multiply operators have equal precedence so they are used in the order they appear from left to right.)
- The result of this calculation is then converted to text using the ToText function. Now we're outside the parentheses so the ToText function takes precedence over the Concatenate operator (+).
- Finally, the percentage character "%" is concatenated to the calculated value which has been converted to text. This creates one continuous text string.

Concatenated text strings

Concatenated text strings are simply strings of text that are tied together via a formula. Typically the first string exists in one field and the second string exists in another. Alternately, the first string exists in one field and it is combined with text that is typed directly into the formula.

One typical use of a concatenated text string is in the salutation of a form letter. The word "Dear" is typed directly into the formula and it is to be combined with the title from the *{file.Title}* field and the last name from the *{file.Lname}* field. The concatenation operator can be used to tie all three strings together.

When concatenating, there are a few simple things to keep in mind:

- All text that's typed directly into a formula must be enclosed in quotation marks.
- Text that's a value in a field doesn't require quotation marks; referencing the field is sufficient.
- Finally, if a space is to appear between two concatenated strings, the space must be entered within the quotation marks, either at the end of the first string or at the beginning of the second.

Multiple field sorts

When sorting, Crystal Reports first sorts the entries (alphabetic or numeric) in the first field selected, putting them in ascending or descending order as specified. Then it sorts any entries in the second field that can be sorted *without disturbing the* sort order *of entries in the first field.* It then sorts any entries in the third field that can be sorted *without disturbing the sort order of the entries in the first two fields.* It follows the same pattern for sorting additional fields.

Example

Assume the following data:

#	Acct	Batch	Source	Ref	Type
1	4900	6	AD	A	1
2	4900	1	AF	+	2
3	8500	6	AB	&	1
4	8500	5	AE	k	1
5	4900	6	AA	&	3
6	8500	4	AC	&	2
7	4900	6	AD	1	4
8	8500	4	AC	A	1
9	9600	1	AA	A	1
10	4900	6	AD		1
11	4900	6	AD	A	2

`Sort 1 ▪ One field`

If you tell Crystal Reports to sort the Acct field only, in ascending order, your sorted data will look like this:

#	Acct	Batch	Source	Ref	Type
1	4900	6	AD	A	1
2	4900	1	AF	+	2
3	4900	6	AA	&	3
4	4900	6	AD	1	4
5	4900	6	AD		1
6	4900	6	AD	A	2
7	8500	6	AB	&	1
8	8500	5	AE	K	1
9	8500	4	AC	&	2
10	8500	4	AC	A	1
11	9600	1	AA	A	1

All the values in the Acct field have been sorted, and the rows of data have been moved into new positions according to the order of values in the Acct field. Note that there is no apparent sorting yet of the data in the other fields.

`Sort 2 ▪ Two field`

If you tell Crystal Reports to sort first on the Acct field and then on the Batch field, your sorted data will

look like this:

#	Acct	Batch	Source	Ref	Type
1	4900	1	AF	+	2
2	4900	6	AD	A	1
3	4900	6	AA	&	3
4	4900	6	AD	1	4
5	4900	6	AD		1
6	4900	6	AD	A	2
7	8500	4	AC	&	2
8	8500	4	AC	A	1
9	8500	5	AE	k	1
10	8500	6	AB	&	1
11	9600	1	AA	A	1

Notice how the Batch entries for Acct 4900 (Rows 1■6) and for Account 8500 (Rows 7
■10) are now sorted numerically. Notice too that the remaining three fields have not yet been sorted (for example, Rows 2
■6 in the Source field are out of alphabetical order.

```
Sort 3 ■ Three field
```

Now, if you tell Crystal Reports to sort on the Acct field, then on the Batch field, and then on the Source field, Crystal Reports sorts the Source field alphabetically for each unique Account/Batch combination.

#	Acct	Batch	Source	Ref	Type
1	4900	1	AF	+	2
2	4900	6	AA	&	3
3	4900	6	AD	A	1
4	4900	6	AD	1	4
5	4900	6	AD		1
6	4900	6	AD	A	2
7	8500	4	AC	A	1
8	8500	4	AC	&	2
9	8500	5	AE	k	1
10	8500	6	AB	&	1
11	9600	1	AA	A	1

The values in the first three fields are now sorted. The sorting process continues in the same way for additional sort fields.

Dates stored in text or number fields

Crystal Reports allows the use of date fields (fields specific to the date data type) for storing dates. Date fields allow for storing the date in a month/day/year or similar format.

Some users, however, prefer to store dates in text or number fields. In these cases they store the date as a serial or date number (some number of days since a base date).

Crystal Reports allows you to work with either type of date entry.

- Dates stored in date fields require no conversion; they are stored as dates and can be used directly as dates.
- Dates stored in number or text fields, however, must be converted to report dates before they can be used.

Converting a date number (number field) to a report date

To convert a date number stored in a number field to a report date, use the following formula:

```
Date(yyyy,mm,dd) + DateNum
```

«where (yyyy,mm,dd) is the base date in year, month, day format, and DateNum is the date number (the number of days since the base date).»

What Crystal Reports is doing in this situation is converting the base date to a report date using the Date function.

Then, once the date is converted, it is adding the number of days since the base date to the base date to arrive at the desired report date.

Converting a date number (text field) to a report date

To convert a date number stored in a text field to a report date, use the following formula:

```
Date(yyyy,mm,dd) + ToNumber(DateNum)
```

«where (*yyyy,mm,dd*) is the base date in year, month, day format, and *DateNum* is the date number (the number of days since the base date).»

This formula works in the same way as the previous formula (for converting numbers to dates), but with one additional step. The ToNumber function converts the number stored as text to an actual number that can be added to the base date. Numbers stored as text cannot be used in calculations without first being converted to numbers via the ToNumber function.

Formatting numbers stored in text fields

Employee numbers, customer numbers, ZIP codes, and other similar numbers are often stored in text fields. When a number is stored in a text field, its data type may be unclear to us, especially if you are working with unfamiliar data. Only when you select the Format|Field command does the data type become readily apparent.

The Format|Field command is the command you use for changing the format of a selected field. It presents a set of options that are specific to the data type of the selected field. If a number is stored in a text field it is considered to be text, even though it looks like a number. Thus, the Format|Field command presents only text formatting options for such a number.

See Also

Crystal Reports data types

Index to formula topics

Paradox secondary index files

Crystal Reports Version 1.1 and newer support Paradox secondary index files (.x??, .y??) in addition to Paradox .px index files. You can select a secondary index with the Link Options dialog box, which is accessed via the Visual Linking Expert feature.

Using memo fields with Crystal Reports

Paradox IV, dBASE, and Btrieve all allow the use of memo fields, fields that can contain large blocks of text or mini-documents. While earlier versions of Crystal Reports allowed a limited use of memo fields, Versions 1.1 and newer offer comprehensive memo field support.

- You can place a memo field in your report as you would any other field and the entire memo field will print (earlier versions printed only the first 80 characters).

  By default, Crystal Reports allots a space 20 characters wide for each memo field.

- You can expand or narrow a memo field in the same way you expand or narrow any field (by dragging one of the handles on the field box, or by using the Shift
-Right Arrow or Shift
-Left Arrow combination).
- The program will word
-wrap within the space allotted if you have activated the *Print on multiple lines* option in the Format Memo dialog box. If you expand or narrow the field, word wrap will adjust to the available space (where possible), again, if the Print on multiple lines option is activated.
- If your memo field includes return characters, Crystal Reports accepts and interprets them to provide line breaks in the report memo field where they occurred in the original (where possible).

NOTE: You can limit the number of lines on which your memo fields print using the Format Memo dialog box.

NOTE: You can not select memo fields in record or group selection formulas.

NOTE: If records contain two or more memo fields, Record B doesn't begin to print until the longest memo field in Record A is done printing

The Status Bar

The status bar at the bottom of the report window displays valuable information to help you use Crystal Reports more efficiently:

Button bar functions

When the cursor is over a button on the Button Bar, the Status Bar displays a short description of the button's function. For example, the following Status Bar entry (right side of Status Bar) appears when the cursor is over the Insert Summary button:

Menu command descriptions

When you highlight a menu command, the Status Bar displays a short description of the command . For example, the following Status Bar entry (right side of Status Bar) appears when you highlight the Database|Verify on Every Print command:

To highlight a menu command, click the menu name and move to the command using the Down Arrow key.

Current selections

When you select or place a picture, field, text field, graphic line or box, special field, or formula, the status bar displays the name of the item selected. It displays:

- the file name for a picture,
- the alias and field name for a field,
- the text in a text field,
- the words Line for a line and Box for a box,
- the field type for special fields (PrintDate, RecordNumber, etc.) and
- the formula name for a formula.

Graphic coordinates

When you select or create a bitmapped picture, a graphic box, or a graphic line, the program displays its coordinates.

Resizing sections

The Design Window first appears with default section sizes.

You can expand or reduce report sections by dragging the lines that separate the sections. When you position the I-beam cursor over one of those lines, the cursor changes to a double

-arrow resizing cursor. Once that cursor appears, you can resize as needed.

Alternately, to expand a report section, you can click the section of interest and press Enter as many times as needed. Once you've expanded a section, you can reduce by deleting unneeded lines with the Backspace key.

Finally, you can click the Right Mouse Button in the gray area to the left of the Design Window in the section you want to resize. Select Add Line from the pop-up menu that appears. This will add a single line to the selected section. To reverse the process, select Delete Line from the pop up menu.

Bit-mapped picture concepts

Bit-mapped pictures are the kind of pictures (logos, etc.) that are generally produced by paint programs and scanners. They are composed by the graphic designer as a pattern of tiny dots (pixels) on screen, and they are printed as a pattern of tiny dots on your report. Even though there are some limitations to what an individual can create with dots, a skilled graphics designer can nonetheless achieve some stunning effects that can add visual impact to your report.

Crystal Reports allows you to use bit-mapped pictures in your reports from a wide variety of sources:

- scanners,
- paint programs,
- video capture cards,
- screen capture programs,
- CompuServe,
- commercial graphics developers, and
- shareware and public domain picture suppliers.

As long as the picture is stored in one of the popular picture formats that work with Crystal Reports, you can use it in your report.

Preview Window

When you want to preview your report before printing, you use the program's Print Preview option. When you select this option, the program gathers the data, makes the necessary calculations, and prints the report in the Preview Window on electronic "paper." With the data in place, you can review the spacing and formatting of your report and see the actual results of all your summaries, formula calculations, and record and group selections.

A different look and feel

Working in the Preview Window has a different look and feel than working in the Design Window.

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 Window, a single field box represents all those values. When you select the field, sizing handles appear on the box and the box changes color.

In the Preview Window, however, you are working with actual data. Instead of a field box representing many field values, the values themselves appear. When you select a field, a sizing box appears around every value from the field selected. Likewise, when you select a summary field, a sizing box appears around every related summary value. While the look and feel is different, the process of building and modifying a report is the same in both the Design Window and the Preview Window. You should find it easy to work with your reports in both places.

No need to rerun the report

When you print your report to the Preview Window, the program retrieves the required data from the specified database(s). Depending on the size of your database(s) and the complexity of your report, this retrieval process can take anywhere from a few moments to an hour or more. If the retrieval process takes a long time, you clearly don't want to repeat it more often than necessary. The fine tuning capability in the Preview Window was designed with this in mind. Here's how it works.

Let's assume that you create a report that includes fields A, B, and C. When you call up the Preview Window, the program retrieves the data in fields A, B, and C from the active database(s) and holds it temporarily with the report. As long as you're working with the retrieved data, you can manipulate it in any number of ways and the program will never have to go back to the database.

If you want to add additional data once you're in the Preview Window, however, you force the program to run the report once again. It has to go back and retrieve the original data as well as your new requirements. Procedures that force the program to rerun the report include:

- adding databases, tables, links, or fields that weren't in the original report,
- adding formula fields referencing fields that weren't in the original report, and
- revising record selection criteria to include records not already included in your report.

With some simple planning you can avoid these revisions and the extra processing time they entail. Here's how:

1. Build your original report with all of the records, fields, and formulas that you think you might possibly need. This causes the program to retrieve all of the necessary data the first time it runs the report. (If you don't include everything in your final report, you've lost nothing.)
2. Rework the report in the Preview Window until you get it the way you want it. Since you brought in all the data the first time, there's no need for the program to go back and get it again.

NOTE: You can force the program to go back and get fresh data any time you want by using the Report|Refresh Report Data command or by Clicking the refresh (lightning bolt) button on the button bar.

NOTE: Don't get carried away and try to include every possible database, field, and link. At some point your initial processing time penalty can outweigh any potential time savings you are seeking.

Saving your data with the report

The program gives you the option of saving only the report definition or saving your data with the report as well.

- If you save only the definition:
- your report will require less disk space (enough for the definition only), but
- you will need to rerun the report before you print it.
- If you save your data with the report:
- your report will require slightly more disk space (enough for the report definition and the compressed data), but
- you won't need to rerun the report before you print it.

If the report can run quickly, saving the data with the report may not be important to you. But if your report takes time to run, you may find the Save Data with Report option a real convenience.

NOTE: Saving data with your report is also a good idea if you want to save the data for reference later. For example, if you run a report on a weekly basis, you might want to save the data with the report each week. With the data saved, you can make historical comparisons easily, working with the actual data that existed at the time each report was created.

When you open a report that you have saved with the data, the report template appears in the Design Window followed by the Preview Window displaying actual report data. The Date and Time displayed in the Preview Window identifies the data used in the report.

See also

File|Print Preview

Report Templates

Instead of starting from scratch when building a report, you might find it easier to begin with a template from a similar report that you created earlier.

A template is a copy of a report. It contains all of the data, links, formulas, and pictures that your original has, but *it isn't tied to your original report in any way*. It's a separate entity. When you modify a template, your original report remains unchanged.

Templates boost your efficiency. Use them whenever you think they can save you time.

ODBC ▪ An Overview

Click here for a diagram of ODBC

The Microsoft Open Database Connectivity(ODBC) standard is a program interface that allows a single application such as Crystal Reports to access data stored in various Database Management Systems (DBMS) using a common language. The ODBC system serves as a translator between an application and the data it is trying to access.

The ODBC translation is a two step process. First, an application such as Crystal Reports must be able to talk to ODBC. Second, ODBC must be able to talk to the DBMS.

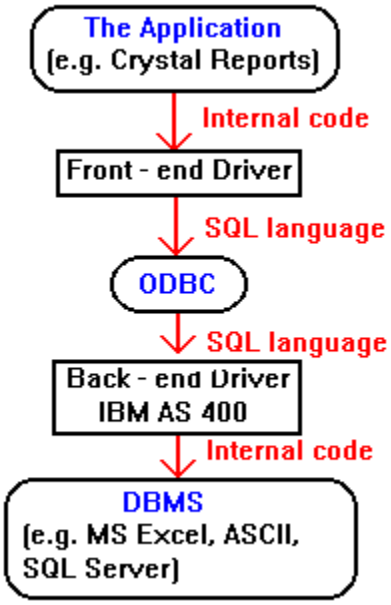
- To talk to ODBC, the application uses a Dynamic Link Library specially designed for the task. This DLL is called a **Front end ODBC driver**.
- To talk to the DBMS, the DBMS must also provide a DLL to allow ODBC to talk to its database files. The DLL provided by the DBMS is called a **Back end ODBC driver**.

The ODBC system sits between the drivers as a common standard by which they can communicate using the Structured Query Language (SQL).

See Also

Using Crystal Reports with Microsoft Access

How an application uses ODBC to access DBMS systems:



Crystal Reports and ODBC

Crystal Reports comes with front end drivers to make it fully compatible with the ODBC system. In addition, the Crystal Reports package includes back end ODBC drivers that allow you to use data stored in Microsoft Access, Microsoft Excel, and ASCII text files when creating reports.

Though Crystal Reports is compatible with many popular database formats, you may be using a DBMS that is not directly available through Crystal Reports. Your DBMS may, however, provide an ODBC driver that allows Crystal Reports to use your data through the ODBC system. To find out if your DBMS provides ODBC drivers, consult the documentation that came with the DBMS software, or contact the software vendor that sells the DBMS system.

Logging onto an SQL server (or other ODBC data source)

NOTE: While this topic emphasizes logging on to a SQL Server, accessing any ODBC data source is done in much the same fashion. Your data source may or may not require that you enter sign on and password information.

Crystal Reports provides three ways of logging on to a SQL server:

- logging on as part of the report creation process, and
- logging on outside the report creation process.

Logging on while creating a report

You can log onto a SQL server as part of the process of activating a SQL table for use in a report.

Whenever you select File|New|Custom Report, File|New|Custom CrossTab, File|New|Custom Mailing Label, or Database|Add Database to Report, the Choose Database File dialog box appears. You can use this dialog box to select a non-SQL database file for use in your report, but you can also use it as a gateway for logging on to a SQL server and activating a SQL database and table for use in your report.

To activate a SQL database/table during the report creation process

1. To begin the report creation process, do one of the following:

- if you want to select the first SQL table for use in a report (and no other non-SQL databases or tables are active), select File|New|Custom Report.
- if you want to select the first SQL table for use in a mailing labels report (and no other non-SQL databases or tables are active), select File|New|Custom Mailing Label.
- if you want to select the first SQL table for use in a report (and other non-SQL databases are already active), select Database|Add Database to Report. The Choose Database File dialog box appears.

2. Click the SQL Server button. The Log On to Server dialog box appears listing the various SQL server types available on your system. Select the server type that you want to log on to and Click OK when finished.
3. A dialog box appears requesting server-specific logon information. You use this dialog box to identify yourself and to specify the database you want to activate.

NOTE: If you want to activate multiple databases from the same server, you will need to log on to the server each time you want to activate a database.

4. Some of the following items will be in the dialog box (depending on the server type requested).

SQL Server

Enter the name of the SQL server you want to log on to.

Database

Enter the name of the database you want to activate in the specified SQL server.

User ID

Enter the name you use to log on to the specified server.

Password

Enter the password you use to log on to the specified server.

Dict Path

When using Netware SQL, enter the path for the data dictionary (.dbf) files.

Data Path

When using Netware SQL, enter the path for the data files.

Enter the requested logon information and Click OK when finished. Crystal Reports logs you onto the specified server and takes you to the Choose SQL Table dialog box.

5. The Choose SQL Table dialog box works in a similar manner to the Choose Database File dialog box. You use the Choose SQL Table dialog box to select the table you want to activate for use in your report.

Logging on outside the report creation process

You can log onto a SQL server using the Database|Log On Database Server command when you first call up Crystal Reports (or at any other time while using the program). The Log On Database Server command was created for those times you want to log onto a SQL server to have the server and database activated and standing by for later use in a report. For example, if you want to review or revise an existing report and then create a new report using a SQL table, you might log on to the SQL server and activate a database when you first call up Crystal Reports. Then, when you're finished revising the existing report, you can select a SQL table from the active server/database and create your new report.

To log on, use the Database|Log On Database Server command.

Activating a second table (and additional tables) from a SQL database

When you have already activated one table in a SQL database and you want to activate an additional table from that database, use the following procedure:

1. Select Database|Add Database to Report. The Choose SQL Table dialog box appears.
2. Select the table you want to activate and Click OK when finished.
3. The Visual Linking Expert dialog box. appears. Follow the instructions for Visual Linking to complete the link and specify any options.
4. Repeat Steps 1■3 for each additional table you want to activate from the active SQL database.

NOTE: If you have logged on to a SQL database and you want to activate a new table but from a different SQL database, Click the Log On Server button in the Choose SQL Table dialog box. This returns you to the Log On To Server dialog box where you can select the server type for the next database you want to activate.

Setting SQL defaults

Crystal Reports enables you to speed up the SQL table selection process by setting SQL defaults via the File|Options command. Once your defaults are set, you can bypass one or more dialog boxes on the way to selecting a SQL table for use in a report. The SQL options you can set via the SQL Options button either optimize your way into SQL or control the set of tables you are allowed to report on.

To set up your SQL defaults:

1. Select File|Options. The File Options dialog box appears.
2. Click the SQL tab. SQL Options appear.

Hide and Show

The Hide and Show options allow you to hide a report section that is showing or to show a report section that is hidden. Hide and Show are available through two methods: the right mouse button menu and the Show/Hide Sections dialog box.

Right mouse button menu

- When you click a visible report section in the gray area at the left of the Design Window and then click the right mouse button, the Hide option appears on the pop up menu. Click Hide and Crystal Reports hides the selected section.
- When you click a hidden report section in the gray area at the left of the Design Window and then click the right mouse button, the Show option appears on the pop up menu. Click Show and Crystal Reports redisplays the selected section.

NOTE: The Status Bar displays the name of the section the cursor is in when the cursor is in the gray area to the left of the Design Window. This includes hidden sections. When a hidden section name appears on the Status Bar, you can Click the right mouse button and select Show from the pop up menu to redisplay the section.

Show/Hide Sections dialog box

1. The Show/Hide Sections dialog box is available through two methods:

- When you Click in the gray area at the left of any section (visible or hidden) in the Design Window, the Show/Hide Sections option appears just below the Show or Hide option. Click Show/Hide Sections and the Show/Hide Sections dialog box appears, or
- Select the Show/Hide Sections command from the Edit menu. When you select this command, the Show/Hide Sections dialog box appears.

2. Each section that appears in your report is listed in the Sections list box with either an S or an H to the left of the section name. For example S ■ Details.
   - S means the section is currently shown.
   - H means the section is currently hidden.
3. Select the section you would like to change the status of in the Sections list box.
4. Click either the Show section or Hide section option button at the bottom of the dialog box to change the status of the section. The letter next to the section name in the Sections list box will change accordingly.
5. Click OK, and Crystal Reports returns you to the Design Window:

- If you changed the status of a visible section to Hidden, the section will be hidden in the Design Window.
- If you changed the status of a hidden section to Show, the section will be opened and visible in the Design Window.

Support Questions and Answers Contents

Following is a list of indexes to topics in Crystal Reports that may help to solve problems or questions you may have while creating your report.

Click on the topic of interest.

Environment (windows, mouse, grid, etc.)

Command topics

Databases topics

Formulas, functions, and operators

Installation, configuration, and setup

Visual Linking Expert topics

Report layout and formatting

Selection formulas

Grouping and Sorting topics

Environment (windows, mouse, grid, etc.)

Click on a topic below for more information:

- Using program with Norton Desktop for Windows
- Source of Paradox network error messages
- Source of Server Not Found message
- Using Share with Crystal Reports

Command topics

Click on a topic below for more information:

- Using Cut, Copy, Paste in the Formula Editor
- Removing databases using Remove from Report command
- Finding fields Remove from Report says to remove
- Cutting and copying fields in the Design Window
- Sources of Error Recognizing File message

Databases topics

Click on a topic below for more information:

- Finding existing index for lookup database
- Activating/linking additional databases
- Paradox secondary index files

Formulas, functions, and operators

Click on a topic below for more information:

- Using Cut, Copy, Paste in the Formula Editor
- Formula Compiler Warnings

Installation, configuration, and setup

Click on a topic below for more information:

- Making certain program can find DLL's

Report layout and formatting

Click on a topic below for more information:

Formatting numbers stored in text fields

Selection formulas

Click on a topic below for more information:

Fine tuning record selection formulas

Making certain that Crystal Reports can find DLL's

Crystal Reports utilizes several DLL's (Dynamic Link Libraries) during its operation. DLL's are program modules that each perform specific tasks. It follows that Crystal Reports must be able to find these DLL's in order to operate properly. When Crystal Reports is installed, dll's are placed in new directories and these changes are reflected in the path. For this reason, it is important that you reboot your PC after you have installed Crystal Reports. Rebooting activates the new path statement that includes the directory that contains the DLL's. Without rebooting, the computer doesn't know that CRW is in the path, and Crystal Reports will not know where to look for the needed DLL's.

Using program with Norton Desktop for Windows

If you are using Norton Desktop for Windows and you find Crystal Reports to be behaving strangely (you can't use the mouse, etc.), the problems can probably be traced to certain Norton Desktop for Windows incompatibilities with Borland's Object Windows Library (OWL.DLL) which is used by Crystal Reports.

To correct the problem

These problems were a known problem with Version 1.0 of Norton Desktop for Windows. The developer of that program says that the problems have been corrected in Version 2.0.

If you are using Crystal Reports with Version 1.0 of Norton Desktop for Windows, one suggested method for trying to fix the problem is as follows:

1. Go into the Norton Desktop for Windows control panel.
2. Turn the launcher and launch manager off.
3. Reboot your system and call up Crystal Reports again.

Removing databases using Remove from Report command

When using the Database|Remove from Report command, you may find yourself selecting a database to delete in the Databases box (Remove from Report dialog box)and then clicking Done to delete it. The Remove from Report dialog box disappears, and you assume that the selected database has been deleted. You later find that it is still an active database.

How to correct this problem

When you select a database in the Databases box, *clicking Done does not delete it*. *You must Click the Remove button* to complete the deletion process.

Finding fields Remove from Report says to remove

When using the Database|Remove from Report command, you get the following message if there are fields in the active report from the database you are trying to delete:

```
There are fields in the report from this file.
Please remove them before deleting the file.
```

If you have looked for the fields and can't find them, print the report definition (to the printer or Preview Window) using the File|Print|Report Definition command. By reviewing the report definition carefully, you should be able to locate those places in which the fields are used. More than likely you will find the fields hidden in selection formulas, report formulas, or sort criteria.

Cutting and copying fields in the Design Window

You may find yourself trying to use the Cut, Copy, and Paste commands on fields and formulas in the Design Window without success. The Cut, Copy, and Paste commands work only with text in the Design Window, not with fields. This is true whether you select the commands from the Edit menu, the Button Bar, or the keyboard.

To cut fields in the Design Window:

Select the field. At this point you have three choices:

- press the Delete key,
- click the right mouse button and select Delete Field from the menu that appears, or
- select the Clear command from the Edit menu.

To copy fields in the Design Window

Since there is no Copy command available for use with fields in the Design Window, enter the field again for each additional copy you need.

Sources of Error Recognizing File message

If you are getting an Error Recognizing File message, you may find the source of the problem in the following discussion:

If the problem occurs when you try to select a report

There are two typical situations that result in the Error Recognizing File message when you are trying to activate a report:

Situation 1

It is possible that you have tried to open an existing report via the File|New command instead of the File|Open command.

- The File|New command is the command you use to begin the creation of a new report.
- The File|Open command is the command you use to open an existing report.

When you select File|New|Custom Report and select a file with the extension .RPT in the Choose Database File dialog box, Crystal Reports responds with the error message.

Conversely, when you select File|Open and attempt to select a file with an extension other than .RPT, you get an Error Loading Report message.

Situation 2

If you have used the correct menu command and are still getting the error message, it is possible that the .RPT file you have tried to open is one generated by Paradox or dBASE instead of by Crystal Reports. Crystal Reports only recognizes those .RPT files that it has generated itself.

If the problem occurs when you try to select a database

There are two typical situations that result in the Error Recognizing File message when you are trying to activate a database:

Situation 1

It is possible that you are using the correct command (File|New), but that you are selecting a database that is not a compatible database. In such a case, review your selection and make certain that you are selecting a database that works with Crystal Reports.

Situation 2

If you are trying to activate a Btrieve database, you may be selecting a .DAT file instead of a .DDF file. Crystal Reports cannot use .DAT files because they are user definable and don't store any dictionary information. The necessary information is stored in the .DDF files. Select the appropriate .DDF file and try again.

CRPE.DLL directory must be in path

If you are using the Crystal Reports print engine, and calls to the print engine don't find CRPE.DLL, your path statement may not contain the directory in which the DLL is stored.

Crystal Reports automatically adds the directory to your path statement during installation unless you direct it not to. If you have directed it not to add the directory to your path statement, not entered the directory to the path yourself, and not included the path as part of the declaration statement for each of the print engine calls, Crystal Reports doesn't know where to find CRPE.DLL.

How to correct this problem

To correct this problem, do one of the following:

- add the directory containing CRPE.DLL to your path statement, or
- include the full path with each declare or register statement for each print engine call (c:\directory\CRPE.DLL, not just CRPE.DLL).

Fine tuning record selection formulas

You may have run into a situation in which you create a record selection formula (using Report|Edit Record Selection Formula), and, while header and footer information prints on your report, no detail information appears. The problem is a selection formula that is rejecting all records, and this usually occurs because of some inadvertent error in the creation of the selection formula.

Errors to Avoid:

Upper/lower case inconsistencies

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 *{file.FirstName}* field, but all the entries in the *{file.FirstName}* field are mixed case ("Bob", for example), the selection formula will find no matches and thus not print any details for the report.

- You can solve this problem by using the UpperCase or LowerCase functions in your selection formula to convert field data to a consistent case before Crystal Reports begins its selection. For example, if you were using this formula:

  ```
  {file.FirstName} = "BOB"
  ```

  you can change the formula to this:

  ```
  UpperCase({file.FirstName}) = "BOB"
  ```

  and get the results you want. This last formula first converts the value of the *{file.FirstName}* 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" "b" will be a match, regardless of case, because the case will be converted first to uppercase for consistency.
- 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 case correct on any text you are trying to match. If in doubt, use the UpperCase (or LowerCase) function to assure consistency and proper matches.

NOTE: Another formula that does much the same as that above is:

```
"BOB" in UpperCase({file.FirstName})
```

Number in text field not in quotes

When a number is stored in a text field, it is text even though it looks like a number. Whenever you use a value from a text field in a record selection formula, you must surround the value with single or double quotation marks. If your selection formula is set to look for a number in a text field and you fail to surround the number with quotes in the selection formula, the selection formula will find no matches and thus not print any details for the report.

For example, this selection formula:

```
{file.CustNum} = 12345
```

won't find any matches, even though the value 12345 appears in the *{file.CustNum}* field of many records. To select records with the characters 12345 in a text field, you must put quotes around the characters you're attempting to match, like this:

```
{file.CustNum} = "12345"
```

Check your selection formula closely, and make sure that any numbers you are attempting to match in a text field are surrounded by single or double quotation marks.

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 {file.FormAddrs}
```

won't find any matches with the form of address "Mr." because there is an extraneous 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.

Troubleshooting record selection formulas

If you contact the company with a record selection formula problem, technical support will ask you to perform these steps on your formula. You can save a considerable amount of time by working through the process on your own. Then, if you still can't find the error, contact technical support.

Source of Paradox network error messages

If you receive a Paradox network error message but you're not on a network, the source of the problem is usually incorrect settings in the Paradox Engine Configuration Utility (PXENGCFG.EXE). Faulty settings can tell the Paradox engine that you're running on a network when in fact you are not.

The configuration utility defaults to non-network use, but if someone changed the settings for network use and you then tried to run in a non-network mode, you may begin experiencing network error messages.

To correct the problem

1. Select Run from the File Menu in the Windows Program Manager. The Run dialog box appears.

NOTE: Use Start | Run from the Taskbar in Windows 95

2. Type in PXENGCFG.EXE and press Enter. (No path statement is necessary unless you have previously installed the utility in a different location.) The Paradox Engine Configuration Utility dialog box appears.
3. Select Network Configuration. The Paradox Engine Network Configuration dialog box appears.
4. Change the PARADOX.NET Path setting to C:\ (the default setting).
5. Click OK when finished to return to the previous dialog box.
6. Click Save to save the new settings, and then select File|Exit from the menu in that dialog box to return to the Windows environment.

Using Share with Crystal Reports (Windows 3.1 only)

SHARE.EXE is a DOS program that provides certain networking and multi-tasking functions. Some programs will only work if SHARE.EXE is first loaded; others don't have such a need.

When working with Crystal Reports, you need to load SHARE.EXE if you will be working with Paradox files or Object Linking and Embedding (OLE) objects. Here is why:

The Paradox engine (PXENGWIN.DLL) requires that SHARE be loaded before it will function. If you attempt to activate a Paradox file without first loading SHARE, the Paradox program PDBPDX.DLL attempts to load the Paradox engine, finds that SHARE isn't loaded, and issues an error message. Your only option is to exit the program, load SHARE, and resume.

- If you are not using Paradox files and if disk storage space is limited, you can delete the two Paradox-related DLL's, PXENGWIN.DLL and PDBPDX.DLL from your hard disk.

OLE allows a user to embed or link an object created in one application to a file created in a second application. Because OLE lets you work with data from more than one application at a time, SHARE.EXE must be loaded on your system.

If you are unsure whether you will need to use Paradox files or OLE, you should go ahead and load SHARE.EXE onto your system in case you need it later.

Source of Server Not Found message

When Crystal Reports searches for the program file CRPE.DLL, it will look first in the current directory, then in the Windows directory, then in the Windows system directory, and finally in the path. Since the installation procedure automatically installs the program in the CRW directory and updates the path statement in the AUTOEXEC.BAT file (unless you have selected different options), the program should have no problem finding the CRPE.DLL file under normal circumstances.

If you have installed Crystal Reports to a directory other than the default directory, and if you have not modified the path statement to include the new directory, you may get a *Server Not Found* message.

How to solve this problem

To recover from a *Server Not Found* message, modify the path to point to the directory that holds CRPE.DLL, or move the program to the Windows directory, the Windows system directory, or a directory in the current path.

Deleting blank lines from your report

- By default the Design Window allots three lines for the Page Header section and three lines for the Page Footer section of your report. The defaults may allot more lines than you need for those items on your report.
- Additionally, you may expand a section on your report by a random number of lines prior to inserting text and data, just to make sure you have enough room for your entries. You may find that you have added more lines than necessary

Printing the report without first deleting the unneeded blank lines can leave gaps in your report that make the report less attractive visually and more difficult to read.

To delete unneeded blank lines

If an entire section is blank (i.e., if you aren't putting anything into the Page Footer section of your report), you can eliminate the allotted blank lines by eliminating the entire report section via the Hide Section option of the Format Section command.

If you have text and/or data in a section and just want to remove the extraneous blank lines, Click the I■ beam cursor on the blank line you want to delete. This sets the insertion point. Once the insertion point is set. press the Backspace key (the key that deletes the previous character); Crystal Reports deletes the line on which the insertion point is set.

Example

Assume that you have entered text in the first line of the Page Header section and that you have entered data fields in the details section. You want to delete the bottom blank line in the Page Header section. To do this:

1. Position the I■beam cursor on the last (bottom) line of the Page Header section and Click the left mouse button to set the insertion point. The insertion point appears at the left edge of the Design Window text box.
2. Press the Backspace key one time. The bottom blank line disappears.

To delete the remaining blank line

When you deleted the bottom blank line in the Page Header section, the insertion point moved up to the remaining blank line (what had been Line 2, the center line in the section). To delete this line, press the Backspace key once again. Now all that remains is the line of text you entered in the section.

Using Cut, Copy, Paste in the Formula Editor

The Cut, Copy and Paste commands on the Crystal Reports Edit menu and on the Button Bar do not work in the Formula Editor. The Windows' keyboard-activated Cut, Copy, and Paste commands do work in the Formula Editor, however.

Crystal Reports and REG.DAT

The Microsoft Windows registration database (REG.DAT) does not list Crystal Reports as an OLE 2.0 compatible application. Crystal Reports does, however, support OLE 2.0.

A report designed in Crystal Reports can act as an OLE Container application, but not as an OLE Object application. The REG.DAT registration database only registers OLE 2.0 object applications. These are the applications that appear, for example, in the Object Type list box of the Insert Object dialog box.

Since Crystal Reports is not an Object application, you will not see it listed in the Registration Info Editor (REGEDIT.EXE) when you view the registration database. Be aware, though, that Crystal Reports does support OLE 2.0 as a full featured Container application.

How to identify the "top" groups

A group is a set of records that are related to each other in some way. In a customer list, for example, a group could consist of all those customers living in the same ZIP code, or in the same state. In a sales report, a group could consist of all the orders placed by the same customer, or all of the orders generated by a specific sales rep. Crystal Reports allows you to group data in a variety of ways (see grouping and sorting data)

One of the reasons you create reports may be to identify "top" groups: the best sales reps, the biggest customers, the states with the most customers, etc. Conversely, you may want to identify the "bottom" groups, the products with the fewest returns, the cars with the lowest operating cost, the SKUs with the lowest sales, etc.

Identifying the top or bottom N groups is extremely useful when you're creating Cross-Tab reports or graphs. By limiting the number of groups that appear in your report, you make your Cross

-Tab report or graph much easier to read and to understand.

Identifying these "top" groups is easy with Crystal Reports following these steps:

- create the report,
- group the data and summarize the data in each group, and
- specify the number of groups you want to print using the Top N/Sort Group Expert.

Creating the report

You create the report (activate the database(s), insert the fields, etc.) following the steps discussed under Creating a report.

Grouping and summarizing the data

You group and summarize the data (sum, average, or count the values in each group, calculate the variance or standard deviation, or determine the maximum or minimum value in the group) following the steps discussed in Grouping data with Crystal Reports (See grouping and sorting data).

Sorting the summarized groups

You sort the summarized groups using the steps discussed for the Report|Group Sort Order command.

How to create a table

NOTE: Crystal Reports contains two table styles in the Report Style Expert, Table and Drop Table. These styles create tables automatically and in most cases should be sufficient for your needs. If you need to create a custom table, you can do it manually using the following procedure.

You can create attractive tables for your report quickly and easily using Crystal Reports' box and line drawing functions. A short example will demonstrate the basic principles involved.

NOTE: If you want to reproduce this example yourself using Crystal Reports, you will find the database from which the data is drawn (detail.dbf) located in the CRW directory (or the directory where you installed Crystal Reports sample data).

To create the example table

1. Using *detail.dbf*, place the following fields side by side in the Details section and resize the fields to get the "look" you want:

   ```
   {detail.ORDERNUM}
   {detail.QTY}
   {detail.ITEMNUM}
   {detail.PRICE}
   ```

   Leave a little room between the fields so you can enter table lines.

2. Create the following formula and name it EXTEND:

   ```
   {detail.QTY}*{detail.PRICE}
   ```

   See Formulas ▪ an overview if you need instructions on creating formulas.

3. Place the formula in the Details section to the right of the detail.PRICE field.

4. Subtotal the formula EXTEND so it prints a subtotal whenever the value in detail.ORDERNUM changes.

5. Draw a single box around all the data in the Details section. This will place a box around each row of data that appears in the Details section of your report.

6. Draw vertical lines between each of the fields in the Details section. Each line should extend exactly from the top to the bottom of the graphic box you drew in Step 2.

7. Draw a single graphic box around the subtotal data.

NOTE: Make certain that the right edge of the box aligns with the right edge of the graphic box you drew in Step 5, and make certain that the left edge of the box aligns with the vertical line you drew between the last (rightmost) two fields (Step 6).

Mailing labels and label type items

NOTE: Crystal Reports has a Mailing Label Expert that will lead you step by step through the label creation process. You can use the following procedure if you want to create mailing labels using the File|New|Custom Mailing Labels command.

Using Crystal Reports with your printer, you can easily create:

- address labels,
- shipping labels,
- audiotape and videotape labels,
- diskette labels,
- file folder labels,
- rotary file cards,
- postcards,
- name badges,
- and a host of related items that come mounted for printing in laser, ink jet, or dot matrix printers.

These items:

- come in a variety of shapes and sizes,
- are mounted on single sheet and/or tractor feed paper, and
- are laid out on the carrier paper in a number of different configurations.

Despite this diversity, Crystal Reports enables you to set up and print your data on virtually any of these label-type items quickly, and with a minimum of effort.

Creating label type items the process

There is a three-fold process to creating mailing labels:

- setting the paper size,
- specifying label size and layout, and
- setting up the label stock in your printer.

Setting the paper size

Paper size refers to the size of the carrier paper.

- If you are using laser labels, your paper size will generally be standard letter size, 8 1/2 by 11 inches.
- If you are using dot matrix labels, your paper size may also be the standard 8 1/2 by 11 inches, but it could also be any of a number of irregular paper sizes (4 1/2 by 6 inches, 4 1/2 by 12 inches, etc.)

You set the paper size (if different than the current default) via the File|Printer Setup menu option. If you need to specify a new user-defined paper size, you do that via the Printer section in the Windows Control Panel.

NOTE: On dot matrix labels, the part of the carrier paper that includes the tractor holes should not be included in the paper size width setting. As a rule of thumb, measure the entire width of the carrier paper (including the tractor holes) and then subtract 3/4". Enter the resulting amount as your paper size.

Specifying label size and layout

Labels come in a variety of shapes and sizes and are mounted in a number of different layouts. You specify the label size, layout, and other printing parameters via the File|New|Custom Mailing Label command.

Setting up the label stock in your printer

Crystal Reports assumes that your label stock is set up for printing. If you need help with this aspect of the process, please consult the manual that came with your printer.

General instructions for creating label-type items

The information is general in nature so as to provide guidelines for the entire range of potential uses. You can set up most popular Avery labels simply by selecting the label number of interest from the Choose Mailing Label Type scroll list in the Mailing Labels dialog box.

Setting up labels with borders

In setting up most labels, you measure from label edge to label edge to determine label height and width. Then, if you enter label text on the top line and print your label data flush left, Crystal Reports starts printing slightly below the top edge of the label and slightly to the right of the left edge of the label.

When you use labels with borders, however, the borders often appear right where the printing might begin. Thus, if you measure these labels from edge to edge, you stand a good chance of having your text begin on top of a border instead of inside the borders.

When setting up labels that have borders, you have to factor the border into your measurements. As a general rule when working with labels with borders, treat the white space inside the border as if it alone were the label.

Setting up circular labels

Setting up circular labels is easy to do, but you have to exercise some care in the way you set up your data for printing:

- the shape of the label is limiting.
- Lines of text at the top and bottom of the label cannot be as long as a line of text in the center of the label. And
- the Crystal Reports Label Expert displays only square and rectangular sample labels, so some judgment and trial printing may be necessary to place the data exactly as you want it.

Multi-column "telephone book" reports

In a typical report, data flows in a single stream straight down the page. In a telephone book report, however, each page is divided into columns and the data flows from column to column (down the first column, then down the second column, etc.). Data in such a report is normally organized into blocks or details (for example, all the data about a given customer: customer number, company name, address, phone number, etc.). The details flow after one another in some organized fashion (numerically, by customer number; alphabetically by customer name, etc.).

To set up this kind of report:

- activate the database(s) you want to use,
- format the report for multiple column printing,
- set up your details, and
- print the report.

Activating the database(s) you want to use

For complete instructions on activating the database(s) you want to use in your report, see File|New|Custom Report.

Formatting the report for multiple column printing

You format a report for multiple column printing using the Format|Section command.

When you format the report for multiple column printing, you specify:

- the size of each detail (block of data),
- the size of the gaps that you want to appear (horizontally and vertically) between details, and
- the path you want the program to follow when printing your data.

Using Group Sort, Record Sort Order together

When you group data, Crystal Reports first sorts the data and then breaks it into groups. If you have set a sort order (using the Report|Record Sort Order command) in addition to the grouping sort, the grouping sort takes precedence. In other words, the grouping sort is done first; non-grouping sorts follow, even if they were entered first. There are times in which you may not want this to happen.

Consider the following data:

Customer	Item	Quantity
A	A1	4
A	A2	2
A	A2	3
A	A1	1
B	A2	4
B	A1	2
B	A1	1
B	A2	3

Assume that you want the data broken down by customer so that all Customer A data appears before Customer B data, plus, you want a quantity subtotal for each item. In other words, you want your data to look like this on the completed report:

Customer	Item	Quantity	Subtotal
A	A1	1	
A	A1	4	
			5
A	A2	3	
A	A2	2	
			5
B	A1	1	
B	A1	2	
			3
B	A2	3	
B	A2	4	
			7

Getting your data like this requires two separate sorts:

- the data must be sorted by customer, and
- the data for each customer must be sorted (and subtotaled) by item.

A logical approach to doing this sorting seems to be to use Report|Record Sort Order to set up *{file.Customer}* as a sort field, and then use Insert|Subtotal to subtotal *{file.Quantity}* every time *{file.Item}* changes. Unfortunately, this approach does not produce the desired results. When you use that approach, your data comes out looking like this:

Customer	Item	Quantity	Subtotal
A	A1	1	

A	A1	4	
B	A1	1	
B	A1	2	
			8
A	A2	3	
A	A2	2	
B	A2	3	
B	A2	4	
			12

Because the grouping sort (that is a part of the subtotaling activity) takes precedence over the customer sort (set up via Report|Record Sort Order), the sorts are performed in the wrong order to produce the results you want.

- Data is first sorted by item (the grouping sort), so that all Item A1 appears before item A2.
- Then the data is sorted by customer (the non
- grouping sort), so that, for item A1, all customer A comes before customer B, and for item A2, all customer A comes before customer B.

Clearly you need to sort the data by customer before it is sorted by item.

How to solve this problem

To solve the problem you group your data by customer (using Insert|Group Section), and then you subtotal your data so that the Quantity field is subtotaled every time the item changes. This time, since both sorts are part of grouping activity, the sorts take place in the order in which they are entered into Crystal Reports. (To eliminate extraneous spacing caused by the extra grouping, you can hide the customer group sections using Format|Section/Hide Section.)

1. Select Insert|Group Section. The Insert Group Section dialog box appears.
2. In the scroll box, select the field that you want to be used as the first sort field.
3. Set the Sort Direction to in ascending order (A to Z, 1 to 9) or in descending order(Z to A, 9 to 1) in the next scroll box, and set the Condition bottom scroll box if the field you select is a date field .
4. Click OK when finished.
5. Select the field that you want subtotaled.
6. Select Insert|Subtotal. The Insert Subtotal dialog box appears.
7. In the top scroll box, select the field that you want to trigger a subtotal whenever it's value changes.
8. Set the sort direction to in ascending order (A to Z, 1 to 9) or in descending order (Z to A, 9 to 1), and set the Condition if the field you select is a date field.
9. Click OK when finished.
10. If you want to hide the first group field, select Format|Section. The Format Section/Sections dialog box appears.
11. Select the group header section (Group header #1:) for the first group you entered, and Click OK when finished. The Format Section dialog box appears.
12. Click Hide Section and Click OK when finished.
13. Select Format|Section one more time, and select the group footer section (Group footer #1:) for the first group you entered and Click OK when finished. The Format Section dialog box appears.
14. Click Hide Section and Click OK when finished.

15. Now, when you print your report, the data will be sorted by the first sort field (the sort and group by field for the first group) and then by the second sort field (the sort and group by field for the subtotal group).

NOTE: The data is sorted using the first sort field even though you have hidden the first group. Hiding only keeps the data from printing; it doesn't prevent the sorting from taking place.

Cross-Tab Reports

NOTE: Crystal Reports has a Cross-Tab Expert that will lead you step by step through the report creation process. You can use the following procedure if you want to create Cross

-Tab reports using the File|New|Custom Cross
-Tab command.

Use File|New|Custom Cross-Tab to create a Cross

-Tab report. Cross
-Tabs are reports that present summarized data in a row versus column structure similar to that of a spreadsheet. These reports are easy to read, and they are particularly useful for making rapid comparisons or identifying trends. To illustrate, consider the following data representing only a portion of a fairly cumbersome columnar report.

CUSTOMER	ITEM	QUANTITY
A	1001	14
A	1002	43
A	1003	14
A	1101	13
A	1102	15
B	1001	48
B	1002	11
B	1003	20
B	1101	14
B	1102	13
C	1001	23
C	1002	4
C	1003	17
C	1101	1
D	1001	5
D	1002	8

The report shows the quantity of each item purchased by each customer. But its columnar format makes the report long and difficult to use when trying to compare customers' purchasing habits. Now consider the same data presented in a Cross-Tab report:

	1001	1002	1003	1101	1102	Total
A	14	43	14	13	15	**99**
B	48	11	20	14	13	**106**
C	23	4	17	1		**45**
D	5	8	2	6	19	**40**
E	39	25	8	4	7	**83**
F	18	8	1	14	10	**51**
G	18	12	8		3	**41**
H	11	13	3	3	2	**32**

I	4				4	**8**
J	10	5		6	1	**22**
K	19	6	4		11	**40**
L	24	5		2	9	**40**
Total	**233**	**140**	**77**	**63**	**94**	**607**

This report shows the quantity of each item that was bought by each customer:.

- customers (A, B,...L) make up the row headings,
- items (1001,1002..1102) make up the column headings,
- the quantity purchased is the summarized field.
- the value at each row/column intersection is the quantity of a particular item bought by a particular customer, for example, the total number of item 1102 purchased by Customer "F".
- the total at the end of each row is the total of all the quantity values in that row, for example, the total number of items purchased by customer "D" .
- the total at the bottom of each column is the total of all the quantity values in that column, for example, the total number of item 1001 purchased by all customers.

The report is compact, and you can compare your customers' purchasing habits in a hurry.

- Cross

▪Tab reports are powerful reports that can boost your efficiency, but they're easy to create using Crystal Reports.

NOTE: When you use the File|New|Custom Cross▪Tab command while using another report, Crystal Reports creates a new report window and opens the new report in that window. The report you were working on remains unchanged in its own window.

To build a Cross▪Tab report

The following instructions assume that you are building a Cross▪Tab report using a database file as your data source. If you are using a Dictionary, a SQL table, or a Report as your data source, the steps you need to follow to call up the Cross

▪Tab dialog box will be somewhat different.

1. Select File|New|Custom Cross▪Tab. The Choose Database File dialog box appears.
2. Select the database from which you want to create a Cross▪Tab report, and Click OK when finished. The Cross

▪Tab dialog box appears.

NOTE: If you have inactivated the Use Default Alias option via the File|Options command, the Alias Name dialog box will appear before the Cross▪Tab dialog box.

Dialog box overview

The Cross▪Tab dialog box makes it easy for you to set up sophisticated Cross

▪Tab reports in a hurry. To use this dialog box:

- you select the field(s) that you want to appear as column headings and place them in the Columns box,
- you select the field(s) that you want to appear as row headings and place them in the Rows box, and
- you select the field that you want summarized and place it in the Summarized Field box. (A summarized field is the field whose value you want to appear at the row/column intersections and to be totaled at the end of each row and at the bottom of each column.)

NOTE: In order to print a Cross▪Tab report, you must enter at least a summarized field in the Cross

▪Tab dialog box. That is the only field that is required.
NOTE: You can have only one summarized field per Cross▪Tab report.

NOTE: There is no practical limit on the number of row/column headings you can have in a Crystal Cross-Tab report. When you have selected more than one heading for a row or column, the heading at the top of the list becomes the outmost heading, the next heading on the list falls just inside the first heading, etc. For example, if your list in the Columns box has Item at the top and Color just below it, your report will display data broken down first by item, and then, for each item, the data will be broken down by color.

Selecting, placing, and removing fields

There are two ways you can select and place a field in the Cross-Tab dialog box

- Drag and Drop, and
- using the Add/Set buttons:

Drag and Drop

To drag and drop a field, simply highlight the field name in the list in the Fields box. Then, with the cursor on the highlighted field name, depress the left mouse button and, while keeping it depressed, drag the field where you want it to go. Release the mouse key to drop the field when you get it into position.

NOTE: When you drag and drop a field on top of another field, the new field will appear in the field list beneath the old one.

NOTE: You can also drag and drop fields between the Row, Column, and Summarized Field boxes or within any of those boxes. For example, if you have two fields in the Row box and you want to reverse the order in which they appear, drag the top field onto the bottom field and the fields will reverse positions.

NOTE: If you have a field in the Summarized Field box and you drag a new field into that box, the new field will replace the old field.

Add/Set buttons

These buttons send the selected field to the specified destination (Rows box, Columns box, etc.). To use, highlight the field name in the Fields box list and Click the appropriate button to send it to the destination you want . For example, to add a highlighted field to the Columns box, Click the Add Column to Cross-Tab button and the program will enter the highlighted field in the Columns box.

The Cross-Tab dialog box contains a number of other boxes and buttons:

- **Remove Field from Cross-Tab button** removes a highlighted field from the report. This command works with fields in the Rows, Columns, or Summarized Field boxes.
- **Options button** enables you to modify group options for any grouped field used in the Cross-Tab report. The Cross-Tab Group Options dialog box appears. Once in that dialog box you can select ascending or descending sort order, for grouped date fields you can select the date condition, and for grouped boolean fields you can select the boolean condition.
- **Browse Field Data button** displays the data in a selected field. To review data in any field, highlight the field and Click the Browse Field Data button. This option works in the Rows, Columns, and Summarized Field boxes, and in the Fields box as well.
- **Fields box** is similar to the Insert Database Field dialog box. It contains a list of all of the fields and formula fields that are available for use in your report. You select fields from the list and place them in either the Rows box, the Columns box, or the Summarized field box. As you place fields in your report, the program adds a list of those fields to the Fields box list under the heading *report fields*.
- **Columns box** contains a list of all the fields that you want to use as column headings. You add fields to the list from the Fields box. These headings will appear on the horizontal (x) axis of your report.
- **Rows box** contains a list of all the fields that you want to use as row headings. You add fields to the list from the fields box. These headings will appear on the vertical (y) axis of your report.
- **Summarized Field box** box contains the field that you want summarized in your report. For example,

in a report that shows the amount ordered by item by customer, the amount field is the summarized field. Values from that field will appear on the report at all the row/column intersections at which an amount is called for.

- **New Formula button** enables you to create a formula for use in your report. When you Click this button, the New Formula dialog box appears. Enter a name for your report in that dialog box and Click OK. The Formula Editor appears. Create your formula and Click the Accept button when you're finished. Your formula will appear in the Fields list under the Report Fields heading. At this point you can use it in your report just as you would use any field.

NOTE: If your formula does not include fields from the open database(s), (if it's a constant formula for example [1=1]), it will not appear in the Fields list and therefore won't be available for use in your report.

- **Edit Formula button** enables you to edit a formula that you're using or that's available for use in your Cross
Tab report. When you highlight a formula in the Fields list or the Columns, Rows, or Summarized Field boxes and then Click this button, the program opens the Formula Editor with the highlighted formula active. Make your modifications and Click the Accept button when finished.

NOTE: This button is active only when you have a formula highlighted.

1. From the Fields box, select the field you want summarized and drag it into the Summarized Field box.
2. Again from the Fields box, select whatever fields you want for row headings and drag them into the Row box.
3. Again from the Fields box, select whatever fields you want for Column headings and drag them into the Column box.
4. Click OK to exit the Cross
Tab dialog box and move to the Cross
Tab Designer. Once in the Designer, you can format fields, group your data, etc. to get the report the way you want it.
5. Print your report using one of the File|Print options.

***NOTE: To add or delete report fields, you must be in the Cross
Tab dialog box.***

NOTE: When you have multiple row headings or multiple column headings in a report, the program summarizes the data in the order in which the headings appear in their respective boxes. For example, if the top field in the Row box is State and the next field is City, the report will summarize by state and then within each state, by city.

Linking databases

If you want to use linked databases in your report, use the following procedure:

1. Select File|New|Custom Cross
Tab and then select your first database from the Choose Database File dialog box when it appears.
2. Select your summarized field from the Fields box, and select any row or column headings you want.

NOTE: At a minimum, you must enter a summarized field.

3. Click the OK button and the program closes the Cross
Tab dialog box and takes you to the Cross
Tab Designer.
4. Select Database|Add Database to Report, select your second database, and set up your link.
5. Repeat Step 4 as many times as are necessary.
6. To return to the Cross
Tab dialog box, Click the right mouse button while the mouse cursor is positioned in the gray area to the left of the Cross
Tab Designer. Select Cross

▪Tab Layout from the pop
▪up menu that appears.

7. Once you're back in the Cross▪Tab dialog box, finish creating your report using the linked databases. All the fields from all of the linked databases will now appear in the Fields box.

Returning to the Cross▪Tab dialog box

To return to the Cross▪Tab dialog box:

1. Click the right mouse button while the mouse cursor is positioned in the gray area to the left of the Cross▪Tab Designer.
2. Select Cross▪Tab Layout from the pop

▪up menu that appears.

NOTE: Alternately, you can select Format\|Section, select Cross▪Tab from the Format Section (Sections) dialog box when it appears, Click OK, and then Click the Cross

▪Tab Layout button from the Format Section (Formatting) dialog box when it appears.

Inverting your Cross▪Tab report

Inverting a Cross▪Tab report means to exchange the headings on the horizontal and vertical axes of the report. What were column headings become row headings, and vice

▪versa. There are many reasons you might want to do this, but the most common reason is to get the report to fit better on a page.

1. With your Cross▪Tab report set up, Click the right mouse button in the gray Cross

▪Tab area to the left of the Cross
▪Tab Designer edit box. A pop
▪up menu appears.

2. Select Invert Cross▪Tab and the program inverts the report for you.
3. Repeat the process if you want to return the report to its original format.

Using a formula as a field in your report

You can create a formula and use it as a row or column heading or as a summarized field in your report. Once created, you use the formula field just as you would use any other field.

1. Click the New Formula button. The New Formula dialog box appears.
2. Enter a name for your formula and Click OK when finished. The Formula Editor appears.
3. Enter your formula and Click the Accept button when finished. Your formula appears in the Fields box under the heading *Report Fields*. All formula names are preceded by an @ sign, so if you called your formula *Quota*, for example, it would appear in the fields box as @*Quota*.
4. Add the formula to the Rows, Columns, or Summarized Field box as you would any other field.

Grouping data in your Cross▪Tab report

You can group data in your Cross▪Tab report just as you do in any other report. You select a row or a column heading as your sort and group by field and the program breaks the report into a series of mini Cross

▪Tab reports, each one showing a single group.

To group data in your Cross▪Tab report.

1. In the Cross▪Tab Designer, select the row or column field you want to use as your sort and group by field
2. Select Insert\|Group Section. The Insert Group Section dialog box appears.

3. Select the sort and group by field you want to use and the sort direction and Click OK when finished. The program groups your CrossTab to your specifications.

Using date and Boolean fields as summarized fields

When you use date or Boolean fields as summarized fields, the program handles them differently than it does when you use them as row or column headings.

- When used as row or column headings, the program prints the actual field values. (In a date field, it prints the first or last date in the period specified.)
- When used as summarized field, however, the program summarizes the values and prints a count of values (or, if you specify, some other summary), not the actual field value.

Adjusting the number of Browse values

When you select a Browse button or the Browse Field Data command, the program stops gathering values after 5000 milliseconds or 100 records are gathered, whichever comes first. You can adjust these values if you wish by changing the current value for either of the following lines in the [Crystal Reports] section of the CRW.INI file (located in the Windows directory):

```
BrowseTimeout
MaxNBrowseValues
```

These two lines are written into the CRW.INI file when you install Crystal Reports 4.0. If you are using a version prior to 4.0, your CRW.INI file will not contain these lines so you will have to type them in yourself and then enter the values you want.

Running Totals with Multiple Groups

You want to display a total, summary or calculation of field(s) in your report.

To keep a running total of values you have to create and use variables. The variable should be initialized and reset in the group header section, used as a counter in the detail section and displayed as a result in the group footer section. You must create a formula for each of these separately.

```
      @CounterReset ■ (formula name)
      WhilePrintingRecords;
      NumberVar Counter;
      Counter := 0;
@CounterCalculation ■ (formula name)
WhilePrintingRecords;
NumberVar Counter;
Counter := Counter + {file.field};
@CounterDisplay■ (formula name)
WhilePrintingRecords;
NumberVar Counter;
Counter;
```

@CounterReset, will be placed in the group header section, @CounterCalculation, will be placed in the Detail section and @CounterDisplay will be placed in the group footer section. Note that you don't have to display these formulas, you can hide them individually and they will still work.

If you have multiple groups and want to get sub■totals of your displayed value and want to carry that value onto the next group, you have to create more variables and formulas. In the following example, the report is grouped by Route and Department. (Department is the Outer group [G1], and Route is the Inner group [G2]

■ There are many routes per department).

```
      @DeptCounterReset ■ (formula name)

■ [in the outer group header]
WhilePrintingRecords;
NumberVar DeptCounter;
DeptCounter := 0;
@RouteCounterReset ■ (formula name)

■ [in the inner group header]
WhilePrintingRecords;
NumberVar RouteCounter;
RouteCounter := 0;
@CounterCalculation ■ (formula name)

■ [in the detail section]
WhilePrintingRecords;
NumberVar RouteCounter;
RouteCounter := RouteCounter + {file.field};
      @RouteCounterDisplay ■ (formula name)

■ [in the inner group footer]
WhilePrintingRecords;
NumberVar DeptCounter;
NumberVar RouteCounter;
DeptCounter:=DeptCounter + RouteCounter;
RouteCounter;
@DeptCounterDisplay ■ (formula name)
```

- [in the outer group footer]

```
WhilePrintingRecords;
NumberVar DeptCounter;
DeptCounter;
```

@DeptCounterReset will be placed in the Department group header section, @RouteCounterReset will be placed in the Route group header section, @CounterCalculation, will be placed in the Detail section, @RouteCounterDisplay will be placed in the Route group footer section and @DeptCounterDisplay will be placed in the Department group footer section. Note that, again, you don't have to display these formulas, you can hide them individually and they will still work.

The Crystal ActiveX Control

Properties Methods Error Messages

NOTE: Subreports cannot be modified at runtime through the ActiveX control but they can be using direct calls to the Report Engine API. If you want to modify subreports at runtime, you will not be able to use the ActiveX control with your application; you will have to create a Custom Print Link using the Report Engine API.

Crystal ActiveX Control OverviewACTIVEX_Crystal_ActiveX_Control_Overview

Adding the ActiveX control to your project

Using the ActiveX control

Creating Bound Reports with the ActiveX controlACTIVEX_Creating_Bound_Reports_with_the_ActiveX_control

Upgrading VB applications to accept a new Crystal ActiveX controlACTIVEX_Upgrading_VB_applications_to_accept_a_new_Crystal_ActiveX_control

Error Messages

Indexes

Crystal ActiveX Control Properties IndexACTIVEX_Crystal_ActiveX_Control_Properties_Index

Crystal ActiveX Control Methods IndexACTIVEX_Crystal_ActiveX_Control_Methods_Index

ActiveX Control Properties

All of the properties for this control are listed in the following table. Properties marked with an asterisk(*) are standard Visual Basic control properties. See the *Visual Basic Language Reference* or Visual Basic Help for documentation of these properties.

*Index	*Left	*Name	*Top
Action	Connect	CopiesToPrinter	DataFiles
Destination	DetailCopies	DiscardSavedData	EMailCCList
EMailMessage	EMailSubject	EMailToList	EMailVIMBCCList
Formulas	GraphData	GraphOptions	GraphText
GraphType	GroupCondition	GroupSelectionFormula	GroupSortFields
LastErrorNumber	LastErrorString	LogOnInfo	MarginBottom
MarginLeft	MarginRight	MarginTop	Password
PrintDay	PrinterCollation	PrinterCopies	PrinterDriver
PrinterName	PrinterPort	PrinterStartPage	PrinterStopPage
PrintFileCharSepQuote	PrintFileCharSepSeparator	PrintFileName	PrintFileType
PrintFileUseRptDateFmt	PrintFileUseRptNumberFmt	PrintMonth	PrintYear
RecordsPrinted	RecordsRead	RecordsSelected	ReportDisplayPage
ReportFileName	ReportLatestPage	ReportSource	ReportStartPage
SectionFont	SectionFormat	SectionLineHeight	SectionMinHeight
SelectionFormula	SessionHandle	SortFields	SQLQuery
Status	StoredProcParam	UserName	WindowBorderStyle
WindowControlBox	WindowControls	WindowHeight	WindowLeft
WindowMaxButton	WindowMinButton	WindowParentHandle	WindowState
WindowTitle	WindowTop	WindowWidth	

Crystal ActiveX Control Overview

Properties Methods Error Messages

ActiveX is Microsofts new Internet technology designed to bring more powerful applications to desktops and networks. ActiveX moves beyond applications that produce static documents to a Windows environment that provides active controls, documents, and client applications that can operate and interact not only with each other, but also with network intranets and the global Internet.

ActiveX controls, formerly known as OLE controls, provide plug-in capabilities that let you add application components, and even entire applications, to your own development projects without writing a line of code. Crystal Reports supports the ActiveX concept with the Crystal ActiveX Control. Use the ActiveX Control to easily add all of the report processing power of Crystal Reports to your own Visual Basic 4.0, Visual C++, Borland C++, Delphi, and other applications.

NOTE: Your development tools may refer to an ActiveX Control by any of the following names: OLE Control, ActiveX Control, or Custom Control. As long as the term used refers to a control with an .OCX filename extension, it is synonymous with the term ActiveX Control used here.

See Also:

Adding the ActiveX control to your project

Using the ActiveX control

Adding the ActiveX control to your project

This section demonstrates how to add the Crystal ActiveX Control to an application project being designed in Visual Basic 4.0. If you wish to use the ActiveX Control in a different development environment, please refer to the documentation that came with your development tools for information on adding an ActiveX or OLE control to your project.

The Crystal ActiveX Control is installed in the \WINDOWS\SYSTEM directory when you install Crystal Reports. You add the ActiveX Control to your Visual Basic project using the ActiveX controls command on the Visual Basic Tools menu.

1. Open Visual Basic.
2. Open the project to which you want to add the ActiveX Control.
3. Choose the ActiveX controls command from the Tools menu. The ActiveX controls dialog box appears.
4. If Crystal Report Control appears in the Available Controls list, click the check box next to it, click OK, and skip to Step 8.
5. If Crystal Report Control does not appear in the Available Controls list, click Browse. The Add ActiveX control dialog box appears.

NOTE: Crystal Report Control is the name of the Crystal ActiveX Control when it is added to a development project. The term ActiveX Control refers to a type of control, while Crystal Report Control is the name of the ActiveX Control provided by Crystal Reports.

6. Use the controls in the Add ActiveX control dialog box to locate and select the CRYSTL32.OCX (32-bit) file. This file is installed in your \WINDOWS\SYSTEM directory by default. Once you locate and select the file, click Open.
7. Crystal Report Control will now appear in the Available Controls list box. Click the check box next to the name of the control, and click OK.
8. Visual Basic adds the Crystal ActiveX Control to your toolbox.
9. When you want to add the ActiveX Control to a form, double-click the tool and the program installs it on the active form.

NOTE: The ActiveX Control can be added to AUTOLOAD.MAK to automatically load the Control to your project. Refer to Visual Basic documentation for information on adding controls to AUTOLOAD.MAK.

NOTE: For instructions on how to add an ActiveX Control or OLE control to development applications other than Visual Basic, refer to the documentation that came with the development application you are using.

Using the ActiveX control

Once you have the ActiveX Control object on your form, you build the connection between your application and Crystal Reports by setting the object's properties at design time or changing properties at runtime. The ActiveX properties let you specify:

- the name of the report you want to print in response to an application event,
- the destination for that report (window, file, or printer),
- the number of copies you want to print (if your report is going to the printer),
- print file information (if your report is going to a file),
- print window sizing and positioning information (if your report is going to a window),
- selection formula information (if you want to limit the records in your report),
- sorting information, and
- other related properties.

Crystal ActiveX Control properties can be changed either at design time or at runtime. Note, however, some properties are available only at runtime. These properties do not appear at design time.

NOTE: For a complete description of each property in the Crystal ActiveX Control, refer to the Crystal Reports Developers online Help.

CHANGING PROPERTIES FOR THE ACTIVEX CONTROL

1. Click the ActiveX control on your form to select it.
2. Right-click and choose the Properties command from the shortcut menu that appears. The Crystal Report Control Properties dialog box appears.
3. Use the tabs and controls in this dialog box to change the ActiveX Control properties at design time.

NOTE: ActiveX Control properties also appear in the Visual Basic Properties box. Properties can be changed here much like they are changed for the Crystal ActiveX Control.

CHANGING PROPERTIES AT RUNTIME

You can set most of the ActiveX Control properties at runtime by adding simple entries to your procedure code. Runtime property settings replace settings you make via the Properties list at design time.

Use the Action property or the PrintReport method to actually process the report at runtime. The Action property and the PrintReport method can only be used at runtime, and are the only means by which a report can actually be generated by the ActiveX Control.

Creating Bound Reports with the ActiveX control (OCX)

In Crystal Reports, a Crystal ActiveX Control can be bound directly to a Visual Basic Data Control. Crystal Reports is also bundled with True DBGrid, a bound grid control by Apex Software Corporation, which enables users to generate reports using the grid layout. These Crystal ActiveX Control features offer the following benefits:

- Generating reports in Visual Basic programs is made even easier and no longer requires an existing .RPT file.
- A powerful feature of Visual Basic is users can perform ad
hoc queries by executing SQL statements in the RecordSource property of the Data Control. By directly binding a Crystal ActiveX Control to a Data Control, users can now create reports of dynaset data which are the results of such ad
hoc queries.
- Using True DBGrid as the report source (see ReportSource property, users can perform database queries, view the resulting data in a table format, and generate reports at the same time. Furthermore, users can perform simple report customization by configuring True DBGrid (hiding columns or adjusting column widths) at design time or during program execution).

The following sections describe the Crystal ActiveX Control features.

Bound Report Driver And Bound Report Files

When you use Crystal Reports to generate reports from database files of a particular file format (e.g., Paradox file format), you need to have the appropriate report driver (e.g., PDBPDX.DLL) to retrieve data from the databases. Similarly, when you generate reports by binding to a Visual Basic Data Control, a Bound Report Driver (PDBBND.DLL) is used to retrieve data from the Data Control. Make sure PDBBND.DLL is in your \WINDOWS\SYSTEM directory or search paths, along with other database drivers.

To generate reports from a specified database file, you first use Crystal Reports (CRW.EXE) to create a report file (.RPT) and customize the report layout. You can now generate reports directly by binding the Crystal ActiveX control to a TrueDBGrid Control without the need of an existing .RPT file. Bound reports generated this way have a default layout and have access to the field list of the Data Control only. To allow for report customization, you can create a Bound .RPT file during Visual Basic design time or runtime. You can then use Crystal Reports to customize the Bound .RPT files (just like you do with any other .RPT files) and even link the bound data to other database tables.

Crystal ActiveX Control Properties

Several properties are added to the Crystal ActiveX Control in order to support Bound reports. These new properties are described below.

- **Custom**

This property allows you to create Bound .RPT files at Visual Basic design time and is not available at runtime. After a Bound .RPT file is created, programmers can then use Crystal Reports to customize the report layout or even link the bound data to other database tables.

- **DataSource *(Data Control)***

This property can be read/write at design time only and is not available at runtime. This property is ignored if the ReportSource property is 0 (Report files). To generate Bound reports, set this property to the Data Control you want to retrieve data from. The Data Control must already be on the form before this property can be set.

- **BoundReportFooter *(Boolean)***

This property can be read/write both at design time and runtime. This property is ignored if the ReportSource property is 0 (Report files). Default is *False* and the Bound reports generated will not have page numbers printed. If set to True, page numbers will be printed at the bottom of a page.

- **BoundReportHeading *(string expression)***

This property can be read/write both at design time and runtime. This property is ignored if the ReportSource property is 0 (Report files). It specifies the report title at the beginning of a Bound report.

If blank, no report title will be printed.

- **ReportSource *(numeric expression)***

This property can be read/write both at design time and runtime. The allowed values are:

0 ■ Report files

1 ■ Bound TrueGrid Control

3 ■ All Data Control Fields

The default value is 0 ■ Report files, and the ReportFileName property must be assigned to an existing report path name (.RPT). This is equivalent to when the new Bound report features were not available and all reports were generated from existing .RPT files.

When set to 1 or 3, the ReportFileName property will be ignored and no .RPT file is needed. Reports will be created using data retrieved from Data Control. The reports generated directly from the Data Control are identical to the reports generated from the respective Bound .RPT files created using the (Custom) property described above.

The best way to learn about bound reports is to perform the steps below:

Creating a bound report using the Crystal ActiveX control

1. Add the following controls to your Visual Basic Form:

- Data Control
- Crystal ActiveX control
- Command Button

2. On the Data Control:

- Set the DatabaseName property to the name of the database being reported on.
- Set the RecordSource property (this can be a database table or a SQL query statement).

3. On the Crystal ActiveX control:

- Set the DataSource property to the Data Control (for example: Data1).
- Set the ReportSource to 3 - All Data Control Fields.

4. On the Command Button, add the following code for the Click event:

```
Private Sub Command1_Click()
CrystalReport1.Action = 1
End Sub
```

Run the application, click the command button, and the Crystal ActiveX control will retrieve data from the Data Control and create the report. The report will appear as a simple columnar report. There are no runtime properties to control any report formatting. However, this can be accomplished at design time by editing the report designed by the ActiveX control (a report template) in Crystal Reports.

Creating a formatted bound report

1. Add the Data control, ActiveX control, and a command button to your form.
2. On the Data control, set the DatabaseName property and the RecordSource property as you did in the previous example.
3. On the ActiveX control:

- Set the DataSource property to the Data Control, e.g: Data1.
- Set the ReportSource property to 3 - All Data Control Fields.
- Open the Custom property and select the Data-Bound Report Tab.
- Click on the Save Report As button and enter a name for the report.

4. Open the report template in Crystal Reports and apply any formatting that you want including spacing between columns, font size, colors, grouping, and totaling. Save the report template again when finished.
5. In your Visual Basic application, set the following properties for the ActiveX control:

- Set the ReportSource to 0 - Report File.
- Set the ReportFileName to the .RPT file that you saved (include the complete path of the file).

6. On the command button, add the following code to the Click event:

```
Private Sub Command1_Click()
CrystalReport1.Action = 1
End Sub
```

Now, the application will create the report at runtime with the formatting you have applied.

Creating a formatted bound report at runtime

The following steps describe an alternative method of creating formatted bound reports:

1. Create your Visual Basic application as in the first example above.
2. Set the ActiveX control to print to a Preview window, and run the application.
3. Click on the Export button in the Preview window, and export the report to a disk file in .RPT format.
4. Once the report has been exported, you can open it up in Crystal Reports.
5. Perform all formatting changes that you want and save the report.
6. Return to the Visual Basic application and stop it if it is still running.
7. On the ActiveX control:

- Set the ReportSource to 0 - Report File.
- Set the ReportFileName to the .RPT file that you created.

8. Run the Visual Basic application and you will be able to see your bound report with the formatting changes you've made.

Note: When creating formatted reports for use with the bound data control in Visual Basic, you will not be able to refresh the data from within Crystal Reports since the data does not exist outside of the Visual Basic application.

Note: If you plan on using a formatted bound report, you will not be able to modify anything in the SELECT statement of the data control. The report needs all these fields and will fail if they are not all there. The formatted report can not report on any new fields.

Note: When passing properties at runtime using bound reports (e.g. SortFields), the syntax is slightly different. For example, the following syntax would be used for the Formulas and SortFields properties in a normal report:

```
CrystalReport1.Formulas(0) = "COMMISSION= {TableName.FIELDNAME}"
CrystalReport1.SortFields(0) = "+{TableName.FIELDNAME}"
```

However, for a bound report, the following syntax would be used:

```
CrystalReport1.Formulas(0) = "COMMISSION= {Bound Control.FIELDNAME}"
CrystalReport1.SortFields(0) = "+{Bound Control.FIELDNAME}"
```

Upgrading VB applications to accept a new Crystal ActiveX control

if you have built a Visual Basic application using an older version of CRYSTL32.OCX (the Crystal ActiveX control), follow one of these two procedures to run the application using the new version of the ActiveX control installed with Crystal Reports :

Saving your form as text

When you save your project using the older version of the ActiveX control, select the Save As Text option in the Save File As dialog box for each form in your project that uses the Crystal ActiveX Control. Then, after replacing the older ActiveX control with the new one, the form should open properly with the new Crystal ActiveX Control installed.

Replacing the old ActiveX control

If you have not saved the form as text, you will get two error messages when you attempt to open a form using a newer version of the ActiveX control.

1. The first error message specifies "Invalid file format." Click OK to move on.
2. The second error message says that there is a loading error. Click Yes to continue. When the form opens, a red box appears on the form in place of the Crystal ActiveX control. Select the red box and delete it using the Del key.
3. Now, click the Crystal ActiveX control icon in the tool box to place the newer version of the ActiveX control on the form.
4. Save the form.
5. Repeat Steps 1■4 for each additional form that contains the Crystal ActiveX control. From this point on, the forms should work properly with the new version of the ActiveX control.

Crystal ActiveX Control Properties Index

Click on a letter below to go to the index topics beginning with that letter.

A

Action	Triggers the printing of the report.

B

BoundReportFooter	True or False property that specifies whether or not the Page Footer is to appear in the Default Bound Report.
BoundReportHeading	String property that allows the user to give a title to the report.

C

Connect	Logs on to a SQL server or a non-SQL database that is password-protected (such as a Paradox database).
CopiesToPrinter	Specifies the number of copies to be printed if you are printing to a printer.

D

DataFiles	Specifies the location of the database files or tables used in the report.
DataSource	Specifies which data control the Crystal ActiveX Control is to look at for the data.
Destination	Specifies the destination to which your report is to be printed (Window, Printer, or File).
DetailCopies	Specifies the number of copies of each record in the Details section that the program is to print.
DiscardSavedData	If data is saved with the specified report, setting this property to 1 (True) discards the data.

E

EMailCCList	Specifies the "CC" list to which you want your e-mail message sent.
EMailMessage	Specifies the message line in your e-mail message.
EMailSubject	Specifies the subject line in your e-mail message.
EMailToList	Specifies the "To" list to which you want your e-mail message directed.
EMailVIMBCCList	Specifies the "Blind CC" list to which you want your e-mail message copied.

F

Formulas	Specifies a new string for an existing formula.

G

GraphData	Sets the data used for a specified graph.
GraphOptions	Sets a number of options for the specified graph.
GraphText	Sets the various text components for the specified graph.
GraphType	Sets the kind of graph used in the selected section in the

	specified report.
GroupCondition	Specifies what kind of change in the Group Condition Field will trigger the creation of a group.
GroupSelectionFormula	Specifies the groups to be used when printing the report.
GroupSortFields	Specifies the group field(s) that are to be used to sort your data when the report is printed.
I	
Index	See Visual Basic documentation.
L	
LastErrorNumber	Returns the error code for the last runtime error.
LastErrorString	Returns the error string for the last runtime error.
Left	See Visual Basic documentation.
M	
MarginBottom	Sets the bottom margin for the specified report.
MarginLeft	Sets the left margin for the specified report.
MarginRight	Sets the right margin for the specified report.
MarginTop	Sets the top margin for the specified report.
N	
Name	See Visual Basic documentation.
P	
Password	Enters the password needed to use database tables on a restricted Access .mdb file.
PrintDay	Sets the day component of the print date (if different from the actual date the report is printed).
PrinterCollation	If you specify more than one copy to be printed (using the PrinterCopies property), PrinterCollation specifies whether or not the copies will be collated.
PrinterCopies	Sets the number of report copies to be printed.
PrinterDriver	Sets the name of the printer driver that is to print the report.
PrinterName	Sets the name of the printer that is to print the report.
PrinterPort	Sets the name of the printer port that is to print the report.
PrinterStartPage	Sets the first page to be printed.
PrinterStopPage	Sets the last page to be printed.
PrintFileCharSepQuote	Sets the quote character used to enclose alphanumeric field data when printing to a file using Character Separated format.
PrintFileCharSepSeparator	Sets the character(s) you want to use to separate the fields when printing to a file using the Character Separated Value format.
PrintFileName	Specifies the name of the file to which the report is to be printed.

PrintFileType	Specifies the type of the file to which the report is to be printed.
PrintFileUseRptDateFmt	When printing to a file, indicates whether or not the program should save dates in the same date format (MDY, DMY, etc.) that is used in the report or instead optimize the dates for the file format you have selected.
PrintFileUseRptNumberFmt	When printing to a file, indicates whether or not the program should print numbers in the same format (decimal places, negatives, etc.) that you have used in the report or instead optimize the numbers for the file format you have selected.
PrintMonth	Sets the month component of the print date (if different from the actual date the report is printed).
PrintYear	Sets the year component of the print date (if different from the actual date the report is printed).

R

RecordsPrinted	Determines the number of records actually printed.
RecordsRead	Determines the number of records actually read.
RecordsSelected	Determines the number of records selected.
ReportDisplayPage	Indicates which page of a multi-page report is currently being displayed in the Print window.
ReportFileName	Specifies the report to be printed.
ReportLatestPage	Determines the last page printed in the specified report.
ReportSource	Specifies the report source as a report file, a data control, or a True Grid data control.
ReportStartPage	Determines the first page printed in the specified report.

S

SectionFont	Specifies the font for a section.
SectionFormat	Specifies the format for a section.
SectionLineHeight	Specifies the line height for a section.
SectionMinHeight	Sets the minimum section height for the specified report section.
SelectionFormula	Specifies the records to be used when printing the report.
SessionHandle	Sets the session handle for a user once the UserName and Password properties have opened an Access .mdb file for use by the report.
SortFields	Specifies the field(s) that are to be used to sort your data when the report is printed.
SQLQuery	Sets the SQL query string used by the specified report.
Status	Determines the print status for the specified report.
StoredProcParam	Sets the stored procedure parameters when using a report based on SQL stored procedures.

T

Top	See Visual Basic documentation.
U	
UserName	Enters the name given to a user for logging on to a protected Access .MDB file to obtain data files needed by the report.
W	
WindowBorderStyle	Specifies the type of border for the print window.
WindowControlBox	Specifies whether or not the print window is to have a control (system menu) box in the upper left hand corner when the report is printed to a window.
WindowControls	Specifies whether or not the print controls are to appear in the Print window when printing a report to a window.
WindowHeight	Sets the height of the print window when the report is printed to a window.
WindowLeft	Sets the distance, in pixels, that the print window is to appear from the left edge of the parent window. If the print window is a top level window, the distance is measured from the left edge of the screen.
WindowMaxButton	Specifies whether or not the print window is to have a maximize button when the report is printed to a window.
WindowMinButton	Specifies whether or not the print window is to have a minimize button when the report is printed to a window.
WindowParentHandle	Specifies the handle of the parent window if the print window is to be the child of another window.
WindowState	Sets the state of the Print window, normal, minimized, or maximized, when the report is printed to a Print window.
WindowTitle	Specifies the title you want to appear in the print window title bar when the report is printed to a window.
WindowTop	Sets the distance, in pixels, that the print window is to appear from the top edge of the parent window. If the print window is a top level window, the distance is measured from the top edge of the screen.
WindowWidth	Specifies the width of the print window in pixels.

Crystal ActiveX Control Methods Index

FetchSelectionFormula	Retrieves the selection formula from the report.
PrintReport	Triggers the printing of the report.
ReplaceSelectionFormula	Replaces the existing selection formula.
RetrieveDatafiles	Retrieves "table" locations from the report.
RetrieveLogOnInfo	Retrieves logon info from the report.
RetrieveSQLQuery	Retrieves SQL Query from the report.
RetrieveStoredProcParams	Retrieves Stored Procedure parameter values from the report.

DetailCopies Property

Description

Specifies the number of copies of each record in the Details section that the program is to print.

Usage

[*form*.]*Report*.**DetailCopies**[= *NumCopies%*]

For example:

```
Report1.DetailCopies = 3
```

« Specifies that three (3) copies of each record in the details section are to be printed. »

Remarks

If DetailCopies is set to a value less than or equal to zero, the value is ignored and 1 copy of the detail section of the report is printed.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

DiscardSavedData Property

Description

If data is saved with the specified report, setting this property to 1 (True) discards the data.

Usage

[*form*.]*Report*.**DiscardSavedData**[= *TrueFalse%*]

For example:

```
Report1.DiscardSavedData = 1
```

« Discards the data saved with the specified report. »

Remarks

For *TrueFalse%* use one of the following: False = 0, True = 1.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

EMailCCList Property

Description

Specifies the "CC" list to which you want your E-mail message sent.

Usage

[*form*.]*Report*.**EMailCCList** [= *CCList$*]

For example:

```
Report1.EMailCCList = "John Brown;Jane Doe"
```

« Sends a CC of the E-mail message to both John Brown and Jane Doe »

Remarks

- Applies to both VIM and MAPI.
- Multiple names must be semicolon separated.

Data Type

String

Availability

Read and Write (Design time and Runtime)

EMailMessage Property

Description

Specifies the string you want to appear as the body of your E■mail message.

Usage

[*form*.]*Report*.**EMailMessage**[=*Message$*]

For example:

```
Report1.EMailMessage = "The meeting is at 4:00"
```

« Sets "The meeting is at 4:00" as the body of the E■mail message. »

Remarks

Applies to both MAPI and VIM.

Data Type

String.

Availability

Read and Write (Design time and Runtime)

EMailSubject Property

Description

Specifies the subject line in your E■mail message.

Usage

[*form*.]*Report*.**EMailSubject**[= *Subject$*]

For example:

```
Report1.EMailSubject = "Staff meeting"
```

« Sets "Staff meeting" as the subject line in an E■mail message. »

Remarks

Applies to both MAPI and VIM.

Data Type

String

Availability

Read and Write (Design time and Runtime)

EMailToList Property

Description

Specifies the "To" list to which you want your E-mail message directed.

Usage

[*form*.]*Report*.**EMailToList** [=*ToList$*]

For example:

```
Report1.EMailToList = "Jane Doe"
```

« Makes "Jane Doe" the only name in the "To" list. »

Remarks

- Applies to both MAPI and VIM
- Multiple names must be semicolon separated.

Data Type

String

Availability

Read and Write (Design time and Runtime)

EMailVIMBCCList Property

Description

Specifies the "Blind CC" list to which you want your E-mail message copied.

Usage

[*form*.]*Report*.**EMailVIMBCCList** [=*BCCList$*]

For example:

```
Report1.EMailVIMBCCList = "John Jacobs;Jane Doe"
```

« Makes "John Jacobs" and "Jane Doe" the names for the BCC list »

Remarks

- Applies to VIM only, not MAPI.
- Multiple names must be semicolon separated.

Data Type

String

Availability

Read and Write (Design time and Runtime)

GraphData Property

Description

Gets and sets the data used for a specified graph.

Usage

[*form*.]*Report*.**GraphData**(*ArrayIndex%*)[= *sectionCode;graphNum;row;col;field;direction$*]

For example:

```
Report1.GraphData(0)= "DETAIL; 1; GROUP1; GROUP2; 0; COLANDROW"
```

« The value in Group 1 is used for the rows of the graph, the value in Group 2 is used for the columns of the graph, the first summarized field added to the report is used to set the value of the risers of the graph, and values in both columns and rows are used to create the graph.»

Remarks

1. With GraphData, you can specify changes to one or more graphs at runtime. Those changes then take place sequentially when you make the "Action=1" call. The array index value for GraphData simply specifies the sequence number for the change. Thus:

   ```
   Report1.GraphData(0) = "DETAIL; 3; Group1; Group2; 666; COLANDROW"
   ```

   when making changes to one graph only, but

   ```
   Report1.GraphData(0) = "HEADER; 3; Group1; Group2; 666; COLANDROW"
   Report1.GraphData(1) = "DETAIL; 3; Group1; Group2; 666; COLANDROW"
   ```

   when making changes to more than one graph.

2. Use the following table as a guide in supplying the required values for this property:

Parameter		**Expected value**
sectionCode	Specifies the section in which you want to modify a graph.	Please refer to the section code table.
graphNum	The number of the graph within the section you want to modify.	Graphs in a section are numbered, starting with zero, left to right first, then top to bottom.
row	The Group number in the report used to create rows in the graph.	GROUP1, GROUP2, GROUP3, . . ., GROUP9
col	The Group number in the report used to create columns in the graph.	GROUP1, GROUP2, GROUP3, . . ., GROUP9
field	The summarization field containing values to be used as the value of each riser in the graph.	The first summarization field added to a report is numbered 0, the second added to a report is numbered 1, etc.
direction	Whether the values in the rows, the columns,	ROWS, COLS, ROWANDCOL, or COLANDROW

or both are used to create the graph.

Data Type

Array of strings

Availability

Read and Write|Runtime

GraphOptions Property

Description

Gets and sets a number of options for the specified graph.

Usage

[*form.*]*Report.***GraphOptions**(*ArrayIndex%*)[= *sectionCode; graphNum; fontFace; barDirection;labelRisers;gridLines;legend;max;min$*]

For example:

```
Report1.GraphOptions(0) = "FOOTER;0;Arial;H;T;F;X;max;min"
```

« Sets the font to Arial, sets horizontal bars, shows a data value on every riser (labelRisers = T), and toggles the grid lines off in the first Graph in the Page Footer section. »

Remarks

1. With GraphOptions, you can specify changes to one or more graphs at runtime. Those changes then take place sequentially when you make the "Action=1" call. The array index value for GraphOptions simply specifies the sequence number for the change. Thus:

   ```
   Report1.GraphOptions(0) = "DETAIL; 1; Arial; H; T; F; legend; max; min"
   ```

   when making changes to one graph only, but

   ```
   Report1.GraphOptions(0) = " 1; title string; subtitle string; footnote
   string; series string; group string; x string;y string; z string"

   Report1.GraphOptions(1) = " 1; title string; subtitle string; footnote
   string; series string; group string; x string;y string; z string"
   ```

   when making changes to more than one graph.

2. Use the following chart as a guide in entering the required property values:

Parameter		**Values expected**
sectionCode	Specifies the section in which you want to modify a graph.	Please refer to the section code table.
graphNum	Specifies which graph in the section you want to modify.	Graphs in a section are numbered, starting with zero, left to right first, then top to bottom.
fontFace	Specifies the font face you want to use for the entire graph.	Actual name of font, e.g., Arial
barDirection	In a bar graph, specifies the direction in which you want the graph bars to appear.	H = horizontal, V = vertical, X = as is
labelRisers	Specifies whether or not you want to show the data value on every riser.	T= true, F = False, X = as is
gridLines	Specifies whether or not you want to show grid lines.	T= true, F = False, X = as is
legend	Specifies whether or not you want to show a legend.	T= true, F = False, X = as is
max	Specifies the maximum value you want included in your graph.	Enter a number.

min	Specifies the minimum value you want included in your graph.	Enter a number.

Data Type

Array of strings

Availability

Read and Write|Runtime

GraphText Property

Description

Gets and sets the various text components for the specified graph.

Usage

[*form.*]*Report.***GraphText**(ArrayIndex%)[= *sectionCode; graphNum;title; subTitle; footnote;series;group;x;y;z$*]

For example:

```
Report1.GraphText(0) = "HEADER; 0;;;;;;new x label;
new y label;new z label"
```

« Resets the x, y, and z labels for the first graph in the Page Header section. »

Remarks

1. Select your section code from the section code table.
2. With GraphText, you can specify changes to one or more graphs at runtime. Those changes then take place sequentially when you make the "Action=1" call. The array index value for GraphText simply specifies the sequence number for the change. Thus:

   ```
   Report1.GraphText(0) = "DETAIL; 1; title string; subtitle string;
   footnote string; series string; group string; x string;y string; z
   string"
   ```

   when making changes to one graph only, but

   ```
   Report1.GraphText(0) = " 1; title string; subtitle string; footnote
   string; series string; group string; x string;y string; z string"

   Report1.GraphText(1) = " 1; title string; subtitle string; footnote
   string; series string; group string; x string;y string; z string"
   ```

   when making changes to more than one graph.
3. "title", "subTitle", "footnote", "series", "group", "x","y", and "z" are the strings you want to label the appropriate parts of the graph.

Data Type

Array of strings

Availability

Read and Write|Runtime

GraphType Property

Description

Gets and sets the kind of graph used in the selected section in the specified report.

Usage

[*form.*]*Report.***GraphType**(*ArrayIndex%*)[= *sectionCode;graphNum;graphType$*]

For example:

```
Report1.GraphType(0)= "GH1; 0; PIE"
```

« Specifies a Pie graph as the first graph (graphNum =0) in the Group Header 1 section. »

Remarks

1. With GraphType, you can specify changes to one or more graphs at runtime. Those changes then take place sequentially when you make the "Action=1" call. The array index value for GraphType specifies the sequence number for the change. Thus:

   ```
   Report1.GraphType(0) = "DETAIL; 0; PIE"
   ```

   when making changes to one graph only, but

   ```
   Report1.GraphType(0) = "HEADER; 0; PIE"
   Report1.GraphType(1) = "DETAIL; 0; PIE"
   ```

   when making changes to more than one graph.
2. Select sectionCode from the section code table.
3. Graph numbers are 0 origin; the first graph in a section is number 0, the second is number 1, etc.
4. Multiple graphs in a section are numbered left to right first, then top to bottom.
5. Select from the following graph types for the graphType value for this property:

For this type of graph	Use this code for graphType
Side■by■side	SIDEBYSIDE
3■D side■by■side	3DSIDE
Stacked bar	STACKEDBAR
3■D stacked bar	3DSTACKED
Percent bar	PERCENTBAR
3■D percent bar	3DPERCENT
Line	LINE
Area	AREA
3■D bars	3DBARS
Pie	PIE
Multiple pie	MULTIPLEPIE
Weighted pie	WEIGHTEDPIE

Data Type

Arrays of strings

Availability

Read and Write|Runtime

GroupCondition Property

Description

Specifies what kind of change in the Group Condition Field will trigger the creation of a group.

Usage

[*form*.]*Report*.**GroupCondition**(*SequentialIndex%*)[= *group;field;condition;sortDirection$*]

For example:

```
Report1.GroupCondition(0) = "GROUP1;{header.ordernum};ANYCHANGE;A"
```

« Specifies that any change in the *ordernum* field in Group1 will trigger a new grouping.»

Remarks

Refer to the following tables for parameter values for this property:

Parameter		Values expected
group	The group in which you want to set the group condition.	The outermost group on the report is GROUP1, the next group is GROUP2, etc.
field	The name of the field that triggers a grouping whenever its value changes.	Enter the name in the following format: {table.fieldname}
condition	Enter the condition that triggers the grouping.	See the tables below.
sortDirection	The direction in which groups are to be sorted.	A = Ascending, D = Descending

Date Field Conditions

Condition	Condition Code
Daily	DAILY
Weekly	WEEKLY
Bi-weekly	BIWEEKLY
Semi-monthly	SEMIMONTHLY
Monthly	MONTHLY
Quarterly	QUARTERLY
Semi-annually	SEMIANNUALLY
Annually	ANNUALLY

Boolean Field Conditions

Condition	Condition Code

To Yes	TOYES
To No	TONO
Every Yes	EVERYYES
Every No	EVERYNO
Next Is Yes	NEXTISYES
Next Is No	NEXTISNO

Condition for all other data types

Any Change	ANYCHANGE

Data Type

Array of strings

Availability

Read and Write|Runtime only

MarginBottom Property

Description

Gets and sets the bottom margin for the specified report.

Usage

[*form*.]*Report*.**MarginBottom**[=*MarginSetting%*]

For example:

```
Report1.MarginBottom = 720
```

« Sets a 1/2 inch bottom margin for the report (1 inch = 1440 twips) »

Remarks

MarginSetting% is the margin you want, in twips.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

MarginLeft Property

Description

Gets and sets the left margin for the specified report.

Usage

[*form*.]*Report*.**MarginLeft**[=*MarginSetting%*]

For example:

```
Report1.MarginLeft = 1440
```

« Sets a 1 inch left margin for the report (1 inch = 1440 twips) »

Remarks

MarginSetting% is the margin you want, in twips.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

MarginRight Property

Description

Gets and sets the right margin for the specified report.

Usage

[*form*.]*Report*.**MarginRight**[=*MarginSetting%*]

For example:

```
Report1.MarginRight=1440
```

« Sets a 1 inch right margin for the report (1 inch = 1440 twips) »

Remarks

MarginSetting% is the margin you want, in twips.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

MarginTop Property

Description

Gets and sets the top margin for the specified report.

Usage

[*form*.]*Report*.**MarginTop**[=*MarginSetting%*]

For example:

```
Report1.MarginTop = 720
```

« Sets a 1/2 inch top margin for the report (1 inch = 1440 twips) »

Remarks

MarginSetting% is the margin you want, in twips.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

PrinterCollation Property

Description

If you specify more than one copy to be printed (using the PrinterCopies property), PrinterCollation specifies whether or not the copies will be collated.

Usage

[*form*.]*Report*.**PrinterCollation**[=*CollationCode*%]

For example:

```
Report1.PrinterCollation = 1
```

« Collates the copies of the specified report »

Remarks

Select your *CollationCode%* value from the following table:

Status	Code
Uncollated	0
Collated	1
Default Collation	2

Data Type

Integer (Enumerated)

Availability

Read and Write (Design time and Runtime)

PrinterDriver Property

Description

Gets and sets the name of the printer driver that is to print the report.

Usage

[*form*.]*Report*.**PrinterDriver**[= *DriverName$*]

For example:

```
Report1.PrinterDriver = "Epson24.drv"
```

« Sets the printer driver to be the Epson 24 pin driver »

Remarks

The PrinterDriver, PrinterName, and PrinterPort properties work together to define the printer that the report is to be sent to. All three properties must be set in order to define a new printer. If all three properties are not set, the printer defined in the report will be used. This may be the users default printer if none has been specified in the report.

Data Type

String.

Availability

Read and Write (DesignTime and Runtime)

PrinterName Property

Description

Gets and sets the name of the printer that is to print the report.

Usage

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

For example:

```
Report1.PrinterName= "Epson LQ■850"
```

«Specifies the Epson LQ■850 printer»

Remarks

The PrinterDriver, PrinterName, and PrinterPort properties work together to define the printer that the report is to be sent to. All three properties must be set in order to define a new printer. If all three properties are not set, the printer defined in the report will be used. This may be the users default printer if none has been specified in the report.

Data Type

String.

Availability

Read and Write (Design time and Runtime)

PrinterPort Property

Description

Gets and sets the port for the specified printer.

Usage

[*form*.]*Report*.**PrinterPort**[= *PortName$*]

For example:

```
Report1.PrinterPort= "LPT1"
```

« Sets the printer port to LPT1 »

Remarks

The PrinterDriver, PrinterName, and PrinterPort properties work together to define the printer that the report is to be sent to. All three properties must be set in order to define a new printer. If all three properties are not set, the printer defined in the report will be used. This may be the users default printer if none has been specified in the report.

Data Type

String.

Availability

Read and Write (Design time and Runtime)

PrinterStartPage Property

Description

Gets and sets the first page to be printed.

Usage

[*form*.]*Report*.**PrinterStartPage**[= *StartPage%*]

For example:

```
Report1.PrinterStartPage = 7
```

« Specifies that printing is to begin with Page 7 of the report. »

Remarks

If a value less than or equal to 0 is used for PrinterStartPage, the value is ignored and printing starts with Page 1.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

PrinterStopPage Property

Description

Gets and sets the last page to be printed.

Usage

[*form*.]*Report*.**PrinterStopPage**[=*StopPage%*]

For example:

```
Report1.PrinterStopPage = 12
```

« Specifies that the printing is to end with Page 12 of the report. »

Remarks

Use a value of -1 for PrinterStopPage to indicate that printing is to continue through to the last page.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

PrintFileCharSepQuote Property

Description

Gets and sets the quote character used to enclose alphanumeric field data when printing to a file using Character Separated format.

Usage

[*form*.]*Report*.**PrintFileCharSepQuote**[=*Quote$*]

For example:

```
Report1.PrintFileCharSepQuote = "   "
```

« Uses the quotation character to surround values saved in character separated format. »

Remarks

- Applies only when PrintFileType is 5
- Character
- separated values.
- Applies only when Destination is 2
- File, 3
- EMail to MAPI, or 4
- EMail to VIM.
- If you assign a string to PrintFileCharSepQuote that is longer than one character, only the first character of that string is used. For example, if you assign "quote" to the property, only "q" will be recognized.

Data Type

String

Availability

Read and Write (Design time and Runtime)

PrintFileCharSepSeparator Property

Description

Gets and sets the character(s) you want to use to separate the fields when printing to a file using the Character Separated Value format.

Usage

[*form*.]*Report*.**PrintFileCharSepSeparator**[=*Separator$*]

For example:

```
Report1.PrintFileCharSepSeparator= "@"
```

« Specifies that the "@" character is to be used for separating field values. »

Remarks

1. Applies only when PrintFileType is 5 ■ Character ■separated values.
2. Applies only when Destination is 2 ■ File, 3 ■ EMail to MAPI, or 4 ■ EMail to VIM.

Data Type

String.

Availability

Read and Write (Design time and Runtime)

PrintFileUseRptDateFmt Property

Description

When printing to a file, indicates whether or not the program should save dates in the same date format (MDY, DMY, etc.) that is used in the report or instead optimize the dates for the file format you have selected.

Usage

[*form*.]*Report*.**PrintFileUseRptDateFmt**[= *TrueFalse%*]

For example:

```
Report1.PrintFileUseRptDateFmt = 1
```

« Specifies that the program should print dates in the same format as used in the report. »

Remarks

1. Applies only when PrintFileType is 0 ■ Record, 1
■ Tab
■separated, 3
■ DataInterchangeFormat (DIF), 4
■ CSV, or 5
■ Character Separated.
2. Applies only when Destination is 2 ■ File, 3
■ EMail to MAPI, or 4
■ EMail to VIM.
3. For *TrueFalse%*, use one of the following values: False = 0, True = 1.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

PrintFileUseRptNumberFmt Property

Description

When printing to a file, indicates whether or not the program should print numbers in the same format (decimal places, negatives, etc.) that you have used in the report or instead optimize the numbers for the file format you have selected.

Usage

[*form*.]*Report*.**PrintFileUseRptNumberFmt**[=*TrueFalse%*]

For example:

```
Report1.PrintFileUseRptNumberFmt = 1
```

« Specifies that the program should print numbers in the same format as used in the report. »

Remarks

1. Applies only when PrintFileType is 0 ■ Record, 1
■ Tab
■separated, 3
■ DataInterchangeFormat (DIF), 4
■ CSV, or 5
■ Character Separated
2. Applies only when Destination is 2 ■ File, 3
■ EMail to MAPI, or 4
■ EMail to VIM.
3. For *TrueFalse%*, use one of the following values: False = 0, True = 1.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

PrintMonth Property

Description

Gets and sets the month component of the print date (if different from the actual date the report is printed).

Usage

[*form*.]*Report*.**PrintMonth**[= *Month%*

For example:

```
Report1.PrintMonth= 7
```

« Sets July as the print month. »

Remarks

- Enter a value from 1
- 12 with January = 1, December = 12.
- The PrintYear, PrintMonth, and PrintDay properties function together. You must change the value of all three to change the print date. If you do not change all three, the print date saved with the report is used. This may be the current date if a specific date is not saved with the report.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

PrintYear Property

Description

Gets and sets the year component of the print date (if different from the actual date the report is printed).

Usage

[*form*.]*Report*.**PrintYear**[=*Year%*]

For example:

```
Report1.PrintYear = 1994
```

« Sets the year component of the print date to 1994. »

Remarks

- Enter the print year as a four
- digit number.
- The PrintYear, PrintMonth, and PrintDay properties function together. You must change the value of all three to change the print date. If you do not change all three, the print date saved with the report is used. This may be the current date if a specific date is not saved with the report.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

RecordsPrinted Property

Description

Determines the number of records actually printed.

Usage

[*form*.]*Report*.**RecordsPrinted**

For example:

```
Printed& = Report1.RecordsPrinted
```

« Fetches the number of records printed and stores it in the Printed variable. »

Remarks

If the report being printed contains one or more group selection formulas, the value returned by RecordsPrinted maybe significantly less than the value returned by RecordsSelected. Otherwise, this value should equal RecordsSelected.

Data Type

Long

Availability

Read■only|Runtime

RecordsRead Property

Description

Determines the number of records actually processed.

Usage

[*form*.]*Report*.**RecordsRead**

For example:

```
Read% = Report1.RecordsRead
```

« Fetches the number of records read and saves it in the Read variable. »

Data Type

Long

Availability

Read■only|Runtime

RecordsSelected Property

Description

Determines the number of records selected for inclusion in the report out of the total number of records read.

Usage

[*form*.]*Report*.**RecordsSelected**

For example:

```
Selected& = Report1.RecordsSelected
```

« Fetches the number of records selected and saves it in the Selected variable. »

Remarks

RecordsSelected will return a value anywhere between zero and the value returned by RecordsRead. The value returned by RecordsSelected depends on the queries and selection formulas set up in the report.

Data Type

Long

Availability

Read■only|Runtime

ReportDisplayPage Property

Description

Indicates which page of a multi-page report is currently being displayed in the Preview window.

Usage

[*form*.]*Report*.**ReportDisplayPage**

For example:

```
Result% = Report1.DisplayPage
```

« Fetches the number of the displayed page and stores it in the Result variable. »

Data Type

Integer

Availability

Read-only|Runtime

ReportLatestPage Property

Description

Determines the last page printed in the specified report.

Usage

[*form*.]*Report*.**ReportLatestPage**

For example:

```
Latest% = Report1.LatestPage
```

« Fetches the number of the last page printed and stores it in the Latest variable. »

Data Type

Integer

Availability

Read■only|Runtime

ReportStartPage Property

Description

Determines the first page printed in the specified report.

Usage

[*form*.]*Report*.**ReportStartPage**

For example:

```
StartPage% = Report1.ReportStartPage
```

« Fetches the number of the first page printed and stores it in the StartPage variable. »

Data Type

Integer

Availability

Read-only|Runtime only

ReportTitle Property

Description

Sets a report title and comments that supplement the DOS report name at design time or run time. Setting this property will let you set the Report Title, but not the comments.

Usage

[*form*.]*Report*.**ReportTitle**[= *Title$*]

For example:

```
Report1.ReportTitle = "Monthly Sales Report"
```

« Sets the report title to be Monthly Sales Report »

Remarks

The value of the ReportTitle is not displayed in any part of the report, but is provided for your own use.

Data Type

String

Availability

Read and Write (Design time and Runtime)

SectionFont Property

Description

Sets the font for one or more sections in the specified report.

Usage

[*form*.]*Report*.**SectionFont**(*SequentialIndex%*)[= *sectionCode; fontName; size; italic; bold;underline;strikethru$*]

For example:

```
Report1.SectionFont(0)="Footer;Arial;12;N;N;N;Y"
```

« Sets the font for the footer section to 12 point, Arial, strikethru. »

Remarks

With SectionFont, you can specify changes to one or more sections at runtime. Those changes then take place sequentially when you make the "Action=1" call. The array index value for SectionFont simply specifies the sequence number for the change. Thus:

```
Report1.SectionFont(0)= "DETAIL;Arial;12;N;N;N;Y"
```

when making changes to the DETAIL section only, but

```
Report1.SectionMinHeight(0) = "HEADER;Arial;12;N;N;N;Y"
Report1.SectionMinHeight(1) = "DETAIL;Arial;12;N;N;N;Y"
```

when making changes to more than one section.

Use the following table as a guide in supplying the required values for this property:

Parameter	Data type	Value expected
sectionCode	string	Please refer to the section code table.
fontName	string	The actual font name, e.g. Arial or Helvetica
size	number	The size of the font in points, e.g., 12 or 16
italic	character	T = true, F = False, X = as is1
bold	character	T = true, F = False, X = as is1
underline	character	T = true, F = False, X = as is1
strikethru	character	T = true, F = False, X = as is1
		1X (as is) uses the value saved with the report

Data Type

Array of strings

Availability

Read and Write|Runtime

SectionFormat Property

Description

Sets the format for one or more sections in the specified report.

Usage

[*form.*]*Report*.**SectionFormat**(SectionArrayIndex%)[= *sectionCode; visible; newPageBefore; newPageAfter;keepTogether;suppressBlank;resetPageNAfter;printAtPageBottom$*]

For example:

```
Report1.SectionFormat(0)= "GH2;F;X;X;X;X;X;X"
```

« Hides the Group Header 2 section (visible = F) while maintaining default settings for all other switches »

Remarks

With SectionFormat, you can specify changes to one or more sections at runtime. Those changes then take place sequentially when you make the "Action=1" call. The sequential index value for SectionFormat simply specifies the sequence number for the change. Thus:

```
Report1.SectionFormat(0)= "DETAIL;T;F;F;X;X;X;X"
```

when making changes to the DETAIL section only, but

```
Report1.SectionFormat(0) = "HEADER;T;F;F;X;X;X;X"
Report1.SectionFormat(1) = "DETAIL;T;F;F;X;X;X;X"
```

when making changes to more than one section.

Use the following table as a reference when entering parameter values for this property:

Parameter	Expected value
sectionCode	Please refer to the section code table.
visible	T = true, F = False, X = as is1
newPageBefore	T = true, F = False, X = as is1
newPageAfter	T = true, F = False, X = as is1
keepTogether	T = true, F = False, X = as is1
suppressBlank	T = true, F = False, X = as is1
resetPageNAfter	T = true, F = False, X = as is1
printAtPageBottom	T = true, F = False, X = as is1
	1X (as is) uses the settings saved with the report

Data Type

Arrays of strings

Availability

Read and Write|Runtime only

SectionLineHeight Property

Description

Specifies the line height in twips. A twip is 1/1440 inch; there are 20 twips in a point.

Usage

[*form*.]*Report*.**SectionLineHeight**(SequentialIndex%)[= *sectionCode;line;height;ascent$*]

For example:

```
Report1.SectionLineHeight(0) = "GH0; 1; 500; 300"
```

« Sets the line height for the second line in the group header zero section to a height of 500 twips with an ascent of 300 twips »

Remarks

1. With SectionLineHeight, you can specify changes to one or more sections at runtime. Those changes then take place sequentially when you make the "Action=1" call. The sequential index value for SectionLineHeight simply specifies the sequence number for the change. Thus:

   ```
   Report1.SectionLineHeight(0)= "DETAIL;1;500;300"
   ```

   when making changes to the DETAIL section only, but

   ```
   Report1.SectionLineHeight(0) = "HEADER;1;500;300"
   Report1.SectionLineHeight(1) = "DETAIL;1;500;300"
   ```

   when making changes to more than one section.

2. Use the following table as a guide in supplying the required values for this property:

sectionCode	Specifies the section code for the report section(s) from which you want to set a new line height.
lineN	Specifies the line(s) for which you want to set the line height. Line numbers in a section are 0 origin: the first line number is 0, the second is 1, etc.
height	Specifies the line height in twips. A twip is 1/1440 inch; there are 20 twips in a point.
ascent	Specifies the ascent in twips. Ascent is the distance from the top of the allotted line space (line height) to the baseline of the font. The ascent parameter is used to specify the position of the baseline if you specify an oversized or undersized line height. If you set ascent to 0, the program puts the baseline at the top of the space; if you set ascent to the same value as height, the program sets the baseline at the bottom of the space. For any other baseline, specify the ascent in twips.

Data Type

Arrays of strings

Availability

Read and Write|Runtime only

SectionMinHeight Property

Description

Sets the minimum section height for the specified report section.

Usage

[*form*.]*Report*.**SectionMinHeight**(*SequentialIndex%*)[=*sectionCode;minHeight$*]

For example:

```
Report1.SectionMinHeight(0) = "ALL; 500"
```

« Sets the minimum height for all sections to 500 twips. »

Remarks

With SectionMinHeight, you can specify changes to one or more sections at runtime. Those changes then take place sequentially when you make the "Action=1" call. The array index value for SectionMinHeight simply specifies the sequence number for the change. Thus:

```
Report1.SectionMinHeight(0)= "DETAIL;500"
```

when making changes to the DETAIL section only, but

```
Report1.SectionMinHeight(0) = "HEADER;500"
Report1.SectionMinHeight(1) = "DETAIL;500"
```

when making changes to more than one section.

Data Type

Array of strings

Availability

Read and Write|Runtime

SQLQuery Property

Description

Gets and sets the SQL query string used by the specified report.

Usage

[*form*.]*Report*.**SQLQuery**[=*SQLQuery$*]

For example:

```
Report1.SQLQuery = "SELECT authors.au_id, authors.au_lname,
authors.au_fname FROM pubs2.dbo.authors authors WHERE authors.au_lname >
Madison"
```

« Queries the SQL database to return only the records where the authors last name falls after Madison alphabetically. »

Remarks

You may only change the WHERE and FROM sections of an SQL query. Although the property requires that you enter the entire SQL query, the SELECT section must not be different from the original query.

Data Type

String

Availability

Read and Write (Design time and Runtime)

Status Property

Description

Determines the print status for the specified report.

Usage

[*form*.]*Report*.**Status**

For example:

```
Status% = Report1.Status
```

« Fetches the print status and saves it to the Status variable. »

Remarks

The Status property will return one of the following values:

0 ■ The report has not been printed or has not finished printing.
3 ■ The report has finished printing.
5 ■ The report has been cancelled by the user.

Data Type

Integer

Availability

Read■only|Runtime

StoredProcParam Property

Description

Gets and sets the stored procedure parameters when using a report based on SQL stored procedures.

Usage

[*form*.]*Report*.**StoredProcParam**(Parameter Array Index%)[= *newParameter$*]

For example:

```
Report1.StoredProcParam(0) = "06/14/1989"
```

« Sets the first stored procedure parameter to the date June 14, 1989 »

Remarks

StoredProcParam sets the value of the specified parameter in a SQL database table that is based on a stored procedure. Pass the value you wish to set the parameter to as a string. If the parameter expects a different data type, you still must pass the value as a string. For example, to pass the integer value 396, use the string "396".

Data Type

Arrays of strings

Availability

Read and Write|Runtime only

WindowControls Property

Description

Specifies whether or not the print controls are to appear in the Preview window when printing a report to a window.

Usage

[*form*.]*Report*.**WindowControls**[=*TrueFalse%]*

For example:

```
Report1.WindowControls = 1
```

« Specifies that print controls are to appear in the Preview window. »

Remarks

For *TrueFalse%* use one of the following: False = 0, True = 1.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

WindowState Property

Description

Sets the state of the Preview window, normal, minimized, or maximized, when the report is printed to a Preview window.

Usage

[*form*.]*Report*.**WindowState**[= *State%*]

For example:

```
Report1.WindowState= 2
```

« When the report is printed to a Preview window, the Preview window appears maximized when opened. »

Remarks

Use the following values to set the WindowState property:

0 = Normal

The Preview window appears neither minimized nor maximized. It appears in a default size and position previously defined by your application or by Windows.

1 = Minimized

The Preview window appears minimized as an icon close to the lower left hand corner of the screen. The icon can be restored to display the window in a normal state.

2 = Maximized

The Preview window is maximized when opened to fill the entire screen.

Data Type

Integer

Availability

Design time; Runtime

PrintDay Property

Description

Gets and sets the day component of the print date (if different from the actual date the report is printed).

Usage

[*form*.]*Report*.**PrintDay**[=*Day%*]

For example:

```
Report1.PrintDay = 23
```

« Sets 23 as the print day. »

Remarks

- Enter a value from 1 to 31
- The PrintYear, PrintMonth, and PrintDay properties function together. You must change the value of all three to change the print date. If you do not change all three, the print date saved with the report is used. This may be the current date if a specific date is not saved with the report.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

PrinterCopies Property

Description

Gets and sets the number of report copies to be printed.

Usage

[*form*.]*Report*.**PrinterCopies**[=*NumCopies*%]

For example:

```
Report1.PrinterCopies = 3
```

« Specifies that the program is to print three (3) copies of the report. »

Remarks

The number you use for PrinterCopies must not be zero or a negative value.

Data Type

Integer

Availability

Read and Write (Design time and Runtime)

ActiveX Control Section Codes

Select from the following section codes whenever you are asked to provide a section code value as a parameter for one of the ActiveX control properties:

To specify this section	Use this code	To specify this section	Use this code
All sections	ALL	Group Footer 1 section	GF1
Title section	TITLE	Group Footer 2 section	GF2
Page Header section	HEADER	Group Footer 3 section	GF3
Group Header 1 section	GH1	Group Footer 4 section	GF4
Group Header 2 section	GH2	Group Footer 5 section	GF5
Group Header 3 section	GH3	Group Footer 6 section	GF6
Group Header 4 section	GH4	Group Footer 7 section	GF7
Group Header 5 section	GH5	Group Footer 8 section	GF8
Group Header 6 section	GH6	Group Footer 9 section	GF9
Group Header 7 section	GH7	Grand Total section	GRNDTTL
Group Header 8 section	GH8	Page Footer section	FOOTER
Group Header 9 section	GH9	Summary section	SUMMARY
Details section	DETAIL		

Table of Properties

All of the properties for this control are listed in the following table. Properties marked with an asterisk(*) are standard Visual Basic control properties. See the *Visual Basic Language Reference* or Visual Basic Help for documentation of these properties.

Action	*Index	PrintFileName	SessionHandle
Connect	LastErrorNumber	PrintFileType	SortFields
CopiesToPrinter	LastErrorString	PrintFileUseRptDateFmt	SQLQuery
DataFiles	*Left	PrintFileUseRptNumberFmt	Status
Destination	MarginBottom	PrintMonth	StoredProcParam
DetailCopies	MarginLeft	PrintReport	*Top
DiscardSavedData	MarginRight	PrintYear	UserName
EMailCCList	MarginTop	RecordsPrinted	WindowBorderStyle
EMailMessage	*Name	RecordsRead	WindowControlBox
EMailSubject	Password	RecordsSelected	WindowControls
EMailToList	PrintDay	ReportDisplayPage	WindowHeight
EMailVIMBCCList	PrinterCollation	ReportFileName	WindowLeft
Formulas	PrinterCopies	ReportLatestPage	WindowMaxButton
GraphData	PrinterDriver	ReportStartPage	WindowMinButton
GraphOptions	PrinterName	ReportTitle	WindowParentHandle
GraphText	PrinterPort	SectionFont	WindowState
GraphType	PrinterStartPage	SectionFormat	WindowTitle
GroupCondition	PrinterStopPage	SectionLineHeight	WindowTop
GroupSelectionFormula	PrintFileCharSepQuote	SectionMinHeight	WindowWidth
GroupSortFields	PrintFileCharSepSeparator	SelectionFormula	

CopiesToPrinter Property

Description

Specifies the number of copies to be printed if you are printing to a printer (if the value you assign to the Destination property is *1* ■ *Printer*).

Usage

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

For example:

```
Report1.CopiesToPrinter = 3
```

« prints three copies of the specified report. »

Remarks

The number you enter must not be a zero or a negative value.

Data Type

Integer

Availability

Design time; Runtime

Destination Property

Description

Specifies the destination to which your report is to be printed (Window, Printer, or File).

Usage

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

For example:

```
Report1.Destination = 0
```

« sends the specified report to a print window. »

Remarks

Select one of the following print destinations:

- = Window (sends the report to a print window)
- = Printer (sends the report to a printer)
- = File (prints the report to a disk file for printing at a later time or for importing into other applications. If you select this property, you will also have to set the PrintFileName and PrintFileType properties.)
- = EMail via MAPI (sends the report to another person on your network via MAPI Email (Microsoft Mail). The report is attached to the EMail letter in the format specified by the PrintFileType property.)
- = EMail via VIM (sends the report to another person on your network via VIM Email (cc:Mail). The report is attached to the EMail letter in the format specified by the PrintFileType property.)

Data Type

Integer (Enumerated)

Availability

Design time; Runtime

PrintFileName Property

Description

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

Usage

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

For example:

```
Report1.PrintFileName = "c:\crw\cust_rpt.txt"
```

« prints the report to a file named "cust_rpt.txt" in the c:\crw directory. »

Remarks

- You can Double
- Click this property or Click the Ellipsis in the Settings box to call up the Choose Print Filename dialog box. In that dialog box, select the name of the file and the path to which you want the program to print the report.
- Select a value for this property only if you are printing to a file (if the value you assigned to the Destination property is *2*
- *File*).

NOTE: If you want to specify the PrintFileName at runtime, make certain that you enclose it in quotes in your code.

Data Type

String

Availability

Design time; Runtime

PrintFileType Property

Description

Specifies the type of print file used when printing a report to a file.

Usage

[*form*.]*Report*.**PrintFileType**[= *FileType*%]

For example:

```
Report1.PrintFileType = 1
```

« prints the report to a file in a tab separated format. »

Remarks

Select one of the following print file types if you are printing to a file (if the value you assigned to the Destination property is *2* ■ *File*).

0 ■ Record

Record style (columns of values). Doesn't use commas or separators. Outputs every record with a fixed field width.

1 ■ Tab separated

Tab separated values. Presents data in tabular form. Encloses alphanumeric field data in quotes and separates fields with tabs.

2 ■ Text

Text style. Saves the data in ASCII text format with all values separated by spaces. This style looks most like the printed page.

3 ■ DIF

Saves the data in DIF (data interchange format) format. This format is often used for the transfer of data between different spreadsheet programs.

4 ■ CSV

Comma separated values. Encloses alphanumeric field data in quotes and separates fields with commas.

5 ■ Character Separated

Saves the data as character separated values in ASCII text format. All values are separated by a character or characters specified by the PrintFileCharSepSeparator property.

6 ■ Tab separated Text

Saves the data in ASCII text format with all values separated by tabs.

7 ■ Crystal Reports RPT

Standard Crystal Reports RPT format is used. Most often used for sending the report to another user via EMail.

8 ■ Excel 2.1 XLS

Exports the report as a Microsoft Excel 2.1 Worksheet.

9 ■ Excel 3.0 XLS

Exports the report as a Microsoft Excel 3.0 Worksheet.

10 ■ Excel 4.0 XLS

Exports the report as a Microsoft Excel 4.0 Worksheet.

11 ■ Lotus 1
■2
■3 WK1

Exports the report as a Lotus 1■2
■3 WK1 format worksheet.

12 ■ Lotus 1
■2
■3 WK3

Exports the report as a Lotus 1■2
■3 WK3 format worksheet.

13 ■ Lotus 1
■2
■3 WKS

Exports the report as a Lotus 1■2
■3 WKS format worksheet.

14 ■ Quattro Pro 5.0 WB1

Exports the report as a Quattro Pro 5.0 WB1 format file.

15 ■ RTF

Saves the data in Rich Text Format.

16 ■ Word for DOS

Uses the Microsoft Word for DOS format to save the data in the report.

17 ■ Word for Windows

Uses the Microsoft Word for Windows format to save the data in the report.

18 ■ WordPerfect

Uses WordPerfect format to save the data in the report.

Data Type

Integer (Enumerated)

Availability

Design time; Runtime

ReportFileName Property

Description

Specifies the report to be printed.

Usage

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

For example:

```
Report1.ReportFileName = "c:\crw\company.rpt"
```

« prints the report named "company.rpt" that is located in the c:\crw directory. »

Remarks

You can Double-Click this property or Click the Ellipsis in the Settings box to call up the Choose Report File dialog box. In that dialog box, select the name and path of the report you want the program to print in response to a CRW ActiveX control event.

NOTE: If you want to specify the ReportFileName at runtime, make certain that you enclose it in quotes in your code.

Data Type

String

Availability

Design time; Runtime

SelectionFormula Property

Description

Specifies the records to be used when printing the report.

Usage

[*form*.]*Report*.**SelectionFormula**[= *SelectionFormula*$]

Enter the selection formula just as you would enter it in the Formula Editor in Crystal Reports. For example, to include only those records that have a quantity bigger than 5 in the {file.Qty} field, you would enter "{file.QTY} > 5" as your selection formula, i.e., Report1.SelectionFormula = "{file.QTY} > 5"

Remarks

- Make certain that you enclose your selection formula in double quotes.
- If your selection formula includes internal quotes, for example:

```
{file.STATE} = "CA"
```

change all of the internal double quotes to single quotes and then surround the entire selection formula in double quotes like this:

```
"{file.STATE} = 'CA'"
```

- If you have created a selection formula in your report at design time, any selection formula you enter here will be appended to that selection formula. Thus, your records will be selected based on a combination of the two selection formulas.

Data Type

String

Availability

Design time; Runtime

See Also

Edit Record Selection Formula

WindowBorderStyle Property

Description

Specifies the type of border for the print window.

Usage

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

For example:

```
Report1.WindowBorderStyle = 2
```

« sets a sizable border style (Style #2) for the print window. »

Remarks

Select one of the following border styles for the print window:

- None (creates a window with no border).
- Fixed Single (creates a window of a fixed size with a single line border).
- Sizeable (creates a window that can be resized by the user).
- Fixed Double (creates a window of fixed size with a double line border).

Select a value here only if you are printing to a window (if Destination = 0).

Data Type

Integer (Enumerated)

Availability

Design time; Runtime

WindowControlBox Property

Description

Specifies whether or not the print window is to have a control (system menu) box in the upper left hand corner when the report is printed to a window.

Usage

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

For example:

```
Report1.WindowControlBox = True
```

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

Remarks

- Select True if you want the window to contain a control box. Select False if you don't.
- Select a value here only if you are printing to a window (if Destination = 0).

Data Type

Integer

Availability

Design time; Runtime

WindowHeight Property

Description

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

Usage

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

For example;

```
Report1.WindowHeight = 300
```

« sets the height of the print window to 300 pixels. »

Remarks

- If you are not satisfied with the default settings, enter the external height you want for your print window in pixels.
- Select a value here only if you are printing to a window (if Destination = 0).

Data Type

Integer

Availability

Design time; Runtime

WindowLeft Property

Description

Sets the distance, in pixels, that the print window is to appear from the left edge of the parent window. If the print window is a top level window, then the distance is measured from the left edge of the screen.

Usage

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

For example:

```
Report1.WindowLeft = 100
```

« sets the left edge of the print window 100 pixels from the left edge of the screen. »

Remarks

- If you are not satisfied with the default settings, enter the number of pixels you want between the left edge of the screen and the left edge of your window.
- Select a value here only if you are printing to a window (if Destination = 0).

Data Type

Integer

Availability

Design time; Runtime

WindowMaxButton Property

Description

Specifies whether or not the print window is to have a maximize button when the report is printed to a window.

Usage

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

For example:

```
Report1.WindowMaxButton = False
```

« specifies that no Maximize button is to appear in the print window. »

Remarks

- Select True if you want the window to contain a maximize button. Select False if you don't.
- Select a value here only if you are printing to a window (if Destination = 0).

Data Type

Integer

Availability

Design time; Runtime

WindowMinButton Property

Description

Specifies whether or not the print window is to have a minimize button when the report is printed to a window.

Usage

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

For example:

```
Report1.WindowMinButton = True
```

« specifies that a Minimize button is to appear in the print window. »

Remarks

- Select True if you want the window to contain a minimize button. Select False if you don't.
- Select a value here only if you are printing to a window (if Destination = 0).

Data Type

Integer

Availability

Design time; Runtime

WindowTitle Property

Description

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

Usage

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

For example:

```
Report1.WindowTitle = "Quarterly Earnings"
```

« sets the title of the print window (the string that appears on the title bar) to "Quarterly Earnings" »

Remarks

- Make sure that the title is enclosed in quotes.
- Select a value here only if you are printing to a window (if Destination = 0).

Data Type

String

Availability

Design time; Runtime

WindowTop Property

Description

Sets the distance, in pixels, that the print window is to appear from the top edge of the parent window. If the print window is a top level window, then the distance is measured from the top edge of the screen.

Usage

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

For example:

```
Report1.WindowTop = 100
```

« sets the top edge of the print window 100 pixels from the top of the screen. »

Remarks

- If you are not satisfied with the default setting, enter the number of pixels you want between the top of the screen and the top of your window.
- Select a value here only if you are printing to a window (if Destination = 0).

Data Type

Integer

Availability

Design time; Runtime

WindowWidth Property

Description

Specifies the width of the print window in pixels.

Usage

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

For example:

```
Report1.WindowWidth = 480
```

« specifies a print window 480 pixels wide. »

Remarks

- If you are not satisfied with the default setting, enter the external width of your window, in pixels.
- Select a value here only if you are printing to a window (if Destination = 0).

Data Type

Integer

Availability

Design time; Runtime

Error Messages

The following table lists the trapable run-time errors for the Crystal ActiveX Control:

Error number	Message explanation
20500	Not Enough Memory For Operation or Not Enough Memory To Get Selection Formula or Cannot Get Selection Formula. There is not enough memory available to complete the call.
20501	Invalid Job Number. Internal error.
20502	Invalid Text Handle or Parent Window Cannot Be An MDI Form orInvalid Parent Window Handle. You have specified an invalid window handle, or you have specified an MDI form as the parent of a print window.
20503	Buffer Too Small For String. Internal error.
20504	Report Not Found. You have specified a report that does not exist.
20505	No Print Destination Specified or Invalid Print Destination. The Destination property must be 0, 1, or 2. You have specified a print destination other than one of these values.
20506	Invalid File Number. You have tried to set an Nth file name and the file number you specified is out of the existing range: 0<= fileN < N files.
20507	Invalid File Name. There is an error in the file name you specified.
20508	Invalid Field Number. Internal error.
20509	Invalid Field Name. You specified an invalid database field for a sort field. The program can't add the sort field name you specified.
20510	Invalid Formula Name. The formula name you specified is invalid or non-existent.
20511	Invalid Sort Direction. A sort field must be preceded by a + or - to indicate a valid sort direction.
20512	Print Engine Not Open. Internal error.
20513	Invalid Printer The printer driver for the printer you specified is missing.
20514	Print File Exists. The name you have specified for the print file already exists. You must delete the file and print again or specify a different file.
20515	Error In Formula. There is a formula error in the replacement formula text. Review the formula syntax and retry.
20517	Print Engine Already In Use. Internal error.

20520 Print Job Already Started. You are trying to start a print job that has already been started. This can happen if the user starts a print job and then tries to start printing again before the previous printing has finished.

PROGRAMMER'S NOTE: Disable the form while Action is in process and re■enable the form once Action is complete. This will help avoid the conflict that generates this message.

20521 Invalid Summary Field. The summary field specified as a group sort field is invalid or non■ existent.

20522 Not Enough System Resources. There are not enough Windows system resources to process the function.

20524 Print Job Busy. You tried to initiate printing while Crystal Reports is already printing a job.

Programmer's Note: Disable the form while Action is in process and re■enable the form once Action is complete. This will help avoid the conflict that generates this message.

20525 Unable To Load Report. There is something wrong with the report you are trying to open.

20526 No Default Printer. You haven't specified a default printer. Specify a default printer via the Windows Control Panel.

20527 SQL Server Error. There is a problem connecting with the SQL server.

20529 Disk Full. When printing to file or when sorting, the program requires more room than is available on the disk.

20530 File I/O Error. In trying to print to file, the program is encountering another file problem besides disk full.

20531 Incorrect Password. You have specified an incorrect password.

20532 Missing Database DLL. The database DLL is corrupt.

20533 Unable To Open Database File. Something is wrong with the database you have specified. You may need to verify using the Database| Verify Database command.

20534 Error Detected By Database DLL. The database DLL is corrupt.

20535 Unable To Connect:Incorrect Session Parameters. You have attempted to log on using incomplete or incorrect session parameters.

20536 Unable To Connect:Incorrect Log On Parameters. You have attempted to log on using incomplete or incorrect log on parameters.

20537 Unable To Connect:Incorrect Table Location. The table you have specified cannot be found.

20538 Parameter Has Invalid Structure Size. Internal error.

20539 Invalid Print Date You have specified an Invalid date using the PrintYear, PrintMonth, and PrintDay properties.

20540 Missing or Out■of■date Export DLL The DLL required by your export call is either missing or out■of■date.

20541 Error in Export DLL An export DLL has reported an error.

20542 Already at the First Page Youre using the *previous page* button control in the Preview window when youre already at the first page of the report.

20543 Already at the Last Page Youre using the *next page* button control in the Preview window when youre already at the last page of the report.

20544 Report Open by Another User Access to the report file is denied. Another program or user may be using it.

20545 User Has Cancelled Operation The user has Click*ed* the Cancel button in a dialog box during the session.

20546 OLE 2.0 Not Available The program cant open the report (which includes an OLE 2.0 object) because OLE 2.0 cannot be loaded.

20547 Invalid Crosstab Group You have specified an invalid row or column field in your crosstab report.

20548 Crosstab is Missing a Summarized Field You are trying to run a crosstab report without specifying a summarized field.

20550 Invalid Page Number You have specified a page number that does not exist in the current report.

20552 No Stored Procedure in the Report The current report does not use a table based on a stored procedure.

20553 Invalid Parameter The parameter you have specified does not exist in the stored procedure, or the value you have specified for a parameter is invalid for that parameter.

20554 Graph Does Not Exist The graph number you have indicated for the section does not exist.

20555 Graph Type is Invalid The type of graph you

	have specified with the GraphType property is invalid.
	Graph Data is Invalid This error occurs when you use the GraphData property and: the report is a crosstab, and the value of *col* or *row* is > 1, or the report is not a crosstab, and *row* is something other than ■1 or *col* + 1.
20557	Graph col Value Cannot be Changed If the report is not a crosstab, the *col* value of the GraphData property cannot be different than the graphs current value.
20999	Operation Not Yet Implemented. Internal error.

Action Property

Description

Action is the property that triggers the printing of the report.

Usage

[*form*.]*Report*.**Action** = 1

For example:

```
Report1.Action = 1
```

«Prints the specified report »

Remarks

Set the Action property to 1 in your procedure code (Report1.Action = 1) to print the report in response to a user event.

Data Type

Integer

Availability

Write-only at runtime

BoundReportFooter

Description

True or False property that specifies whether or not the Page Footer is to appear in the Default Bound Report.

Usage

[form}.CrystalReport1.BoundReportFooter = True

For example:

```
Report1.BoundReportFooter = True
```

Availability

Design/Runtime

BoundReportHeading

Description

String property that allows the user to give a title to the report.

Usage

[form].CrystalReport1.BoundReportHeading = "Report Title"

For example:

```
Report1.BoundReportHeading = "Sales by Region"
```

«Specifies the title of the report.»

Availability

Design/Runtime

Connect Property

Description

Logs on to a SQL server.

Usage

[*form*.]*Report*.**Connect**[= *Name;UserID;Password;DatabaseQualifier$*]

For example:

```
Report1.Connect = "DSN = Accounting;UID = 734;PWD = bigboard;DSQ =
Administration"
```

« connects to the "Administration" database on the "Accounting" server using the user ID #734 and the password "bigboard." »

Remarks

Enter the parameters necessary to log on to the SQL server that you need to be activated for your report.

Parameters should be in the following format:

```
DSN = name;UID = userID;PWD = password;DSQ = database qualifier
```

- *name* is the server name.
- *user ID* is the name you have been assigned for logging onto the SQL server.
- *password* is the password you have been assigned for logging onto the SQL server.
- *database qualifier* is the database name if your server uses the database concept.

NOTE: Before you can use this property, you must:

- ***install the ODBC driver for whatever SQL database you are planning to use, and***
- ***put the Database/BIN location in your path.***

NOTE: This parameter is required only when it is applicable to the ODBC driver you are using.

Data Type

String

Availability

Design time;Runtime

DataFiles Property

Description

Specifies the location of the database files or tables used in the report.

Usage

[*form*.]*Report*.**DataFiles**(ArrayIndex)[= *Location$*]

- Enter the file name and path of each database file or table in your report for which you want to change the location.
- Use a separate line of code for each file for which you want to change the location.
- The order of files in the array must conform to the order of files in the report. (You can use Database| Set Location to determine the order of files in the report.)
- The first file in the report is array index (0), the second file is (1), etc.

For example, to change the location of the first and third files in a report (first.dbf and third.dbf) to the c:\new directory, use the following syntax:

```
Report1.DataFiles(0) = "c:\new\first.dbf"
Report1.DataFiles(2) = "c:\new\third.dbf"
```

Remarks

- DataFiles is an array property that is available at runtime only.
- Use this property if you want to run the report with files in different locations than specified in the report.
- If you use this property, you do not have to change the locations of all files in the report. Just make certain that the array index for each file you do change matches the position of that file in the report.
- This property is cleared once the print job is printed. If you print a second time, the program reverts to the locations as originally specified in the report.

Data Type

Array of strings

Availability

Runtime

DataSource

Description

Specifies which datacontrol the Crystal ActiveX control is to look at for the data.

Availability

Design time only

Formulas Property

Description

Specifies a new string for an existing formula.

Usage

[*form*.]Report.**Formulas**(ArrayIndex)[= "*FormulaName*= *FormulaText*"]

Enter the formula name and the string that you want to replace the existing string for each formula that you want to change in your report.

For example, to change a formula @COMMISSION to {file.SALES}*.1, and a second formula @TOTAL to {file.SALES} + {file.COMMISSION}, enter the following:

```
Report1.Formulas(0) = "COMMISSION= {file.SALES} * .1"
Report1.Formulas(1) = "TOTAL= {file.SALES} + {file.COMMISSION}"
```

Remarks

- Formulas is an array property that is available at runtime only.
- Use a separate line of code for each formula you want to change.
- Change only those formulas that you want to change.
- The first formula you change must be assigned array index (0), the second must be assigned array index (1), etc.
- The new formula string must conform to Crystal Reports syntax requirements.
- This property is cleared once the print job is printed. If you print a second time, the program reverts to the formulas as originally specified in the report.

NOTE: Spaces are significant in formula names. For this reason, the equal sign must follow the formula name with no intervening spaces.

NOTE: The @ sign is not used when designating a formula name in this property.

NOTE: You cannot use this property to create new formulas. You can only use it to change existing formulas.

Data Type

Array of strings

Availability

Runtime (read and write)

GroupSelectionFormula Property

Description

Specifies the groups to be used when printing the report.

Usage

[*form*.]Report.**GroupSelectionFormula**[= "GroupSelectionFormula"]

Enter the group selection formula just as you would enter it in the Formula Editor. For example, to limit your report to those groups with a subtotal on the header.AMOUNT field less than $10,000 (with subtotals triggered by changes in the header.CUSTNUMB field), you would enter the following as a group selection formula:

```
Sum ( {header.AMOUNT}, {header.CustNumb}) < $10000
```

Remarks

If your group selection formula includes internal quotes, change all of the internal double quotes to single quotes and then surround the entire selection formula in double quotes.

NOTE: If you have created a group selection formula in your report at design time, any group selection formula you enter here will be appended to that group selection formula, connected by an "and." Thus, your records will be selected based on a combination of the two formulas

Data Type

String

Availability

Design time; Runtime

GroupSortFields Property

Description

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

Usage

[*form*.]*Report*.**GroupSortFields**(ArrayIndex)[= "{+|■}*GroupField"*]

Enter the group field(s) on which you want your report to be sorted.
For example, assume that you have broken your data into state groups and had Crystal Reports count the number of customers in each group. In order to print the group with the highest count first, then the group with the next highest count, etc. (descending order), you would enter a string similar to the following:

```
Report1.GroupSortFields(0) = "■Count ({customer.CUSTOMER},
{customer.STATE})"
```

Remarks

- GroupSortFields is an array property available at runtime only.
- Use a separate line of code to specify each group sort field.
- Enter group sort fields in the order that you want them to sort your report. For example, if you want your report to be sorted first on group sort field A and then on group sort field B, specify group sort field A in your first line of code and group sort field B in your second line of code.
- The first group sort field you specify must be assigned array index 0, the second group sort field must be assigned array index 1, etc.
- The index values you assign must be continuous; no gaps are allowed (0,1,2 = OK, 0,1,3 = wrong)
- Array index values must be subscripted in the code immediately after the property name, i.e., Report1.GroupSortFields(0) =.
- If you have specified sort fields for your report at design time, any sort fields you enter here will replace the sort fields in your report.
- If you don't use this property, the program will use the sorting instructions that you specified in the report.
- If you want to clear the group sort fields in your report, use an empty string (Report1.GroupSortFields(0) = "")
- This property is cleared once the print job is printed. If you print a second time, the program reverts to the group sort fields as originally specified in the report.

NOTE: The group sort field entry must follow the sort direction sign (+ or ■) with no intervening space.

NOTE: To find the correct syntax for any group in your report using Crystal Reports for Visual Basic:

- ***select Insert|Formula field,***
- enter any formula name in the Insert Formula Dialog box when it appears,
- Click the scroll button on the Fields list in the Formula Editor when it appears, and
- Double
- ■Click the group field of interest.

Crystal Reports enters the group field name in the Formula Text box. Use the name and syntax from that text box when constructing your group sort field string.

Data Type

Array of strings

Availability

Runtime

LastErrorNumber Property

Description

Returns the error code for the last runtime error.

Usage

[*form*.]*Report*.**LastErrorNumber**

For example:

```
'If error occurs, go to Error Handler
ErrorHandler:
MsgBox  Report1.LastErrorNumber
```

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

Remarks

LastErrorNumber is a runtime

only property.

NOTE: LastErrorNumber must come after the Action call in order to display relevant values. After you have printed your report, you can refer to this property to get an error number (if any). If there was no error in printing, LastErrorNumber = 0.

Data Type

Integer

Availability

Runtime (read and write)

LastErrorString Property

Description

Returns the error string for the last runtime error.

Usage

[*form*.]*Report*.**LastErrorString**

For example:

```
'If error occurs, go to Error Handler
ErrorHandler:
MsgBox  Report1.LastErrorString
```

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

Remarks

LastErrorString is a runtime

only property.

NOTE: LastErrorString must come after the Action call in order to display relevant values. After you have printed your report, you can refer to this property to get an error string (if any). If there was no error in printing, LastErrorNumber = 0.

Data Type

String

Availability

Runtime (read and write)

LogOnInfo Property

Description

LogOnInfo logs on to one or more SQL servers or a password-protected databases...

Usage

[form.]Report.LogOnInfo(ArrayIndex)[= Name;UserID;Password;DatabaseQualifier$]

for example

```
CrystalReport1.LogOnInfo[0] = "DSN = Accounting;UID = 734;PWD =
bigboard;DSQ = Administration"
```

«Connects to the "Administration" database on the "Accounting" server using the user ID #734 and the password "bigboard".»

```
CrystalReport1.LogOnInfo[0] = dsn=;uid=;pwd=bigboard;dsq=
```

«Connects to a password-protected Paradox database. All that is being passed is the password bigboard»

Remarks

- Use a separate line of code for each table for which you want to change the logon info.
- The order of tables in the array must conform to the order of tables in the report. (You can use Database|Set Location to determine the order of tables in the report.)
- The first table in the report is array index (0), the second file is (1), etc. For example, to change the logon info of the first and third tables in a report to the NEW server, use the following syntax:

```
CrystalReport1.LogOnInfo(0) = "DSN = NEW;UID = 734;PWD = bigboard;DSQ =
Administration1"CrystalReport1.LogOnInfo(2) = "DSN = NEW;UID = 734;PWD =
bigboard;DSQ = Administration2"
```

- LogOnInfo is an array property that is available at runtime only.
- Enter the parameters necessary to log on to each SQL server table that you need to change info for in your report. Parameters should be in the following format:

```
DSN = name;UID = userID;PWD = password;DSQ = database qualifier
```

- Name is the server name,
- userID is the name you have been assigned for logging on to the SQL server,
- password is the password you have been assigned for logging on to the SQL server, and
- database qualifier is the database name if your server uses the database concept.

NOTE: This parameter is required only when it is applicable to the ODBC/SQL driver you are using.

NOTE: Before you can use this property for an ODBC/SQL database, you must install the ODBC/SQL driver for whatever SQL database you are planning to use, and put the Database/BIN location in your path.

Data Type

Array of strings

Availability

Runtime

Password Property

Description

Enters the password needed to use database tables on a restricted Access .mdb file.

Usage

[*form*.]*Report*.**Password**[= *Password$*]

For example:

```
Report1.Password = "dogsncats"
```

« enters the password "dogsncats" »

Remarks

Enter the password you have been assigned.

Data Type

String

Availability

Runtime

ReportSource

Description

Specifies where the "report template" is coming from.

Usage

[form].CrystalReport1.ReportSource = [TrueFalse%]

For example:

```
Report1.ReportSource = 1
```

Availability

Design/Runtime

SessionHandle Property

Description

Sets the session handle for a user once the UserName and Password properties have opened an Access .mdb file for use by the report.

Usage

[*form*.]*Report*.**SessionHandle**[= *Handle%*]

Remarks

If you have already opened a Jet session in your Visual Basic application, you can set this property to be the current session handle. Otherwise you will have to use the Password and UserName properties to establish the Jet session.

For example:

```
Report1.SessionHandle = CurrentSessionHandle
```

« sets the session handle to the session handle returned elsewhere in the application and stored in the variable CurrentSessionHandle »

Data Type

Integer

Availability

Runtime (read and write)

SortFields Property

Description

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

Usage

[*form*.]*Report*.**SortFields**(ArrayIndex)[= "{+|■}*SortField*"]

Enter the fields on which you want the data in your report to be sorted.
For example, to sort an order database alphabetically by customer, and then by order date, you can enter code similar to this:

```
Report1.SortFields(0) = "+{orders.CUSTOMER}"
Report1.SortFields(1) = "+{orders.ORDERDATE}"
```

Remarks

- SortFields is an array property available only at runtime.
- Use a separate line of code to specify each sort field.
- Enter sort fields in the order that you want them to sort your report. For example, if you want your report to be sorted first on field A and then on field B, specify sort field A in your first line of code and sort field B in your second line of code.
- The sort field you specify must be assigned array index 0, the second sort field must be assigned array index 1, etc.
- The index values you assign must be continuous; no gaps are allowed (0,1,2 = OK, 0,1,3 = wrong)
- Array index values must be subscripted in the code immediately after the property name, i.e., Report1.SortFields(0) =.
- If you have specified sort fields for your report at design time, any sort fields you enter here will replace the sort fields in your report.
- If you don't use this property, the program will use the sorting instructions that you specified in the report.
- If you want to clear the sort fields in your report, use an empty string (Report1.SortFields(0) = "")
- Enclose field names in braces.
- Sort fields can be database fields or formula fields. If you sort on a formula field, use the @ sign before the formula name, i.e., @FORMULANAME.

Data Type

Array of strings

Availability

Runtime (read and write)

UserName Property

Description

Enters the name given to a user for logging on to a protected Access .mdb file to obtain data files needed by the report.

Usage

[*form*.]*Report*.**UserName**[= *Name$*]

For example:

```
Report1.UserName = "MIS"
```

« enters the user name "MIS" »

Remarks

Enter the name you have been assigned.

The name must be enclosed in quotes if the variable is being assigned at runtime.

Data Type

String

Availability

Runtime

WindowParentHandle Property

Description

Specifies the handle of the parent window if the print window is to be the child of another window.

Usage

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

Remarks

This is a runtime-only property.

For example:

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

« sets the WindowParentHandle to the handle of Form1. This specifies that the print window is to be a child of Form1 »

Data Type

Integer

Availability

Runtime (read and write)

Properties Index

Click on a letter below to go to the index topics beginning with that letter.

A

Action	Action is the property that triggers the printing of the report.

C

Connect	Logs on to a SQL server.
CopiesToPrinter	Specifies the number of copies to be printed if you are printing to a printer

D

DataFiles	Specifies the location of the database files or tables used in the report.
Destination	Specifies the destination to which your report is to be printed (Window, Printer, or File).
DetailCopies	Specifies the number of copies of each record in the Details section that the program is to print.
DiscardSavedData	If data is saved with the specified report, setting this property to 1 (True) discards the data.

E

EMailCCList	Specifies the "CC" list to which you want your E■mail message sent.
EMailMessage	Specifies the message line in your E■ mail message.
EMailSubject	Specifies the subject line in your E■mail message.
EMailToList	Specifies the "To" list to which you want your E■mail message directed.
EMailVIMBCCList	Specifies the "Blind CC" list to which you want your E■mail message copied.

F

Formulas	Specifies a new string for an existing formula.

G

GraphData	Gets and sets the data used for a specified graph.

GraphOptions	Gets and sets a number of options for the specified graph.
GraphText	Gets and sets the various text components for the specified graph.
GraphType	Gets and sets the kind of graph used in the selected section in the specified report.
GroupCondition	Specifies what kind of change in the Group Condition Field will trigger the creation of a group.
GroupSelectionFormula	Specifies the groups to be used when printing the report.
GroupSortFields	Specifies the group field(s) that are to be used to sort your data when the report is printed.

I

Index	*See Visual Basic documentation*

L

LastErrorNumber	Returns the error code for the last runtime error.
LastErrorString	Returns the error string for the last runtime error.
Left	*See Visual Basic documentation*

M

MarginBottom	Gets and sets the bottom margin for the specified report.
MarginLeft	Gets and sets the left margin for the specified report.
MarginRight	Gets and sets the right margin for the specified report.
MarginTop	Gets and sets the top margin for the specified report.

N

Name	*See Visual Basic documentation*

P

Password	Enters the password needed to use database tables on a restricted Access .mdb file.
PrintDay	Gets and sets the day component of the print date (if different from the actual date the report is printed).
PrinterCollation	If you specify more than one copy to be printed (using the PrinterCopies

	property), PrinterCollation specifies whether or not the copies will be collated.
PrinterCopies	Gets and sets the number of report copies to be printed.
PrinterDriver	Gets and sets the name of the printer driver that is to print the report.
PrinterName	Gets and sets the name of the printer that is to print the report.
PrinterPort	Gets and sets the name of the printer port that is to print the report.
PrinterStartPage	Gets and sets the first page to be printed.
PrinterStopPage	Gets and sets the last page to be printed.
PrintFileCharSepQuote	Gets and sets the quote character used to enclose alphanumeric field data when printing to a file using Character Separated format.
PrintFileCharSepSeparator	Gets and sets the character(s) you want to use to separate the fields when printing to a file using the Character Separated Value format.
PrintFileName	Specifies the name of the file to which the report is to be printed.
PrintFileType	Specifies the type of the file to which the report is to be printed.
PrintFileUseRptDateFmt	When printing to a file, indicates whether or not the program should save dates in the same date format (MDY, DMY, etc.) that is used in the report or instead optimize the dates for the file format you have selected.
PrintFileUseRptNumberFmt	When printing to a file, indicates whether or not the program should print numbers in the same format (decimal places, negatives, etc.) that you have used in the report or instead optimize the numbers for the file format you have selected.
PrintMonth	Gets and sets the month component of the print date (if different from the actual date the report is printed).
PrintReport	PrintReport triggers the printing of the report.
PrintYear	Gets and sets the year component of the print date (if different from the actual date the report is printed).

R

RecordsPrinted	Determines the number of records

	actually printed.
RecordsRead	Determines the number of records actually read.
RecordsSelected	Determines the number of records selected.
ReportDisplayPage	Indicates which page of a multi-page report is currently being displayed in the Preview window.
ReportFileName	Specifies the report to be printed.
ReportLatestPage	Determines the last page printed in the specified report.
ReportStartPage	Determines the first page printed in the specified report.
ReportTitle	Specifies the report title.

S

SectionFont	Specifies the font for a section.
SectionFormat	Specifies the format for a section.
SectionLineHeight	Specifies the line height for a section.
SectionMinHeight	Sets the minimum section height for the specified report section.
SelectionFormula	Specifies the records to be used when printing the report.
SessionHandle	Sets the session handle for a user once the UserName and Password properties have opened an Access .mdb file for use by the report.
SortFields	Specifies the field(s) that are to be used to sort your data when the report is printed.
SQLQuery	Gets and sets the SQL query string used by the specified report.
Status	Determines the print status for the specified report.
StoredProcParam	Gets and sets the stored procedure parameters when using a report based on SQL stored procedures.

T

Top	*See Visual Basic documentation*

U

UserName	Enters the name given to a user for logging on to a protected Access .mdb file to obtain data files needed by the report.

W

WindowBorderStyle	Specifies the type of border for the print window.
WindowControlBox	Specifies whether or not the print window is to have a control (system menu) box in the upper left hand corner when the report is printed to a window.
WindowControls	Specifies whether or not the print controls are to appear in the Preview window when printing a report to a window.
WindowHeight	Sets the height of the print window when the report is printed to a window.
WindowLeft	Sets the distance, in pixels, that the print window is to appear from the left edge of the parent window. If the print window is a top level window, then the distance is measured from the left edge of the screen
WindowMaxButton	Specifies whether or not the print window is to have a maximize button when the report is printed to a window.
WindowMinButton	Specifies whether or not the print window is to have a minimize button when the report is printed to a window.
WindowParentHandle	Specifies the handle of the parent window if the print window is to be the child of another window.
WindowState	Sets the state of the Preview window, normal, minimized, or maximized, when the report is printed to a Preview window.
WindowTitle	Specifies the title you want to appear in the print window title bar when the report is printed to a window.
WindowTop	Sets the distance, in pixels, that the print window is to appear from the top edge of the parent window. If the print window is a top level window, then the distance is measured from the top edge of the screen.
WindowWidth	Specifies the width of the print window in pixels.

Methods

FetchSelectionFormula Method

Description

FetchSelectionFormula returns the selection formula from the current report.

Usage

[*form*.]*Report*.**FetchSelectionFormula**

For example:

```
SelectionFormula$ = CrystalReport1.FetchSelectionFormula
```

« retrieves the selection formula from CrystalReport1 »

Remarks

This method does NOT populate the SelectionFormula property and it DOES NOT conflict with setting the property. Both the method and the property can be used in the same code.

Availability

Runtime only

PrintReport Method

Description

PrintReport triggers the printing of the report.

Usage

[*form*.]*Report*.**PrintReport**

For example:

```
Result% = Report1.PrintReport
```

« prints the specified report. »

Remarks

- PrintReport returns a result code, 0 if the call is successful, an error code in the 20XXX range if it fails.
- You can also print a report using the Action property. If something goes wrong, however, you get a runtime error that will terminate your application. For this reason, you will need to set up an error handler.

Availability

Runtime only

ReplaceSelectionFormula Method

Description

ReplaceSelectionFormula overrides the selection formula from the current report with the string that is passed.

Usage

[*form*.]*Report*.**ReplaceSelectionFormula** [(SelectionFormulaString$)]

For example:

```
CrystalReport1.ReplaceSelectionFormula ("{Company.State}='CA'")
```

« uses "{Company.State}='CA'" as the selection formula for the report.»

Remarks

This method DOES NOT use the string in the SelectionFormula property and DOES conflict with setting the property. You CANNOT set the SelectionFormula property and call ReplaceSelectionFormula in the same code sequence. A VB error condition will be raised in such a case.

Availability

Runtime only

RetrieveDatafiles Method

Description

RetrieveDatafiles retrieves all "table" locations from the current report, populates the Datafiles property, and returns the number of "tables" in the report.

Usage

[*form*.]*Report*.**RetrieveDatafiles**

For example:

```
NumberofDatafiles% = CrystalReport1.RetrieveDatafiles
```

« populates the Datafiles property with the table locations from CrystalReport1 »

Remarks

This method can only be called AFTER the ReportFileName property has been set.

Availability

Runtime only

RetrieveLogOnInfo Method

Description

RetrieveLogOnInfo retrieves logon information (except for the password) for all "tables" in the current report, populates the LogOnInfo property, and returns the number of "tables" in the report.

Usage

[*form*.]*Report*.**RetrieveLogOnInfo**

For example:

```
NumberofTables% = CrystalReport1.RetrieveLogOnInfo
```

« retrieves the logon information for all the tables in CrystalReport1 »

Remarks

This method can only be called AFTER the ReportFileName property has been set. This method DOES NOT use the string in the Connect property and DOES conflict with setting the property. You CANNOT set the Connect property and call RetrieveLogOnInfo in the same code sequence. A VB error condition will be raised in this case.

Availability

Runtime only

RetrieveSQLQuery Method

Description

RetrieveSQLQuery retrieves the SQL Query from the current report and populates the SQLQuery property.

Usage

[*form*.]*Report*.**RetrieveSQLQuery**

For example:

```
CrystalReport1.RetrieveSQLQuery
```

« retrieves the SQL query from CrystalReport1 »

Remarks

This method can only be called AFTER the ReportFileName property has been set.

Availability

Runtime only

RetrieveStoredProcParams Method

Description

RetrieveStoredProcParams retrieves all stored procedure parameters from the current report, populates the StoredProcParams property, and returns the number of parameters.

Usage

[*form*.]*Report*.**RetrieveStoredProcParams**

For example:

```
NumberofParams% = CrystalReport1.RetrieveStoredProcParams
```

« retrieves the stored procedure parameters from CrystalReport1. »

Remarks

This method can only be called AFTER the ReportFileName property has been set.

Availability

Runtime only

LogonServer Method

Description

The LogonServer method logs on to the specified server and returns a unique connection id which can be used to log off of this server using the LogoffServer method.

Usage

[*form*.]*Report*.**LogonServer** (dllName$, ServerName$, DatabaseName$, UserID$, Password$)

For example:

```
connectionId% = CrystalReport1.LogonServer ("pdsodbc.dll", "Accounting",
"bobg", "Administration", "bigboard")
```

«Connects to the "Administration" database via the "Accounting" datasource using the user ID "bobg" and the password "bigboard".»

Parameters

Parameter	
dllName	Specifies the name of the Crystal Reports DLL for the server or password protected nonSQL table you want to log onto, for example, "PDSODBC.DLL". Note that the dllName must be enclosed in quotes. DLL names have the following naming convention: PDB*.DLL for standard (nonSQL) databases, PDS*.DLL for SQL/ODBC databases.
ServerName	Specifies the logon name for the server used to create the report. *For ODBC, use the data source name.
DatabaseName	Specifies the logon name for the database used to create the report.
UserID	Specifies the user ID necessary to log on to the server.
Password	Specifies the password necessary to log on to the server. When you are using this structure to retrieve information using the PEGetNthTableLogOnInfo function, the password parameter is undefined.

LogoffServer Method

Description

The LogoffServer method terminates the specified database connection established earlier with the LogonServer method.

Usage

[*form*.]*Report*.**LogoffServer** (connectionId%, boolean allConnections)

For example:

```
CrystalReport1.LogoffServer (1, False)
```

«Terminates database connection 1 and only that connection.»

Parameters

Parameter	
connectionId	Integer value that specifies a specific database connection establish earlier with the LogonServer method.
allConnections	Boolean value that specifies whether or not to terminate ALL database connections that have been established with the LogonServer method. True = Terminate all connections. False = Terminate only the specified connection.

Foreign Language Runtime File Requirements

If you will be distributing a runtime version of the Crystal Reports print engine with your applications in a language other than English, you must make sure that you include the correct resource files to ensure that the files will appear in the appropriate language.

Each language has a two letter abbreviation that appears directly before the extension of the file name. In the table below, the two letter abbreviation has been replaced with **xx**.

16-bit file requirements

Original File Name	Resource File Name
files\co1c4en.dll	co1c40xx.dll
files\crsupprt.dll	cs1c40xx.dll
files\crw.exe	cr1c40xx.dll
files\pdbbnd.dll	p1d4bdxx.dll
files\pdqqbe.dll	p1q4qbxx.dll
files\pdsodbc.dll	p1s4odxx.dll
files\uxddisk.dll	x1d4kxx.dll
files\uxdmapi.dll	x1d4mpxx.dll
files\uxdvim.dll	x1d4vmxx.dll
files\uxfcr.dll	x1f4crxx.dll
files\uxfdif.dll	x1f4dfxx.dll
files\uxfdoc.dll	x1f4dcxx.dll
files\uxfqp.dll	x1f4qpxx.dll
files\uxfrec.dll	x1f4rcxx.dll
files\uxfrtf.dll	x1f4rtxx.dll
files\uxfsepv.dll	x1f4svxx.dll
files\uxftext.dll	x1f4txxx.dll
files\uxfwks.dll	x1f4wkxx.dll
files\uxfwordw.dll	x1f4wdxx.dll
files\uxfxls.dll	x1f4xlxx.dll

All dll's contain the English language strings, dialogs, menus, etc. If you do not specify a language resource in the path, English will automatically be used.

To use this language resource file	Use the dll's that end with these letters
French	fr
Deutsch (German)	de
Italian	it
Japanese	jp

If you specify a language resource other than English, you must also set the language setting in the Windows control panel \ International settings to the same language. If you do not change the setting in Windows control panel, English will automatically be used.

32-bit file requirements

Crystal Reports for Windows 32-bit version uses a naming convention for foreign language resource requirements. These resource files are only required in foreign language applications (other than English).

Database Requirements

Character

1. The first character in the file name is the letter P, it specifies that this resource is a database resource.
2. The second character specifies 1 for 16-bit, or 3 for 32-bit.
3. The third character represents a database type code. Please see the first 3 characters of your database driver dll
4. The fourth character is always a four, the major release version
5-6. The fifth and sixth character represent the database you are using. See chart below.
7-8. The seventh and eighth characters are the language code.

For example, the following chart breaks down a 32-bit application that uses PDB ODBC driver, in German, each appropriate selection marked with a *:

One character specification of DLL type
P = database DLL
One digit bit indicator
1 = 16-bit
3 = 32-bit
1 character database driver type identifier
D = PDB
Q = PDQ
R = PDIR
S = PDS
T = PDCT
1 digit version number (major) (4)
2 characters database identifier
AT = ACT

BD = BND

BE = BDE

BT = BTRV

D2 = DB22

GT = GUPTA

JT = JET

PX = PDX

QB = QBE

NW = NETW

OD = ODBC

OR = ORACLE

ST = SYB10

SY = SYBAS

XB = XBSE

2 characters language codes:

DE = German

FR = French

IT = Italian

JP = Japanese

Example

P 3 D 4 OD DE.DLL

DE	German Language
OD	ODBC database driver
4	Version 4.X
D	PDB database driver type (First three characters of your database driver)
3	
3	32-bit application
P	Database DLL

See Also

Foreign Language Exporting Requirements ■ 32 ■bit

Foreign Language File Requirements for Exporting

Crystal Reports 32■bit uses a naming convention for determining which resource file(s) you may need for an application in a language other than English, that exports. Each type of export and each type of destination requires a separate DLL.

For example, the following file represents a German 32■bit application that is exporting to Disk:

2 characters language code

DE = German
EN = English
FR = French
IT = Italian
JP = Japanese

2 characters export destination or database identifier

DK = Disk

MP = MAPI

VM = VIM

2 character format type identifier

CR = CR (Crystal Reports format)

DC = DOC (Word for DOS/WordPerfect)

DF = DIF

QP = Quattro Pro

RC = REC (Record style)

RT = RTF (Rich Text Format)

SV = SEPV (Char Separated Value)

TX = TXT (Text)

WD = WORDW (Word for Windows)

WK = WKS (Lotus 1■2
■3)
XL = XLS (Excel 2, 3, 4, 5)

1 digit version number (major) (4)

1 character database type identifier

D = Destination

F = Format

One digit bit indicator

1 = 16■bit

2 = 32■bit

One character specification of DLL type

X = export DLL

EXAMPLE:

X 2 D 4 DK DE.DLL

DE	German Language
DK	Export to Disk
4	Version 4.X
D	Destination
2	32■bit application

X Export DLL

applicAnd for each export type you wish to include, you would need X2F4??DE.DLL where ?? is the 2 letter code of the export format you wished to include.

Runtime File Requirements

File Requirements

As a registered user of Crystal Reports, you are entitled to distribute a runtime version of the Crystal Reports Print Engine with your applications at no charge. The following is an overview of the runtime requirements:

For all Applications using Crystal Reports:

The following files must be included on a distribution disk, regardless of the application it accompanies:

32bit file requirements:

File	Description
CRPE32.DLL	interface to the print engine - 32-bit■ 32■bit
MSVCRT20.DLL	Microsoft VC++ Run Time library
CRXLAT32.DLL	include only if using the ToWords functions
CO2C40??.DLL	?? = Language code
P2BBND.DLL	include only if binding Crystal Reports Custom Control to the Visual Basic Data Control
CTL3D32.DLL	Version 2 of 3 D control (needs to be in windows/system32 directory)
PG32.DLL	Include only if you have graphs/charts in your reports

The following files are required if you are doing the following in your application with Crystal Reports:

File	Description
CRYSTLxx.OCX	Using the Crystal ActiveX (Where xx is 16 or 32 [bits])
PG.DLL	Using Graphs in your reports
CO2C40EN.DLL	Using OLE objects in your reports (32■bit)
CRXLATE.DLL	Using the function ToWords (x, # of places) in your report
P2BBND.DLL	binding a Crystal Custom Control to a VB data control (32■bit)

Database Specific Requirements:

The following files are necessary for their respective databases. If your application will not be reporting on one or more of these databases, then do not include the files for those databases.

NOTE: You may use the File|Report Option menu option to see which P*.dll file you report is using (in the convert report box)

Access

ODBC

ODBC■ACCESS

ODBC Excel

ODBC ASCII

SQL Server

Export Specific Requirements

Where to install files

See Also

Crystal Reports Features List

ODBC Access Specific requirements

Include the following files if your application will be using ODBC Access:

NOTE: Be sure to check the requirements for the specific ODBC you are using.

File	Description
RED110.DLL	MS Jet DLL
MSJETDSP.DLL	MS Jet DLL
SIMADMIN.DLL	SIMBA ODBC DLL
SIMBA.DLL	SIMBA ODBC DLL

ODBC Specific Requirements

Include the following files if your application will be using ODBC:

NOTE: Be sure to check the requirements for the specific ODBC you are using.

32-bit file requirements

File (32-bit)	Description
ODBC32.DLL	Microsoft ODBC DLL
ODBCINT.DLL	Resource DLL
MFC30.DLL	MFC DLL
P2SODBC.DLL	Crystals ODBC
ODBC.INI*	ODBC INI file
ODBCINST.INI	Install INI file
CTL3D32.DLL	(NT Only)

*You must ship ODBC.INI files that are configured appropriately for the databases your application is using. You do this using the ODBC Admin program, ODBCADM.EXE. The ODBC.INI file must point to the location in which your ODBC drivers will be installed.

Also add the following the registry

```
\\HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI
\\HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI
```

ODBC Access 1.1 specific requirements

Include the following files if your application will be pointing to an ACCESS MDB file

File	Description
RED110.DLL	MS Jet DLL
MSJETDSP.DLL	MS Jet DLL
SIMADMIN.DLL	SIMBA ODBC DLL
SIMBA.DLL	SIMBA ODBC DLL

ODBC Access 2.0 32-bit specific requirements

File	Description
MSJTER32.DLL	ODBC Error DLL
MSJINT32.DLL	ODBC Error DLL
VBAJET32.DLL	VBA database engine
VBAR232.DLL	VBA DLL
VAEN232.OLB	VBA type library
ODBCTL32.DLL	ODBC Database engine
ODBCJT32.DLL	ODBC Database engine

MSJT2032.DLL ODBC Desktop driver DLL

Microsoft Access via DAO runtime file requirements

Access via DAO 32-bit specific requirements

Filename	Description
P2BDAO.DLL	CRW Physical Database DLL
P2CTDAO.DLL	CRW Physical Dictionary DLL
P2IRDAO.DLL	CRW Physical Directory DLL

ODBC Excel Specific Requirements

Include the following files if your application will be pointing to an Excel .xls file:

NOTE: Be sure to check the requirements for the specific ODBC you are using.

32-bit file requirements

File	Description
ODEXL32.DLL	ODBC Excel driver
MSXL2032.DLL	ODBC Excel ISAM
MSVCRT10.DLL	MS Visual C++ Runtime DLL
MSJTER32.DLL	ODBC Error DLL
MSJINT32.DLL	ODBC Error DLL
VBAJET32.DLL	VBA Database engine
VBAR232.DLL	VBA DLL
VAEN232.OLB	VBA type library
ODBCTL32.DLL	ODBC Database engine
ODBCJT32.DLL	ODBC Database engine
MSJT2032.DLL	ODBC Desktop driver DLL

ODBC ASCII Specific Requirements

Include the following files if your application will be pointing to an ASCII text file:

NOTE: Be sure to check the requirements for the specific ODBC version you are using.

32-bit file requirements

File	Description
ODTEXT32.DLL	ODBC Text driver
MSTX2032.DLL	ODBC text ISAM
MSVCT10.DLL	MS Visual C++ Runtime DLL
MSJTER32.DLL	ODBC Error DLL
MSJINT32.DLL	ODBC Error DLL
VBAJET32.DLL	VBA Database engine
VBAR232.DLL	VBA DLL
VAEN232.OLB	VBA type library
ODBCTL32.DLL	ODBC Database engine
ODBCJT32.DLL	ODBC Database engine
MSJT2032.DLL	ODBC Desktop driver DLL

Microsoft SQL Server Specific Requirements

Include the following files if your application will be reporting on SQL databases:

32-bit

File	Description
dbnmpntw.dll	SQL Server Driver file
SQLSRV32.DLL	SQL Server Driver file
MSVCRT20.DLL	Microsoft VC++ Run Time library

Export Specific Requirements

If your application will give your users the ability to export their reports, you must include files from the following list appropriate to the export options provided:

NOTE: Each format that is supported in 32-bit is the same as the 16-bit name with the exception that the second character is a 2. For example, the UXFCR.DLL 32-bit version is U2FCR.DLL.

NOTE: Some export formats are not supported in 32-bits. Please see the Features List for more information.

16-bit file requirements:

Format DLLs	Description
UXFCR.DLL	Crystal Reports Format 16-bit-bit
UXFDIF.DLL	DIF format
UXFDOC.DLL	Word for DOS and Word Perfect format
WORDDOS.XTD	Only required if exporting to Word for DOS
WPERFECT.XTD	Only required if exporting to Word Perfect
UXFQP.DLL	Quattro Pro
UXFREC.DLL	Record format
UXFRTF.DLL	Rich Text Format
UXFSEPV.DLL	Comma Separated Values Format
UXFTEXT.DLL	Text format
UXFWKS.DLL	Lotus 1-2-3 format
UXFWORDW.DLL	Word for windows format
UXFXLS.DLL	Excel format

Destination DLLs	
UXDDISK.DLL	Disk file destination
UXDMAPI.DLL	MAPI format (Microsoft mail)
UXDVIM.DLL	VIM format (cc: MAIL, Lotus Notes, WordPerfect Office, etc.

Click here for further information on foreign language runtime file requirements

Crystal Reports version with Microsoft Access

Microsoft Access was shipped as a stand-alone product and also shipped as the database engine with Visual Basic. There are three versions of consequence: 1.0. 1.1, and 2.0. Since 1.0 and 1.1 are similar when it comes to Crystal's interaction with them, we will just call them Access 1.x files.

PLEASE READ THIS:

Before we go any further, there are some bits of information that are required to get you to go to the proper set of file requirements for your application.

This applies to all versions of Access:

To find out whether you are using the Jet Engine or ODBC to connect to your database:

- If you created the report and selected the "Data File" button, you are using the Jet Engine.
- If you created the report and selected the "SQL / ODBC" button, you are using ODBC.

Access 1.x users:

If you have never had Access 2.0 on your machine and are using the MS Jet Engine, please continue with the section called "*Access 1.x through the Jet Engine*".

However, if you are using the Jet Engine and have Access 2.0 on your machine as well (assuming you have run the Access Compatibility Layer), please proceed to "*Access 2.0 through the Jet Engine*".

If you are using ODBC, please proceed to "*Access 1.x through ODBC*".

Access 2.0 users:

If you are using the Jet Engine (assuming that you have run the Access Compatibility Layer), please proceed to "Access 2.0 through the Jet Engine".

If you are using ODBC, please proceed to "Access 2.0 through ODBC".

NOTE: The following runtime files are 16-bit only. 32

-bit connectivity supported only via ODBC

Access 7.0

NOTE: Access 7.0 data sources are only supported in the 32-bit version of Crystal Reports.

FILENAME	**DESCRIPTION**
P2BDAO.DLL	CRW Physical Database DLL
P2CTDAO.DLL	CRW Physical Dictionary DLL
P2IRDAO.DLL	CRW Physical Directory DLL
DAO2532.TLB	
DAO3032.DLL	Microsoft Jet DAO Library
VBAJET32.DLL	Visual Basic for Applications Development Environment - Expression Service Loader
VEN2232.OLB	Visual Basic for Applications Object Library
MSJINT32.DLL	Microsoft Jet Database Engine International DLL
MSJTER32.DLL	Microsoft Jet Database Engine

	Error DLL
MSJT3032.DLL	Microsoft Jet Engine Library
VBA232.DLL	Visual Basic for Applications Development Environment
MSRD2X32.DLL	Microsoft ® Red ISAM

The following files are required if you are using one of the following external databases as an Access 7.0 attached table (Installable ISAM):

FILE	**DESCRIPTION**
MSPX3032.DLL	Paradox DLL
MSXB3032.DLL	dBASE (and FoxPro) DLL

To configure the datasource properly for Access 7.0, the following files must be registered:

DAO3032.DLL

MSRD2X32.DLL

MSXL3032.DLL

MSPX3032.DLL

MSTX3032.DLL

MSXB3032.DLL

Developer Contents

ActiveX Properties Index
Creating Bound Reports with the ActiveX control
Crystal Reports with Microsoft Access
Export Specific Requirements
Foreign Language File Requirements for Exporting
Foreign Language Runtime File Requirements
Microsoft Access via DAO
Microsoft SQL Server/Sybase SQL Server Specific Requirements
ODBC Access Specific requirements
ODBC ASCII File Requirements
ODBC Excel Specific Requirements
ODBC Specific Requirements
Runtime File Requirements
The Crystal ActiveX Control