

Crystal Reports 9

Optimizing Reports for the Web

Overview

This technical brief discusses how to take advantage of the performance enhancements in Crystal Reports (CR) 9 when designing reports. The suggestions described are particularly applicable to optimizing the performance of reports distributed over the Web.

Contents

INTRODUCTION	3
MAKING BETTER DESIGN CHOICES	3
<i>Upgrading older reports</i>	<i>3</i>
<i>Choosing between live and saved data</i>	<i>3</i>
Live data	3
Saved data	4
<i>Using summary reports</i>	<i>4</i>
<i>Using subreports</i>	<i>4</i>
On-demand subreports	4
Linked subreports	5
Linking tables instead of linking subreports	5
STREAMLINING YOUR REPORTING ENVIRONMENT	5
<i>Using table indexes</i>	<i>6</i>
<i>Improving table-linking choices</i>	<i>6</i>
Linking tables effectively	6
Using thread-safe database drivers	6
<i>Using stored procedures for faster processing</i>	<i>7</i>
USING ENHANCED RECORD SELECTION FORMULAS	7
<i>Pushing down record selection</i>	<i>7</i>
Example	8
<i>Record selection performance guidelines</i>	<i>8</i>
General	8
PC Databases	8
SQL Databases	9
<i>Tips for writing efficient record selection formulas</i>	<i>9</i>
Use the Formula Workshop	9
Use the Select Expert	9
Use a standard form for selection formulas	9

- Use SQL Expressions in selection formulas 10
- Use AND, OR, and NOT operators 10
- Incorporating parameter fields into record selection formulas.....11*
- Using SQL expressions where appropriate.....12*
- Key uses of SQL Expression fields..... 12
- IMPROVING GROUPING AND TOTALING 13**
- Performing grouping on server.....13*
- Using SQL Expressions for groups, sorts, and totals13*
- Using SQL Expressions for Case Logic.....13*
- Inserting summary and running total fields where possible 14*
- FINDING MORE INFORMATION 15**

Introduction

Whether you distribute your reports over a local network, a corporate Intranet, or the Web, you can use the powerful performance features in Crystal Reports 9 to quickly deliver reports.

The key strategies for optimizing web reports discussed in the following sections often bring substantial performance benefits. When you design new reports (or improve your old ones) in accordance with these strategies, reports run faster and consume fewer processing resources.

Making Better Design Choices

This section discusses design options and considerations to help improve the performance of your reports. When designing your reports, and especially when designing reports for the Web, the goal should be to drive the data that the end user will want to see. In other words, display summarized information making it easier to navigate the report quickly and then drill down to access additional data. In this way, web traffic and response times are minimized because only the data requested is transferred from the database server.

Upgrading older reports

The quickest way to help improve the performance of your old reports is to save them in the latest Crystal Reports format because the newest reporting engine is able to process previous versions of reports faster. To upgrade an older report, open it in Crystal Reports and select **Save** from the **File** menu.

CAUTION	Once a report is saved as a version 9 report, it cannot be opened by a previous version of Crystal Reports. If you need to keep an old report in its original file format, use the Save As command (instead of Save) and type a different name for the version 9 report.
----------------	---

Choosing between live and saved data

When reporting over the Web, choosing whether to use live or saved data is an important consideration.

Live data

Reports with live data enable you to immediately access information on the database server. However, you should consider whether or not you want users to constantly access the database server to refresh their report data. If the data does not constantly change, then those requests to the database do little more than increase network traffic and consume server resources. In such cases, you may prefer to use reports with saved data.

Saved data

Reports with saved data are useful when data is not being continually updated. When you navigate through reports with saved data, and drill down for details on columns or charts, you are not directly accessing the database server; instead, you are accessing the saved data. Consequently, reports with saved data help to minimize not only network traffic, but also the use of server resources.

To save a report with data:

1. On the **File** menu, select the **Save Data with Report** option.
2. Save the report.

Using summary reports

By hiding the **Details** section of a report that contains summary fields, you can view only the data that you wish to see. With the **Details** section hidden, you navigate through the Group tree first to locate the desired data. Then by drilling down on the report, you can request specific data.

To facilitate navigation in this way, you first need to group the data and insert the summary fields that you wish to include on your report. Once you have grouped and summarized your report data, hide the **Details** section (and any other large report sections) as follows:

1. Open your grouped and summarized report in the Crystal Reports Designer.
2. On the **Report** menu, click **Section Expert**.
3. In the **Sections** list, click **Details**.
4. On the **Common** tab, select the **Hide (Drill-Down OK)** check box.
5. Click **OK**.

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

Using subreports

A subreport is an excellent way to include additional data in a main report. There are two types of subreports from which to choose, and the choice of which one to use depends on how the report data is used.

On-demand subreports

If a main report has a section that contains a large number of records that relatively few users view, then consider placing that section into an on-demand subreport. An on-demand subreport appears as a hyperlink in the main report. When you open the main report, no data is retrieved for the on-demand subreport until you click the hyperlink. Therefore, fewer database server resources are used. Users who need to see the extra data click the hyperlink to

view the subreport; users who don't need to see the extra data won't have to download it from the database server.

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

Many report objects—such as large cross-tabs, OLAP grids, advanced charts, and maps—are ideal candidates to include in on-demand subreports. You could also place such report objects in a hidden **Details** section of a report that uses the **Perform Grouping On Server** option. When you do this, the database server performs the majority of the processing, and only a subset of the records is transferred from the server to the local computer (other records are retrieved when you drill down to a hidden section).

Linked subreports

When a subreport is linked, Crystal Reports coordinates the data in the subreport with the matching records in the primary report. When the main report is opened, the subreport records are processed and displayed.

In some cases, placing regular linked subreports in the **Details** section of a main report may hinder performance—especially when your main report contains many records. This is because you are creating a separate subreport for each record, and a separate query must be run for each database record in the main report. As an alternative, consider using linked on-demand subreports in the **Details** section of your main report.

Linking tables instead of linking subreports

Whenever possible link database tables on the **Links** tab of the **Database Expert**, rather than linking subreports. That is, link subreports, that are not on-demand subreports, to the main report. Since each subreport is run as a separate report, linked tables often have a performance advantage.

Streamlining Your Reporting Environment

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

Using table indexes

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

Improving table-linking choices

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

There are many other specific issues to consider when linking tables. These additional considerations, however, are largely contingent upon your reporting environment. In other words, the steps to optimal table-linking performance depend upon your database type, the possibility of indexing tables, and the join type required between tables.

In the majority of reporting scenarios, the recommendations that follow will help to ensure that your tables are linked for improved, if not optimal, performance.

Linking tables effectively

To link tables effectively, follow these steps:

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

Using thread-safe database drivers

If you share and refresh reports over the web, use thread-safe database drivers whenever possible. Since the Crystal Report Engine supports multiple threads, when you make multiple data requests through thread-safe database drivers, the Report Engine can process all of the requests simultaneously. As the result, you are able to view your reports sooner.

The ODBC driver Crdb_odbc.dll is thread-safe for Microsoft SQL Server and Sybase Adaptive Server Enterprise. For a list of supported databases (and versions), refer to Knowledge Base article c2011814 on our [support site](#).

Furthermore, the following native database drivers are thread-safe (see the Release Notes on the Crystal Reports CD for updates):

- Crdb_ado.dll (OLE DB)
- Crdb_cdo.dll (Crystal Data Object)
- Crdb_com.dll (COM data provider)
- Crdb_dao.dll (Microsoft Access)
- Crdb_dataset.dll (DataSet provider)
- Crdb_fielddef.dll (Field Definitions)
- Crdb_filesystem.dll (File System data)
- Crdb_olap.dll (OLAP)
- Crdb_oracle.dll (Oracle)
- Crdb_p2sdb2.dll (IBM DB2)

Using stored procedures for faster processing

A stored procedure is a compiled SQL program, consisting of one or more SQL statements, that resides and runs on your database server. The advantages of using stored procedures are most noticeable when reporting off large sets of data, or when running reports that have complex calculations. In such cases, you should use a predefined stored procedure that performs the complex work for you on the database server.

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

Using Enhanced Record Selection Formulas

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

Pushing down record selection

Crystal Reports analyzes your record selection formula and generates an SQL query from it. This SQL query is then processed by the database server, which subsequently returns the resulting records to Crystal Reports. Crystal Reports then evaluates the record selection formula locally against each of the records retrieved from the database to calculate the set of records for the report.

Unnecessary records are eliminated at two stages: on the database by the SQL query, and within Crystal Reports by the record selection formula. By designing your record selection formula effectively, you can off-load much of the processing to the database server, thus eliminating unnecessary records before returning them to Crystal Reports. This is commonly referred to as "pushing record selection down to the database server."

Ensure that on the **File** menu in the Crystal Reports Designer, select **Report Options**, and select the **Use Indexes or Server for Speed** box.

Example

The example in this section demonstrates the benefits of writing record selection formulas that can be pushed down to the database server.

In the **Orders** table of the Xtreme Sample Database, there are 2192 records, of which 181 have order dates prior to 2001. Suppose you want to report on only those records. On the one hand, you could use this record selection formula:

```
Year ({Orders.Order Date}) < 2001
```

The SQL query generated returns all 2,192 records to the Crystal Report Engine, and then the record selection formula subsequently reduces this to 181 records. To see this, click **Show SQL Query** on the **Database** menu and notice that the SQL query has no WHERE clause. This is because the Crystal Report Engine is not able to push down the **Year** function in the WHERE clause

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

```
{Orders.Order Date} < #Jan 1, 2001#
```

This second formula, however, can be executed on (that is, pushed down to) the database server. The SQL query generated returns only 181 records to the Crystal Report Engine, which does not have to eliminate any other records. Click **Show SQL Query** on the **Database** menu and notice that the resulting SQL query has a WHERE clause.

As this example shows, the report processing speed improves when you enhance your record selection formula. While both formulas generate the same report, the second formula uses the power of the database server to process the data.

Record selection performance guidelines

Consider the following guidelines when creating record selection formulas:

General

- In record selection formulas, avoid data type conversions on fields that are not parameter fields. For example, avoid using the **ToText** function to convert a numeric to a string database field.
- You are able to push down record selection formulas that use constant expressions provided that the formula is written in a standard form. See [Use a standard form for selection formulas](#) for further details.

PC Databases

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

SQL Databases

- You can push down record selection on indexed or non-indexed fields.
- Use indexed fields for better report performance.
- You can push down selection formulas that contain AND clauses as well as OR clauses.
- Record selection formulas containing some types of embedded formulas can be pushed down. See [Use SQL Expressions in selection formulas](#) for further information.
- Incorporate SQL Expression fields to push down formula calculations necessary for the record selection.

NOTE

If you are uncertain as to whether your record selection is being pushed down, in the Crystal Reports Designer, click **Show SQL Query** on the **Database** menu. If the full SQL query is there (including the WHERE clause), then record selection is being pushed down to the database.

Tips for writing efficient record selection formulas

This section assumes that you are familiar with the **Select Expert** and that you are reporting off an SQL database.

Use the Formula Workshop

You can create more types of record selection formulas using the **Formula Workshop** than is possible with the **Select Expert**. To open the **Formula Workshop** to modify record selection, on the **Report** menu, click **Selection Formulas**, and then click **Record**.

Use the Select Expert

Any record selection formula that you create entirely with the **Select Expert** can be pushed down.

Use a standard form for selection formulas

Any selection formula that is of the form: *[DataBaseField][SupportedOperator][ConstantOrParameterExpression]* can be pushed down to the database server, where:

- *DataBaseField* is a database field
- *SupportedOperator* is any of =, <>, <, <=, >, >=, StartsWith, Like or In.
- *ConstantOrParameterExpression* is any expression that involves constant values, operators, functions, and parameter fields. It cannot involve variables, control structures, or fields other than parameter fields. By their very definition, constant and parameter expressions can be evaluated without accessing the database. A constant or parameter expression can

evaluate to a simple value, a range value, an array value, or an array of range values. For example:

```
{?number parameter} - 3
Year ({?run date})
CurrentDate + 5
DateDiff ("q", CurrentDate, CDate("Jan 1, 1996"))
Month (Maximum ({?date range parameter}) + 15)
["Canada", "Mexico", "USA", {?enter a country}]
1000 To 5000
[5000 To 10000, 20000 To 30000, 50000 To 60000]
```

The following is an example of a record selection formula that uses the standard form:

```
{Orders.Order Date} >= CurrentDate - 3
```

An expression that just contains a Boolean field (without the operator and constant parts) can also be pushed down to the database server:

```
{Orders.Shipped}
Not {Orders.Shipped}
```

Use SQL Expressions in selection formulas

SqlExpression SupportedOperator ConstantOrParameterExpression can be pushed down.

For example, the selection formula `{@ExtendedPrice} > 1000` cannot be pushed down if `{@ExtendedPrice} = (Quantity * Price)`. However, if the formula `@ExtendedPrice` is replaced with the equivalent SQL Expression, then the record selection formula will be pushed down.

Use AND, OR, and NOT operators

When using multiple expressions, separate them with AND, OR, and NOT operators, where appropriate. For example:

```
{Orders.Order ID} < Minimum({?number range}) Or
{Orders.Order Amount} >= 1000
(IsNull({Customer.Region}) Or
{Customer.Region} = "BC") And
{Customer.Last Year's Sales} > 2000
```

Incorporating parameter fields into record selection formulas

Instead of displaying all report records each time the report is opened, incorporate parameter fields into your record selection formulas. By prompting users to specify the records that they want to view, you also decrease the amount of data transferred from the database server.

You can add a parameter field to your record selection formula by using the **Select Expert** or the **Formula Workshop**. When using the **Formula Workshop**, treat the parameter field as you would any other field.

The following steps demonstrate how to create and then add a parameter field to a record selection formula using the **Select Expert**:

Creating a parameter field

1. Open the sample report **Group By Intervals.rpt**, which is located in the C:\Program Files\Crystal Decisions\Crystal Reports 9\Samples\En\Reports\Feature Examples folder.

NOTE	If the Field Explorer window does not appear on the left hand side of the Crystal Reports Designer, then on the View menu, click Field Explorer .
-------------	--

2. In the **Field Explorer** window, right-click **Parameter Fields** and select **New**. The **Create Parameter Field** dialog box appears.
3. In the **Name** text box, type "SalesQuota".
4. In the **Prompting text** text box, type "What was last year's sales quota?"
5. Click the **Value type** list and select **Number**.
6. Ensure that the **Discrete value(s)** option is selected, and then click **OK**.

Adding a parameter field to a record selection formula

1. On the **Report** menu, click **Select Expert**. The **Choose Field** dialog box appears.
2. Click the **Customer.Last Year's Sales** field, and then click **OK**. The **Select Expert** dialog box appears.
3. On the **Customer.Last Year's Sales** tab, in the list, click **is greater than**. A new list appears.
4. In this new list, click **{?SalesQuota}**.

NOTE	To view the new record selection formula, click the Show Formula button.
-------------	---

5. Click **OK**. The **Enter Parameter Values** dialog box appears.

6. In the **Discrete Value** text box, enter a numeric value representing last year's sales quota.
7. Click **OK**. The **Change in Record Selection Formula Detected** dialog box appears.
8. Click either the **Use Saved Data** or **Refresh Data** button.

You have now added a parameter field to a record selection formula. The report now displays only the customers for which the value of Last Year's Sales exceeds the numeric value that you specified.

Using SQL expressions where appropriate

Crystal Reports includes its own formula language, which is much more powerful than standard Structured Query Language (SQL). Both Crystal Syntax and Basic Syntax allow you to enhance and hone your formulas in ways that are either difficult or impossible with SQL.

In certain circumstances, however, creating and using SQL Expression fields can speed up the processing of your reports. SQL expressions are like formulas, but they are written in Structured Query Language (SQL). They are useful in optimizing report performance because the tasks they execute are normally performed on the database server (as opposed to a regular formula, which is sometimes executed on the local computer).

Key uses of SQL Expression fields

To improve report processing speeds, avoid using formulas (whether Crystal or Basic syntax) in record selection formulas. Instead, replace the original formula with an equivalent SQL Expression field, and then incorporate that SQL Expression field into your record selection formula. This greatly improves the chances of the processing being done on the database server.

In addition, avoid sorting, grouping, or totaling on a formula field (whether Crystal or Basic syntax). Instead, replace the original formula field with an equivalent SQL Expression field, and then sort, group, or total on that SQL Expression Field. Again, this will greatly improve the chances of the processing being done on the database server.

Finally, if your database supports Case Logic, and your report needs to summarize an **If-Then-Else** formula calculation, replace the formula with an SQL Expression field. In such cases, SQL Expression fields enable the Crystal Report Engine to perform the grouping on the database server.

Improving grouping and totaling

Performing grouping on server

You can reduce the amount of data transferred from the database server by using the **Perform Grouping on Server** option. With this option, much of the data processing is off-loaded to the database server, and only a subset of the data is read initially. Detail data is returned from the database only when you drill down on a report. Server-side processing works only for sorted and grouped reports that are based on SQL data sources.

To enable server-side processing:

1. On the **File** menu, click **Report Options**. The **Report Options** dialog box appears.
2. Select the **Perform Grouping on Server** check box. This check box is inactive if **Use Indexes or Server for Speed** is not selected.

NOTE	When Use Indexes or Server for Speed is selected, you can select the Perform Grouping On Server option from the Database menu.
-------------	---

3. Click **OK**.

Using SQL Expressions for groups, sorts, and totals

For reports using Perform Grouping on Server, avoid sorting, grouping, or totaling on a formula field (whether Crystal or Basic syntax). Instead, replace the original formula field with an equivalent SQL Expression Field, and then sort, group, or total on the SQL Expression field. This will greatly improve the chances of the processing being done on the server.

Using SQL Expressions for Case Logic

If your database supports Case Logic, and your report needs to summarize an If-Then-Else formula calculation, replace the formula with an SQL Expression field, which enables Crystal Reports to perform the grouping on the server.

For instance, suppose that you're reporting off of an MS SQL Server 7 database, which supports Case Logic. You need to include an If-Then-Else calculation in your report, and you need to summarize that calculation for each group in the report. By performing the calculation with an SQL Expression field of the following form, you use your database's ability to process Case Logic:

```
CASE DatabaseTable."DatabaseField"  
WHEN 'SpecifiedValue' THEN Calculation1  
ELSE Calculation2  
END
```

If a record's *DatabaseField* value is equal to *SpecifiedValue*, then *Calculation1* is performed; for all other records, *Calculation2* is performed. By incorporating the SQL Expression field, you take advantage of your database server's ability to process Case Logic. The grouping consequently takes place on the server, even when you summarize the SQL Expression field elsewhere in the report.

NOTE	The SQL syntax in this example is specific to MS SQL Server 7. You may need to refer to your database documentation or consult with your Database Administrator to determine the syntax appropriate to your database.
-------------	---

Inserting summary and running total fields where possible

Where possible, avoid creating formulas containing global variables to calculate summaries or running totals.

Instead, create summaries by clicking the Insert menu and then selecting the appropriate command (subtotal, grand total, or summary). Create running total fields by opening the **Field Explorer**, right-clicking **Running Total Fields**, and then clicking **New**.

DISCLAIMER	<p>The third party products discussed in this white paper were not fully tested in conjunction with Crystal Enterprise prior to its release. Officially supported Crystal Enterprise platforms are listed in the text file Platforms.txt found in the root of the Crystal Enterprise CD as well as in the Knowledge Base article c2009003 (search for this article number at http://support.crystaldecisions.com/kbase), which is more current.</p> <p>The information in this document is provided as a courtesy to assist our customers with the configuration of our product in conjunction with these third party platforms.</p> <p>In the event issues arise with an unsupported configuration, there is no escalation support; however, they will be considered during the development of the next generation of our product.</p>
-------------------	---

Finding More Information

For more information, refer to the section “Designing Optimized Web Reports” in the Crystal Reports 9 help file.

For more information and resources, refer to the product documentation and visit the support area of the web site at: www.businessobjects.com

► www.businessobjects.com

The Business Objects product and technology are protected by US patent numbers 5,555,403; 6,247,008; 6,578,027; 6,490,593; and 6,289,352. The Business Objects logo, the Business Objects tagline, BusinessObjects, BusinessObjects Broadcast Agent, BusinessQuery, Crystal Analysis, Crystal Analysis Holos, Crystal Applications, Crystal Enterprise, Crystal Info, Crystal Reports, Rapid Mart, and WebIntelligence are trademarks or registered trademarks of Business Objects SA in the United States and/or other countries. Various product and service names referenced herein may be trademarks of Business Objects SA. All other company, product, or brand names mentioned herein, may be the trademarks of their respective owners. Specifications subject to change without notice. Not responsible for errors or omissions. Copyright © 2004 Business Objects SA. All rights reserved.