

# Crystal Reports 9 and later

## Evaluation Times and the Multi-Pass Reporting Model

---

### Overview

This paper is written for Crystal Reports (CR) 9 and later, and describes the steps CR takes during the multi-pass reporting model, evaluation time functions, and the relationship between record selection and Crystal Reports' first pass over the data, among other topics.

### Contents

<b>INTRODUCTION .....</b>	<b>2</b>
<b>THE MULTI-PASS REPORT PROCESSING MODEL .....</b>	<b>2</b>
<i>First Pre-pass &amp; BeforeReadingRecords Formulas .....</i>	<i>2</i>
<i>First Pass &amp; WhileReadingRecords Formulas .....</i>	<i>2</i>
<i>Second pre-pass .....</i>	<i>3</i>
<i>Second-pass &amp; WhilePrintingRecords formulas .....</i>	<i>3</i>
Examples of second-pass functions: .....	4
<i>Third pass .....</i>	<i>5</i>
<b>SECTION EVALUATION WITHIN THE MULTI-PASS REPORTING MODEL .....</b>	<b>5</b>
<i>Subreport evaluation .....</i>	<i>6</i>
<i>Special Note on Shared and Stored variables .....</i>	<i>6</i>
<i>Page-on-Demand Architecture .....</i>	<i>6</i>
<i>How multi-pass reporting affects page headers and footers .....</i>	<i>6</i>
<b>SPECIFYING WHICH PASS A FORMULA OCCURS IN .....</b>	<b>7</b>
<i>BeforeReadingRecords .....</i>	<i>7</i>
<i>WhileReadingRecords .....</i>	<i>7</i>
<i>WhilePrintingRecords .....</i>	<i>8</i>
Incorrect example .....	8
Correct example .....	8
<i>EvaluateAfter .....</i>	<i>9</i>
<b>THE RELATIONSHIP BETWEEN RECORD SELECTION AND THE FIRST PASS .....</b>	<b>9</b>
<i>Scenario 1 – Server-side record selection .....</i>	<i>10</i>
<i>Scenario 2 – Record selection performed partially on server and on client .....</i>	<i>10</i>
<i>Scenario 3 – Client-side record selection .....</i>	<i>11</i>
<i>Second-Pass Issues .....</i>	<i>11</i>
<b>FINDING MORE INFORMATION .....</b>	<b>12</b>

## Introduction

Crystal Reports uses a multi-pass report processing model to generate reports. A pass is a process that Crystal Reports uses each time the data is read or manipulated. Depending on the complexity of the report, Crystal Reports may make one, two, or three passes over the data before the report displays onscreen. This multi-pass report processing model enables CR to perform more complex calculations, such as percentages of subtotals, in reports.

For you to take full advantage of this multi-pass model, it is important that you understand the order in which report elements are evaluated when previewing reports. Evaluation time is a concept that controls many behaviors in Crystal Reports. For example, with fields based only on formulas that have already been evaluated, Crystal Reports can create graphs, crosstabs, maps or running totals.

Understanding the order in which Crystal Reports reads and manipulates data allows you to better plan your report. You will know what elements can and cannot be used together to give the desired results. For example, formulas that evaluate at the end of the multi-pass evaluation process (known as “Print-Time” or **WhilePrintingRecords** formulas) cannot be used when creating maps, graphs, or cross-tabs.

## The Multi-Pass Report Processing Model

A pass refers to the process of reading records from the database and then manipulating them according to how you have designed the report. Crystal Reports performs three main passes before the report is previewed.

- In the first pass, Crystal Reports reads the data, record by record, from the database and creates basic summaries. These summaries are stored in a temporary file.
- In the second pass, Crystal Reports reads this temporary file and performs more complex manipulations on the saved data.
- In the third pass, Crystal Reports reads the total page count.

### First Pre-pass & BeforeReadingRecords Formulas

Before making the first pass through the database, Crystal Reports evaluates any constant formulas in the report. Constant formulas are those that have a constant value for the entire report. For example:

2 \* 50

### First Pass & WhileReadingRecords Formulas

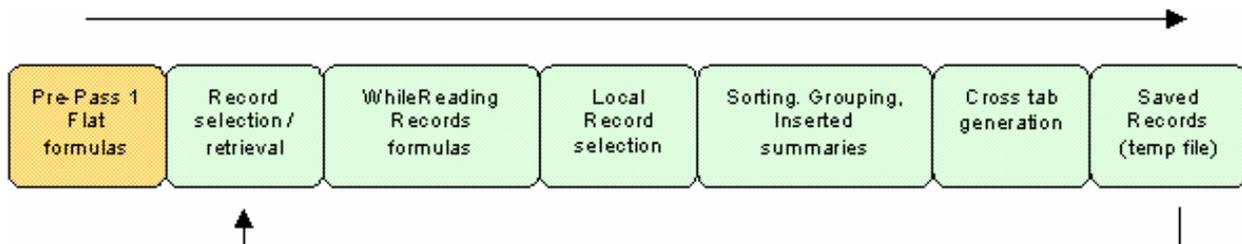
Crystal Reports reads the records from the database during the first pass. During the first pass the following occurs:

- Record retrieval. Where possible, record selection and sorting are pushed down to the database in this step.

- Evaluation of recurring formulas. Recurring formulas are those that contain database fields but do not contain references to subtotals or summary information.
- Local application of the record selection. If the record selection is too complex to be pushed down to the database, it is applied by CR in this step.
- Sorting, grouping, and totaling. CR sorts the records, separates them into groups, and calculates the subtotals and summaries needed for each group.
- Storage of data. After totaling, all the records and totals are stored both in memory and to temporary (.TMP) files. CR does not read the database again, but instead uses this saved data during all subsequent processing.

Referring to records other than the current record (for example, by using the **Previous** or **Next** functions), or including summary functions, forces a formula to occur during the second pass.

The entire first-pass process occurs for each individual record as CR retrieves it from the database.



**Figure 1. Diagram of the first pass**

During the first pass, CR performs the steps above for each record included in the report, one record at a time.

## Second pre-pass

**Heirarchical Grouping** and **TopN/BottomN** sorting occurs here. These actions are based on the summaries created during the first pass. No additional records are read during the second pre-pass.

## Second-pass & WhilePrintingRecords formulas

This pass is where more complex manipulation of the data occurs. At this point CR reads the TMP file containing the saved records from the first pass instead of reading directly from the database a second time.

Pages in a report are formatted on demand. This means CR will not format a page until it is requested by the user or until it is required for the total page count (calculated during the third pass). During the second pass the following will occur:

- Group selection formula.
- Running totals.

- Calculation of formulas marked **WhilePrintingRecords**. **WhilePrintingRecords** formulas contain references to subtotals or summary information and are also known as Print-Time formulas. See examples of second-pass functions below.
- Cross-tab, chart, and map generation.
- OLAP grids.
- Subreports.
- Generation of pages on demand.

<b>NOTE</b>	Although subreports appear in the second pass, you can use on-demand subreports to ensure that your main report remains a single-pass report. With on-demand subreports CR must still make a second pass through the data; however, this pass does not begin until you drill down on the subreport. Consequently, you can increase the performance of reports that contain subreports by using on-demand subreports.
-------------	--

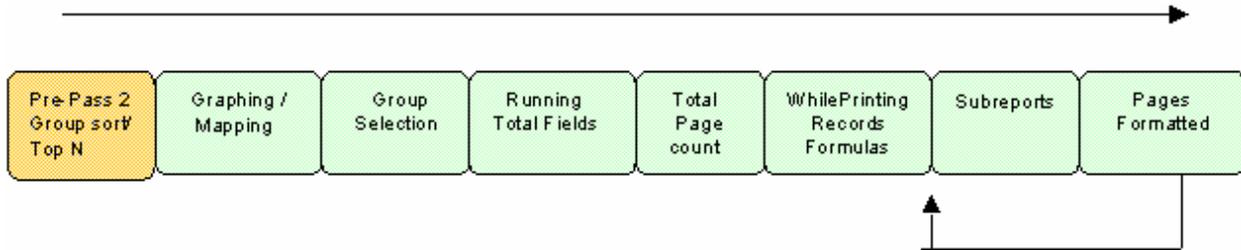
As discussed previously, including certain functions in a formula automatically forces the formula to evaluate during the **WhilePrintingRecords** phase because the function requires all records to have been read before it can calculate correctly.

#### Examples of second-pass functions:

<b>Any Inserted summary field</b>	<b>DistinctCount</b>
<b>WhilePrintingRecords</b>	<b>TotalPageCount</b>
<b>Next</b>	<b>RecordNumber</b>
<b>NextIsNull</b>	<b>GroupNumber</b>
<b>Previous</b>	<b>PageNumber</b>
<b>PreviousIsNull</b>	<b>TotalPageCount</b>
<b>Average</b>	<b>PageNofM</b>
<b>Sum</b>	<b>OnFirstRecord</b>
<b>Count</b>	<b>OnLastRecord</b>

A rule of thumb is to determine whether you can apply a function to a single record, or if more than one record is required for the function to calculate. If the answer is more than one record, this function will occur during the **WhilePrintingRecords** stage.

<b>NOTE</b>	Subtotals, grand totals, and summaries may appear to be incorrect if the report has a group selection formula. This occurs because the grand totals and summaries are calculated during the first pass but the group selection formula filters the data again in the second pass. Running total fields can be used instead of summaries to total data in reports with a group selection formula.
-------------	--



**Figure 2. Diagram of the second pass**

These second-pass steps are evaluated for each section in the report, except for subreports and page formatting. Instead, these are evaluated as you scroll through each new page. This is known as *page-on-demand architecture*.

### Third pass

The total page count is determined during the third, final pass. This applies to reports that use the total page count, or Page N of M, special fields.

## Section Evaluation within the Multi-Pass Reporting Model

In addition to the multi-pass reporting concept, Crystal Reports is described as a *banded report writer*, meaning that CR writes the report vertically. The sections are evaluated in this order:

1. Report Header(s)
2. Page Header(s)
3. Group Header(s)
4. Details
5. Group Footer(s)
6. Report Footer(s)
7. Page Footer(s)

The second pass generally occurs for each section, primarily referring to graphing/mapping, running total fields, **WhilePrintingRecords** formulas and subreports. Objects are usually read from the top of the section to the bottom and then left to right.

However, when two **WhilePrintingRecords** formulas are inserted into a single section, one above another, the first formula inserted will be the first evaluated. If you later reverse the order by moving the formulas around, the evaluation will not change, as they have been locked into their respective evaluation-time positions. To change this order, remove the formulas from the report and re-insert in them in the new order.

An example of how this banded, top-down report structure can affect reports is conditional formatting. You can perform conditional formatting on the Details section based on the value of a formula in the Group Header; CR will have already determined the value of the formula in the Group Header by the time it

reaches the Details section. However, you cannot conditionally format the Details section based on the same formula if that formula is in the Group Footer; because while still evaluating the Details section, CR would not yet know what is contained in the Group Footer.

## Subreport evaluation

Subreports are among the last objects to be evaluated in the second pass for each section. When CR encounters the subreport object, it generates this “report within a report” with its own multi-pass structure. The subreport is evaluated using the same multi-pass method as the main report. The main report then continues and completes the final formatting of the report.

## Special Note on Shared and Stored variables

Evaluation time and subreport evaluation are especially important when using shared variables to pass information from a subreport to the main report. A formula in the main report, which retrieves the values that are stored or calculated in the subreport, should be placed in a separate section that is beneath the section containing that subreport.

This is because the subreport will not have been recognized at the time that the formula is executed. Instead, the formula must be placed in the section that follows; for example, Group Header A for the subreport and Group Header B for the formula. Once the subreport has finished evaluating and Group Header A is completed, the next section is evaluated in the second pass and the formula is executed.

## Page-on-Demand Architecture

Page formatting, like subreports, is not generated for the entire report immediately upon clicking the Preview button.

For example, previewing a 20-page report will initially show only page 1. If there are subreports on the other pages they have not been generated, nor has the formatting for those pages been completed. As you scroll through the report, the subreports and the formatting are executed only as you come across them. This is known as “page-on-demand architecture” and is used to minimize processing time requirements.

You can change this “page-on-demand architecture” by using the **TotalPageCount** function or **PageNoFM** field somewhere in the report. In order to accurately determine the number of pages in the report, all subreports must be evaluated and general formatting must be completed.

## How multi-pass reporting affects page headers and footers

While you can conditionally suppress the contents of the page header and footer sections, you cannot conditionally suppress the space allotted for the sections.

CR reserves space for Page Header and Page Footer sections before the contents of the sections are evaluated. If the conditional suppression criterion is met, the contents of a section do not appear, although the physical space that it would

have occupied on the report still appears. This is because the page footers and headers are created before any section formatting is done.

## Specifying which pass a formula occurs in

There is a set of functions in the formula editor in the **Evaluation Time** folder that can be used to force a specified formula to be evaluated at a certain time. These functions are:

- **BeforeReadingRecords;**
- **WhileReadingRecords;**
- **WhilePrintingRecords;**
- **EvaluateAfter();**

To ensure that a formula is evaluated at the correct stage in the evaluation process, we recommend including the appropriate evaluation time function at the beginning of the formula, followed by a semi-colon.

Without such a declaration, the Report Designer cannot tell when to evaluate the formula in the evaluation process.

The following sections provide more detail about each of the Evaluation Time functions.

### BeforeReadingRecords

The **BeforeReadingRecords** function is rarely used since it applies to flat formulas only. There are a few circumstances that require this function. For example, if you enter a numeric parameter that is multiplied by a given value, the **BeforeReadingRecords** function is used:

```
@Flat formula //This is the formula name
BeforeReadingRecords;
{?NumericParam} * 1.45
```

### WhileReadingRecords

The **WhileReadingRecords** function forces CR to evaluate a formula during the first pass. If the formula contains a function that must occur in the second pass, CR's **Formula Editor** displays the error message "This function cannot be used because it must be evaluated later." In this case, you can either:

- add **WhilePrintingRecords** to the beginning of the formula  
or
- remove the second-pass function from the formula

## WhilePrintingRecords

You can force CR to evaluate any formula during the second pass by using the **WhilePrintingRecords** function. However, once this function is included in a formula, it has all of the limitations of the second-pass formulas.

Generally speaking, **WhilePrintingRecords** should be the default for most formulas unless you intend to insert summaries on that field.

An example of when to use the **WhilePrintingRecords** function is when declaring variables that are to be used in more than one formula on the report, or used in different sections. This is a very common issue when calculating running totals using formulas rather than using the Running Total Expert.

### Incorrect example

The formula below is created and placed in the details section of the report:

```
@SetVariable           //This is the formula name
//increments variable X by the value of {Table.Field}
NumberVar X;
X := X + {Table.Field}
```

**@SetVariable** is meant to display a total that accumulates for each record. While the formula **@SetVariable** will return a value for each record, it will not remember that value when CR moves on to evaluate **@SetVariable** for the next record in the report. In other words, the variable X will not accumulate as shown here:

	Table.Field	@SetVariable
<b>Record 1</b>	2	2
<b>Record 2</b>	8	8
<b>Record 3</b>	15	15

Since there are neither second-pass functions nor summaries in this formula, CR considers it a recurring formula and evaluates it during the first pass on a record-by-record basis. However, to accumulate values, all the records first need to be read by CR. In other words, the formula must be made into a **WhilePrintingRecords** formula:

### Correct example

```
//increments variable X by the value of {Table.Field}
WhilePrintingRecords;
NumberVar X;
X := X + {Table.Field}
```

The formula now returns the cumulative value of the variable X on each detail line of the report:

	<b>Table.Field</b>	<b>@SetVariable</b>
<b>Record 1</b>	2	2
<b>Record 2</b>	8	10
<b>Record 3</b>	15	35

## EvaluateAfter

This function forces CR to evaluate one formula after another formula. This is used when it is critical for the calculation of one formula depends on the results of another formula.

```

@SetVariable           //This is the formula name
NumberVar X;
X := {table.field} * {table.field2}

@UseVariable           //This is the formula name
EvaluateAfter({@SetVariable});
NumberVar X;
X * {table.field3}

```

<b>CAUTION</b>	<p>If you have a series of formulas with @Formula2 in the report footer, and you also have another formula that is evaluated earlier and includes:</p> <pre>EvaluateAfter (@Formula2)</pre> <p>If this happens, @Formula2 will be forced to evaluate twice, potentially causing incorrect results.</p> <p>If you do have formulas that must be evaluated sequentially within a single section, it is strongly recommended that the first formula use the appropriate evaluation time function. The formulas that follow should use the EvaluateAfter function to sequence the formulas so that they are evaluated in the correct order.</p>
----------------	---

## The Relationship between Record Selection and the First Pass

From a report processing perspective, CR ideally needs to parse the entire record selection formula for a report into SQL so that it can be passed to the database when the report is previewed. This enables a report's record selection to be processed on the server, so CR ends up reading only the records that need to be included in the report. By reducing the number of records read by CR, server-side processing speeds up the time it takes to preview a report.

There are three possible ways for CR to handle record selection formulas:

- Translate the entire formula and pass it to the database so that the record selection actually occurs on the database (known as server-side processing).
- Translate part of the selection formula and pass it to the database for record selection, while the remainder of the selection formula is performed by CR (known as client-side processing)
- Not translate any of the formula, so that the entire selection process occurs on the client side.

<b>NOTE</b>	For more detail on how record selection formulas affect report-processing speed, you can search for the technical brief <i>Performance Considerations when Creating Record Selection Formulas</i> (scr_recsele.pdf) at <a href="http://support.businessobjects.com/search">http://support.businessobjects.com/search</a> .
-------------	--

The following three examples illustrate how each type of record selection formula as described above determines the steps CR must take while reading the database during the first pass.

## Scenario 1 – Server-side record selection

Record selection formula is `{table.field} = 1000`

1. Evaluates flat formulas.
2. Translates record selection for the database (SQL statement for ODBC /SQL databases, to native language for other database types. Also includes “Group by” on the server if this option is enabled). Sends record request.
3. Evaluates **WhileReadingRecords** formulas.
4. Creates grouping, sorting and inserted totals.
5. Adds values to cross-tab.
6. Adds values to temporary (.TMP) file.

CR returns to step 1 until all records have been read.

## Scenario 2 – Record selection performed partially on server and on client

Record selection is `{table.field2} = 1000 and {table.field} = @FirstPassFormula`

1. Evaluates flat formulas.
2. Translates the portion of record selection that can be translated (`{table.field2} = 1000`) for the database (SQL statement for ODBC /SQL databases, to native language for other database types. Includes “Group by” on the server if this option is enabled). Sends record request.
3. Evaluates **WhileReadingRecords** formulas.
4. Performs remainder of record selection formula only after `@FirstPassFormula` has been processed.

5. Creates grouping, sorting and inserted summaries.
6. Adds values to cross-tab.
7. Adds values to temporary (.TMP) file.

CR returns to step 1 until all records have been read.

## Scenario 3 – Client-side record selection

Record selection is `{table.field} = @FirstPassFormula`

1. Evaluates flat formulas.
2. Retrieves *all* records from database.
3. Evaluates **WhileReadingRecords** formulas.
4. Performs record selection only after processing `@FirstPassFormula`.
5. Creates grouping, sorting and inserted summaries.
6. Adds values to cross-tab.
7. Adds values to temporary (.TMP) file.

CR returns to step 1 until all records have been read.

Referring to a **WhileReadingRecords** formula in a record selection formula causes CR to request all (as in Scenario 3) or a large portion (as in Scenario 2) of the database to be returned. This can slow performance tremendously. For this reason, it is strongly recommended to not refer to any formulas within the record selection. Even in regular formulas (that is, non-record selection formulas), referring to other formulas is not recommended as it will increase processing time requirements.

Notice also that in all three examples, inserted summaries, grouping and sorting all occur after **WhileReadingRecords** formulas. This explains why you can insert group summaries and grand totals on **WhileReadingRecords** formulas, but not on **WhilePrintingRecords** formulas. For the same reason, you cannot group or perform sorts on **WhilePrintingRecords** formulas.

Rather than inserting a summary or grand total on **WhilePrintingRecords** formulas, create a running total.

For more information on running totals, go to <http://support.businessobjects.com/search> and search for `scr_runningtotal.zip`, which is a document called *Everything You Need to Know about Running Totals*.

## Second-Pass Issues

Because they occur prior to the **WhilePrintingRecords** stage of the second pass, the following features cannot be performed on second-pass formulas (refer to Figure 2):

- Mapping
- Running Total fields
- Grouping

- Sorting
- Inserted summaries

The alternative in such cases is to perform the calculation on the database side using, for example, stored procedures. These calculations are then brought in as simple fields and any of the above procedures can be carried out.

<b>DISCLAIMER</b>	<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 folder of the Crystal Enterprise CD as well as in the Knowledge Base article c2009003 (search for this article number at <a href="http://support.businessobjects.com/search">http://support.businessobjects.com/search</a>), 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 and resources, refer to the product documentation and visit the support area of the web site at: [www.businessobjects.com](http://www.businessobjects.com) .

▶ [www.businessobjects.com](http://www.businessobjects.com)

Business Objects owns the following U.S. patents, which may cover products that are offered and sold by Business Objects: 5,555,403, 6,247,008 B1, 6,578,027 B2, 6,490,593 and 6,289,352. Business Objects, the Business Objects logo, Crystal Reports, and Crystal Enterprise are trademarks or registered trademarks of Business Objects SA or its affiliated companies in the United States and other countries. All other names mentioned herein may be trademarks of their respective owners. Product specifications and program conditions are subject to change without notice. Copyright © 2004 Business Objects. All rights reserved.