**BUSINESS OBJECTS**

# Crystal Enterprise Report Application Server 9

Understanding Performance

## Overview

It is important to understand the architecture behind how the Crystal Enterprise Report Application Server (CE RAS) 9 processes user requests in order to create more efficient applications to access, analyze, and share your data. This document discusses CE RAS performance and provides recommendations to help you design more scalable applications.

## Contents

# Introduction

The Crystal Enterprise Report Application Server (CE RAS) 9 is a powerful product that gives you as a developer the ability to customize the structure and presentation of a report for viewing by your end users. This document discusses CE RAS performance, helping you to make better design choices when designing your applications.

.

# Architecture Overview

To understand the performance of the Crystal Enterprise Report Application Server (CE RAS), it is helpful to begin with an architecture overview. CE RAS is a client/server system that is composed of two basic components: the RAS server, and RAS SDK that provides an interface to the server. The web server manages communication between the web browser and the RAS SDK (see Figure 1).
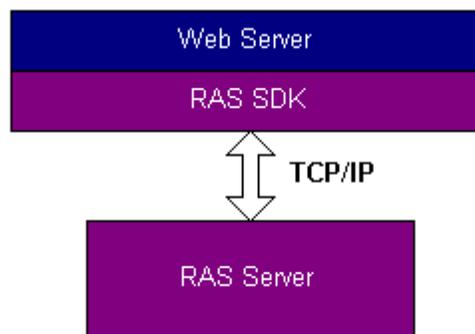


**Figure 1: Architecture Overview of CE RAS**

The RAS SDK includes a number of libraries that allow you to build a web report design application. These libraries are used in your server side code (ASP, CSP, JSP, or Java servlets) and provide an interface to the RAS Server. In this sense, the SDK forms the client part of the client/server system.

The RAS Server provides the services for building and customizing reports. Your application uses the RAS SDK to communicate with the server. The server handles report documents using the Crystal Reports Print Engine (CRPE).

# Handling User Requests

To understand how Crystal Enterprise Report Application Server (CE RAS) handles user requests, it is helpful to trace from start to finish a simple request to open a report. While the following Active Server Pages (ASP) example uses the COM SDK, the Java SDK uses similar methods:

```
<%

Dim clientDoc

Set clientDoc =
CreateObject("CrystalReports.ReportClientDocument")

clientDoc.Open ("C:\Program Files\Crystal Decisions\Report
Application Server 9\Reports\Chart.rpt")

%>
```

After a new **ReportClientDocument** object is assigned to the *clientDoc* variable, the VBScript engine calls the **Open** method and transfers control to the RAS SDK. The **ReportClientDocument.Open** method is found in Clientdoc.dll, and translates the string parameter of the report name into a format that the RAS Server can understand.

Since the RAS SDK and RAS Server use XML to communicate with each other, any SDK on any platform is able to communicate with the RAS Server. In this example, the RAS SDK translates the **Open** method into the following XML text:

```
<Request InstanceID="2" AgentID=";2144009048;"
RequestID="103" Options="0" Priority="5" Time="2002-7-
5T18:16:23.40Z">

<Detail>

<CrystalAnalysis.FetchDataDocumentRequest
xmlns="http://www.crystaldecisions.com/report"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="CrystalAnalysis.FetchDataDocumentRequest"
version="2">

  <ID>103</ID>

  <Options>0</Options>

<DocUID xsi:type="CrystalReports.PropertyBag" id="1"
version="2">

<Property>

  <Name>isremotedoc</Name>

  <Value VariantType="Boolean">-1</Value>

  </Property>

<Property>
```

```
<Name>path</Name>

<Value VariantType="String">C:\Program Files\Crystal
Decisions\Report Application Server
9\Reports\Chart.rpt</Value>

</Property>

</DocUID>

</CrystalAnalysis.FetchDataDocumentRequest>

</Detail>

</Request>
```

After the RAS Server receives the request, it creates a print job to process the report. Each instance of an opened report object has an associated print job ID to allow the RAS Server to distinguish between different reports opened by different users.

After the report document is opened, it is serialized to XML and sent back through TCP/IP to the RAS SDK on the web server. The document is cached by the RAS SDK so that future requests to open this report are processed more efficiently. The document is also cached on the server, as discussed later in this document. Now that the request is completed, additional commands can be processed.

# Performance Implications of Different Features

## Report Design

Beginning with Crystal Reports (CR) 9, you can view potential performance bottlenecks in your report design by using the Performance Information tool. To access this tool, on the **Report** menu in CR 9, click **Performance Information**.

The Crystal Reports Online Help also contains a section called "Designing Optimized Web Reports" that describes how to create efficient reports. By following those recommendations, you can improve the performance of your application.

## Database

Opening a report does not automatically retrieve data from the database. If the report contains saved data, the database connection is not used unless the saved data becomes invalidated and discarded.

Saved data is regenerated only when a parameter value is changed, or a new field is added to or referenced in the report. Removing the last field of a database table in a report also discards the saved data so that the table linking can be re-optimized. Modifying the report definition or look and feel of the report does not refresh the data.

2/16/2004 1:11 PM                                       Page 4

## Database Drivers

The major ODBC database drivers are multithreaded and scale well. Refer to the Crystal Reports documentation to ensure that your driver is threadsafe.

## Number of Records in the Report

The number of records that the report needs to process can significantly impact performance. While the Crystal Enterprise Report Application Server (CE RAS) is capable of handling very large data sets, adding filtering and sorting can potentially increase processing time, depending on how your report is constructed.

For example, you apply a filter to retrieve only 1,000 of the 400,000 records saved with a report. However, it would be more efficient to create a report that retrieves those 1,000 records from the database server using a well constructed SQL statement that includes records based on the value in an indexed field.

A report with several hundred thousand records consumes a great deal of memory, thus slowing down many operations such as sorting and filtering. However, viewing the report is still fairly fast because the first page is appears as soon as it is processed. The formatting and processing of the remaining pages continues in the background.

## Including Total Page Count

If you include the special field **Page N of M** or **Total Page Count**, the entire report must be processed before this value can be calculated. It is recommended that you avoid using these special fields unless your report is very small or you absolutely require them.

## Maximum Server Load

There is a practical maximum of 50 print jobs per RAS Server. If your RAS Server consistently approaches or exceeds this limit, it is recommended that you start another RAS Server instance on a different CPU.

Crystal Enterprise Report Application Server (CE RAS) version 9 introduced advanced job sharing – report instances are only opened when necessary. If you and another user open the same report (and the report is not modified) for viewing, you will share the same report instance.

Job sharing is completely transparent because if you then attempt to modify the report, the print job will split off into a new print job ID. The other user is unaffected. This caching of the report allows one RAS server to support more users than would normally be possible.

| NOTE | Reports with subreports are NOT be cached. |
|------|--------------------------------------------|

## Maps

Map rendering is a single threaded operation that does not scale well. Although maps are supported, carefully consider the effect a map has on the performance of your report when deciding whether or not to include one.

## Stateless Viewer

The Crystal Reports DHML viewers introduced in version 9 do not have to be stored in session variables on the web server. However, this scenario only works for read-only reports. If you wish to modify the report or refresh the data source, you must store the viewer and report objects in session variables because the stateless viewer re-opens the **ReportClientDocument** object each time a different page is viewed.

Since the RAS SDK caches the **ReportClientDocument** object, viewing the report in stateless mode is nearly as fast as it would be if the report object had been stored in session.

## Report Templates

If you are applying the same template to multiple reports, it is recommended to open the report once and cache it, since the template only needs to be read-only.

| NOTE | To highlight text in a PDF document for copying and pasting code, click the **Text Select Tool** toolbar button in Adobe Acrobat.  This procedure applies to Adobe Acrobat 4.0 and 5.0. |
| --- | --- |

| DISCLAIMER | 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 http://support.businessobjects.com/search), which is more current. 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. 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. |
| --- | --- |

# 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

▶ www.businessobjects.com