



# Crystal Reports 9 Developer Technologies

## Comparing the Report Designer Component (RDC) and the Report Application Server (RAS)

### Overview

For application development in Version 9, the Crystal Reports engine can be leveraged through two distinct reporting choices– the Crystal Reports Report Designer Component (RDC) and the new Crystal Enterprise Report Application Server (RAS). As a general guideline, RAS, Crystal Decisions’ newest and most powerful object model, is the *recommended* solution for any web application development projects whereas the RDC is now *recommended*<sup>1</sup> for thick-client applications<sup>2</sup>. For customers who are already using the RDC, the decision between which of these two technologies to use should be carefully considered. The purpose of this document is to compare and contrast the RDC and RAS and to demonstrate the advantages RAS offers for web application development.

### Contents

- WHAT IS THE REPORT APPLICATION SERVER? ..... 2
- WHAT IS THE REPORT DESIGNER COMPONENT? ..... 2
- REPORT MAINTENANCE ..... 2
  - Repository*..... 2
  - Custom Templates*..... 3
  - Custom Functions*..... 3
- APPLICATION DEVELOPMENT ..... 3
- PROCESSING PERFORMANCE ..... 3
  - User Requests* ..... 3
  - Caching*..... 4
  - Caching*..... 4
  - Component versus Standalone Server* ..... 4
  - Server Communication* ..... 4
  - Session Variables* ..... 5
- END USER INTERACTIVITY ..... 5
  - Viewers* ..... 5
  - End User Report Creation* ..... 5
  - Web Publishing* ..... 6
  - Printing* ..... 6
  - Miscellaneous Features*..... 6
- EXTENSIBILITY ..... 6
- MIGRATING RDC APPLICATIONS TO RAS ..... 7

1 From a technical perspective, the RDC can still be used for web applications. However, as demonstrated in this document, RAS is a more powerful, feature-rich solution and strongly recommended for all web application projects.  
 2 Any application in which all report processing occurs on the end user’s machine, not on a server.

## What is the Report Application Server?

The Report Application Server (RAS) is a client-server system that enables report creation, processing and manipulation in a multi-tier environment. It is composed of two basic components: the RAS server and a Software Developers Kit (SDK) which provides an interface to the server. Because RAS has been designed as two separate components, report processing can be offloaded from the web server.

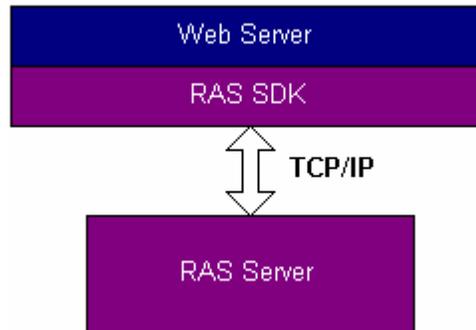


Figure 1: Simple Architecture View for RAS

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

The RAS server provides the services for designing, viewing, processing and customizing reports. Custom-built applications can use the SDK to communicate with the server. The server processes report documents using the Crystal Reports Print Engine (CRPE).

## What is the Report Designer Component?

The RDC is an ActiveX designer object for Microsoft Visual Basic (VB) 5 and 6 users. Basically, it's a VB "plug-in" that allows developers to create, open, design, customize, and run reports within the VB Integrated Development Environment (IDE).

The RDC object model includes more than 850 properties and methods to provide developers with tight control over their application's reporting systems. It consists of the RDC Design Time (Report Designer, CRAXDUI.DLL and CRAXDDRT.DLL), the RDC Runtime (the Automation Server, CPEAUT32.DLL), and the Crystal Reports Viewer, CRVIEWER.DLL).

## Report Maintenance

Crystal Reports 9 introduces several new productivity tools designed to reduce the amount of time spent on creating and maintaining reports. From an application development perspective, only applications based on RAS can take advantage of these new tools for more efficient report maintenance.

### Repository

The Repository is a central library where common report objects can be stored for reuse across multiple reports and single point updating. Applications which use reports that have been designed based on repository objects – images, text objects, custom functions and SQL commands – can be maintained in a very efficient manner. If an object in the repository is changed, you can setup your

application to automatically refresh the reports from the repository or develop a button for ad hoc repository refresh. With the RDC, any changes to reports within your application must be manually applied to each individual report. This can often be a mundane, time-consuming task.

### Custom Templates

In Crystal Reports 9, you can create a set of custom templates to automatically apply a common look-and-feel to all the reports in your application. This replaces the need to manually format individual reports. Custom templates can also include objects stored in the Repository. This can result in more efficient report maintenance and updating process for RAS applications.

Without the use of Custom Templates, as is the case for RDC applications, any report formatting – even common corporate standards or boilerplate requirements - must be manually applied to each individual report.

### Custom Functions

With Custom Functions, you can extract the business logic from formulas so that they can be used across multiple reports. Similar to Custom Templates, Custom Functions can be stored in the Repository for faster formula creation and efficient maintenance. This efficient updating process is available through RAS.

## Application Development

The RDC is COM-based and can be used with any COM-compliant development platform including Visual Basic, Visual C++, ASP.NET and Delphi. RAS can be used in COM, Java and .NET development platforms. You can use JavaScript or VBScript to write your web application.

Both the RDC and RAS can display output in the form of a formatted report page. However, RAS also provides the ability to extract the raw data from a report. This allows you to leverage the information contained in a report for multiple purposes. For example, you can build a World Wide Sales Report and then programmatically extract data from it, transform this data into XML for sharing with a partner. You can also use this concept to let end users search for specific data within a report and export the resulting results grid to other formats for sharing with others.

## Processing Performance

### User Requests

In Version 9 both the RDC and RAS are technically restricted to accept three simultaneous requests - for example, 1<sup>st</sup> page viewing, exporting, drill-down, printing, etc. However, there are significant differences in terms of how requests are managed by each technology. These differences can impact overall system performance and scalability.

With the Advanced Edition RDC, if the server is busy processing three requests and a 4<sup>th</sup> request is received, the request will simply keep retrying every few seconds until the browser times out. By contrast, with RAS, the 4<sup>th</sup>, 5<sup>th</sup>, 6<sup>th</sup> etc. requests are queued in a logical fashion. There is no preset timeout and subsequent requests will not fail.

<b>NOTE</b>	To support unlimited user requests, additional Processor Licenses may be purchased.
-------------	---

## Caching

RAS can also cache reports. This can substantially reduce the amount of processing time required for subsequent report requests. If a user opens a report that already has been opened by another user and has not been modified, the same report instance will be shared between the users. Also, if a user selects to view a report and subsequently requests to view the next page, caching enables the second page to be displayed without re-running the entire report. The effect is completely transparent to the end user; if one of the users sharing the instance attempts an action that will modify the report, the report job is cloned thereby leaving the other users unaffected. From a developer's perspective, this process is also transparent and does not require additional coding.

As a result of caching, a single RAS server can typically support more users than would normally be possible. Without caching, as with the RDC, the entire report must be re-run each time a subsequent request is made.

## Component versus Standalone Server

The RDC is a component that must be installed on the web server and run under the constraints of Microsoft Internet Information Services (IIS) or ASP. As both IIS and ASP are tuned more for web pages with basic querying, customers with intensive report processing requirements will experience performance degradation when trying to use the RDC in this environment.

To contrast, RAS is a web-focused API. This enables the report processing to be offloaded from the web server which results in better overall performance. By running RAS on a standalone server, performance is improved because report processing is no longer subject to the constraints of the web server process.

<b>NOTE</b>	Although not a requirement, it is recommended that RAS run on a separate server to allow for better scalability and functional partitioning between machines.
-------------	---

RAS may also be allocated to run on a specific number of processors on a server. This enables developers to create the most cost-effective solution for the performance they require. Because the RDC is a component which must be installed on the Web Server, the RDC cannot be pinned to a specific processor. If processor licensing is required, the cost can be significant for RDC applications.

With RAS, you may only install one copy per copy of Developer or Advanced Edition purchased. If you purchase two copies of Crystal Reports, you may install two separate RAS but cannot link them together in a web farm to increase the performance of a single application.

With each purchase of Crystal Reports Developer and Advanced Editions, the RDC may be installed on multiple servers as long as you do not connect them together to scale your application beyond the included 3 request limit.

## Server Communication

The API provided in the RDC is "chatty". This means that each action – for example, setting parameters or fonts - incurs an individual command back to the engine. This can lead to inefficiencies. By contrast, RAS operates as a "quiet" API. This means that several operations can be combined into a single server operation.

## Session Variables

The RDC requires the use of ASP session variables. This means that a data object is stored in memory for each person actively using the system. As the user load increases, this can result in inefficient memory usage and scalability. There may also be complexities in running session variables in a web farm. By contrast, RAS does not require the use of a session variable.

## End User Interactivity

### Viewers

The RDC includes a basic HTML Report Page Viewer. This viewer renders a page-by-page view of a report. Users can navigate using hyperlinks, zoom, highlight etc.

RAS includes a set of robust server-side report viewer controls for more control over how a report is displayed within an application and how end users can interact with it:

**Basic HTML Report Page viewer** (COM, Java and .NET) includes the same basic functionality as provided with the RDC

**Advanced DHTML viewer** (COM and Java) – includes customizable Boolean search capabilities to enable end users to perform conditional searches on the data within the report; search results can also be exported to Word, Excel and HTML with dynamic linking back to the original report; also includes out-of-the-box thin-client printing capabilities (via streaming PDF files to the client)

**Report Part Viewer** (COM, Java and .NET) – allows you to display individual report parts from existing reports in your application (rather than the entire page of a report)

**Mobile Report Part Viewer** (COM, Java and .NET) – allows you to provide access to key report parts from existing reports via wireless devices such as cellular phones and PDAs

<b>NOTE</b>	The above section focuses on differences between the viewers. Other viewers which are available in both technologies are not outlined in this paper.
-------------	--

## End User Report Creation

Many web applications today require some form of end user report modification and creation capabilities. The licensing and use of this functionality varies by which technology your application uses.

To provide end users with report creation capabilities using the RDC, the RDC Report Creation API License must be purchased separately. One license is required for the developer to access this API set within the IDE and an additional per named user license must be purchased for each user that the application is redistributed or internally deployed to.

The RAS report creation and modification API set is included with the new Crystal Reports Advanced Edition. Applications which use these RAS APIs can then be deployed internally within the developers' organization at no additional costs. For runtime rights to these APIs, a copy of Advanced must be purchased for each customer redistributed. Developers may also acquire volume licenses from Crystal Decisions' OEM department.

## Web Publishing

The RDC does not provide an out-of-the-box solution for web report access and viewing. RAS includes a web sample, ePortfolio Lite, to address this requirement. Once RAS has been installed on a server, reports created in the Crystal Reports designer can automatically be saved to this location for immediate end user access.

## Printing

Using the RDC, you can programmatically set up printing directly to a printer or to a file. This is a good model for thick-client applications. In a web environment, since the web server normally does not have access to all required printers in an organization, a more efficient model is required. Using RAS, initiating a print automatically exports the report to the client in PDF format so that the end user can execute the print.

## Miscellaneous Features

Currently there are a few RDC reporting features that are not yet supported in RAS. For example, although RAS can process reports which contain cross-tabs and images, end users will not be able to add or modify these objects via the web.

Subreports functionality is also limited in the version 9 release of RAS. With subreports in RAS, only the database location and logon can be set and parameters can be passed to parameterfields. Other subreport features such as the ability to programmatically add or change and existing item in a subreport are not currently available in RAS but will be supported in future releases.

## Extensibility

RAS shares the same object model as Crystal Enterprise Standard and Professional Editions. This enables developers to seamlessly extend their applications in the future to take advantage of other Crystal Enterprise services including:

- Scheduling
- User, report and data security
- Report and instance management
- Automatic clustering, load balancing, and fail-over
- Web-based system administration and configuration
- Analytic and ad hoc reporting
- Automated distribution to various formats and destinations

Only a few lines of coding are required to alter a RAS application to take advantage of these additional services.

To contrast, the RDC does not share the same object model as the Crystal Enterprise family. This means that an application designed using the RDC will need to be completely rewritten in order to leverage additional Crystal services. This can prove to be expensive and resource intensive.

## Migrating RDC Applications to RAS

Although there is initial work involved in converting an application from the RDC to RAS, there are also significant benefits as outlined in this paper. A variety of sample applications are available at [www.crystaldecisions.com/devzone](http://www.crystaldecisions.com/devzone) to help guide you through the migration process.

\*\*\*\*\*

The information contained in this document represents the best current view of Crystal Decisions on the issues discussed as of the date of publication, but should not be interpreted to be a commitment on the part of Crystal Decisions or a guarantee as to the accuracy of any information presented.

This document is for informational purposes only and is not intended as a guarantee or warranty of any kind. The information is provided AS IS. CRYSTAL DECISIONS MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT. Crystal Decisions will not be liable for any damages (direct, indirect or consequential) to you for your consumption, use or reliance on the information provided herein.

© Copyright 2002 Crystal Decisions, Inc. All rights reserved. Crystal Reports, Crystal Enterprise, and Crystal Decisions are the trademarks or registered trademarks of Crystal Decisions, Inc. All other trademarks referenced are the property of their respective owners.

Specifications and product offerings subject to change without notice.