



# Crystal Decisions Developer Technologies

## Comparing and contrasting the .NET Object Model, the Report Application Server Object Model, and the Crystal Enterprise Object Model.

---

### Overview

For Web application development using the Microsoft .NET Platform, Crystal Decisions offers three increasingly advanced reporting object models which developers can utilize:

1. The Crystal Reports for Visual Studio .NET (.NET) object model: bundled in Microsoft Visual Studio .NET and in Crystal Reports 9 Developer and Advanced Editions,
2. The new Report Application Server (RAS) object model: bundled in Crystal Reports 9 Advanced and Developer Editions.
3. The Crystal Enterprise (CE) object model: provide as a standalone offering included with Crystal Enterprise 9 and made available through a new .NET SDK.

The purpose of this document is to compare and contrast the .NET, RAS, and CE object models and to provide direction as to when each object model should be considered for usage over the other available object models.

### Contents

WHAT IS THE .NET OBJECT MODEL? .....	2
WHAT IS THE REPORT APPLICATION SERVER OBJECT MODEL? .....	2
WHAT IS THE CRYSTAL ENTERPRISE OBJECT MODEL? .....	2
APPLICATION DEVELOPMENT .....	3
PROCESSING PERFORMANCE.....	4
<i>User Requests</i> .....	4
<i>Component versus Standalone Server</i> .....	4
<i>Report Component vs. Report Server Communication</i> .....	5
REPORT CREATION API .....	5
EXTENSIBILITY .....	5
MIGRATING .NET APPLICATIONS TO RAS.....	6
APPENDIX A - FEATURE COMPARISON BETWEEN CRYSTAL REPORTS FOR VISUAL STUDIO .NET AND CRYSTAL REPORTS 9 ...	7

## What is the .NET Object Model?

The .NET object model is included with Crystal Reports for Visual Studio .NET and Crystal Reports 9 Developer and Advanced Editions.

The .NET object model was originally designed and developed as an integrated component bundled directly into Microsoft's Visual Studio .NET products. The bundled technology is based on a subset of Crystal Reports 9 features, including an integrated report designer as well as a report and viewer object models. .

Crystal Reports 9, the feature-rich upgrade to Crystal Reports for Visual Studio .NET, includes the above .NET components plus many additional new features. In addition, the .NET object model has been updated with numerous enhancements. While the focus of this paper is on comparing the .NET object model to the RAS object model, we have included a feature comparison of Crystal Reports for Visual Studio .NET and Crystal Reports 9 as [Appendix A](#).

## What is the Report Application Server Object Model?

The Report Application Server (RAS) is a new and powerful object model. 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 Development 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.

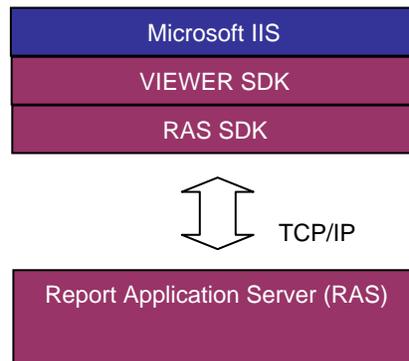


Figure 1: Simple Architecture View for Report Application Server

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

RAS provides the services for designing, viewing, processing and customizing reports. Custom-built applications can use the SDKs to communicate with the server. RAS is available with Crystal Reports 9 Developer Edition and Advanced Edition, as well as with Crystal Enterprise 9.

## What is the Crystal Enterprise Object Model?

The Crystal Enterprise 9 object model is the most advanced enterprise reporting, analysis and web delivery object model available to organizations that have embraced the .NET platform.. Crystal Enterprise provides a robust server-side reporting platform that includes the RAS object model. While Crystal Enterprise allows organizations to purchase an out-of-the-box platform, it equally offers them

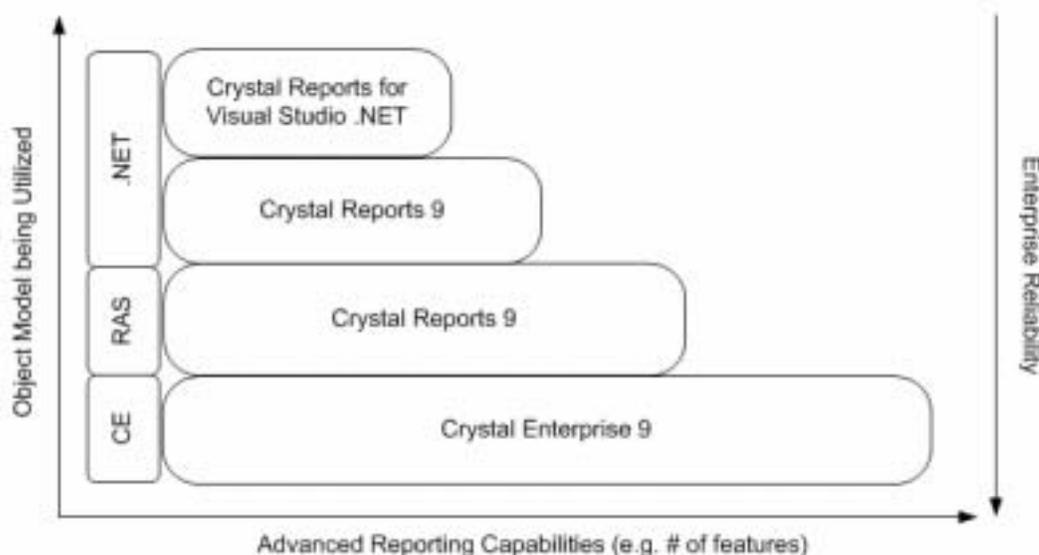
the ability to funnel development resources into highly customizing the front-end(s) that communicate with the Crystal Enterprise back-end. This is achieved by leveraging the Crystal Enterprise .NET SDK. Imagine being able to add new features or enhancing existing features to meet key requirements versus needing to build entire features. For example, Crystal Enterprise provides report scheduling out-of-the-box so that core reports can be run at any desired time, provided in any desired format (e.g. Excel, PDF, XML, Word, etc.), and delivered to virtually any desired location (e.g. Email account, FTP site, printer, etc.).

## Application Development

Both the .NET object model and the RAS object model can display output in the form of a formatted report page. They allow the ability to open reports, save reports, change groupings, and do such things as pass parameters, albeit using different syntax. However, RAS adds the ability to do report creation and modification on the fly at runtime, allowing users to self-serve – to customize their view of enterprise data. It provides the ability to extract raw data from a report. This allows you to repurpose information contained in reports. 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 results grid to other formats for sharing with others.

While the .NET and RAS object models define report formatting and data connectivity, the CE object model is focused on centralized, enterprise-wide report management. The CE object model defines how reports are scheduled, how report “snapshots” are stored, authorization and authentication system management, etc.

Perhaps the best way to illustrate the difference between the .NET object model, RAS object model, and CE object model is to provide a simplified diagram of what additional features and increased enterprise reliability exist with the three object models.



As a general guideline:

- The .NET object model is recommended for small workgroup projects that have relatively straight forward reporting requirements – e.g. smaller reports, fewer end users requesting reports, etc. This technology is a component of the application and scales with additional hardware – scales up with additional processors and scales out with additional servers.

- The RAS object model is recommended for medium-sized departmental applications that need to offload report processing and/or want to allow end-users to personalize their view of data by creating and modifying reports on the fly at runtime.
- The CE object model is recommended for mission-critical medium to enterprise-scale applications. This class of applications process volumes of data from disparate data sources, and presents this information to users for decision support. Typically, the volume of reporting requires precise scheduling and load balancing to ensure that the enterprise information infrastructure is not overloaded. These applications need to deliver information in a multitude of digital formats to a multitude of end user devices. Security, fault-tolerance, system management and auditing are all key requirements.

## Processing Performance

### User Requests

In Crystal Reports version 9, both .NET 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 Crystal Reports for Visual Studio .NET, requests beyond the three request threshold will be retried, until either one of the three in-progress requests completes or the browser times out. RAS, in contrast, will queue the 4<sup>th</sup>, 5<sup>th</sup>, 6<sup>th</sup>, etc. requests. There is no preset timeout and subsequent requests will not fail. RAS also includes, out of the box, the capability to cache reports, to minimize unnecessary database round-trips. This improves the performance of the server, relative to the Crystal Reports for Visual Studio .NET component.

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

### Component versus Standalone Server

.NET is a report component that is installed as an integral part of your web application, atop Microsoft Internet Information Services (IIS). Since the .NET report component is purpose-built for use in small-scale workgroup applications, customers with intensive report processing requirements may experience unacceptable system throughput. The lack of queuing or caching capability in the .NET report component makes it unsuitable for enterprise-scale information distribution. Another variable here is the CPU load of your web application. Since the .NET report component is an integral part of your web application, it consumes the same resources as your application.

To contrast, RAS has been architected for the web. Report processing is offloaded to a separate application tier resulting 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 application 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 application tiers.
-------------	--

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 .NET object model must be installed on the Web Server, it cannot be pinned to a specific processor. If

processor licensing is required, the cost is per processor for every processor on which the application is installed.

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

With each purchase of Crystal Reports Developer and Advanced Editions, the .NET report component may be installed on multiple servers as long as you do not connect them together in a web farm or web garden architecture to scale your application beyond the included 3 request limit – unless web farm/web garden licensing is purchased from Crystal Decisions.

### Report Component vs. Report Server Communication

The .NET report Component is “chatty”. Each transaction – for example, setting parameters or fonts – incurs an additional round trip to the component. This can lead to inefficiencies if reports are complex (e.g. accessing large amounts of data, connecting to multiple databases, performing advanced commands, etc.). In contrast, RAS buffers requests, and processes transactions in batches. Several operations can be combined into a single server operation, thus reducing the inefficiencies.

### Report Creation API

Many web applications today require some form of dynamic report creation capabilities. Allowing end-users to self-serve has become commonplace. The .NET report component does not include report creation APIs. In contrast, RAS includes a complete report creation and modification, available 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 to which the RAS application is redistributed to. Developers may also acquire volume licenses from Crystal Decisions’ OEM department.

### Extensibility

Although available as a standalone product, the RAS object model is an integral component of the Crystal Enterprise object model. This enables developers who have implemented RAS applications to seamlessly migrate 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.

In contrast, the .NET report object model does not share the same object model as the Crystal Enterprise family. This means that an application designed using the .NET report Component will need

to be completely rewritten in order to leverage additional Crystal services. This can prove to be expensive and resource intensive, so an organization should consider future requirements in their development planning taking place today.

## Migrating .NET Applications to RAS

Although there is initial work involved in converting an application from the .NET report object model 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 2003 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.



## Appendix A - Feature Comparison between Crystal Reports for Visual Studio .NET and Crystal Reports 9

	Crystal Reports® for Visual Studio® .NET	Crystal Reports 9
<b>Data sources</b>		
ODBC, OLEDB	✓	✓
User-defined ADO.NET data source	✓	✓
Oracle®, IBM® DB2®, Sybase®, Informix®		✓
XML		✓
User-defined COM data source		✓
Log files		✓
<b>Exporting (New!)</b>		
PDF	✓	✓
HTML	✓	✓
Microsoft® Word / RTF	✓	✓
Microsoft Excel	✓	✓
Microsoft Excel (data only)		✓
XML <sup>1</sup>		✓
Lotus® <sup>1</sup>		✓

Text/CSV <sup>1</sup>		✓
<b>Web delivery</b>		
XML Report Web Services	✓	✓
Crystal Enterprise Report Application Server		✓
<b>Productivity</b>		
Object Repository for component reuse and single-point updating		✓
Custom Functions for sharing formulas across multiple reports		✓
Standalone report designer		✓
Unicode support	✓	✓
Report sharing via Microsoft Smart Tags / Office XP		✓
Report Creation API (for report modification by end users)		✓
Microsoft Excel and Access Add-ins		✓
<b>Viewing</b>		
Report Parts viewer		✓
Mobile viewer (for report access via WML phones, RIM and iPAQ devices)		✓
Boolean search		✓
Ad hoc report creation		✓
Printing and exporting from Web Form Viewer		✓
<b>Charts (New!)</b>		

Gantt		✓
Bar	✓	✓
Line	✓	✓
Area	✓	✓
Pie	✓	✓
Doughnut	✓	✓
3D riser	✓	✓
3D surface	✓	✓
XY scatter	✓	✓
Radar	✓	✓
Bubble	✓	✓
Stock	✓	✓
Numeric axis	✓	✓
Gauge		✓
<b>Report types</b>		
OLAP		✓
Customizable Templates		✓

1. Available through use of RAS Object Model only; not available using .NET object model