



Crystal Reports 10 and .NET

Application Deployment

Overview

This document discusses the options for deploying Visual Studio .NET applications built using Crystal Reports 10 for Visual Studio .NET. Both merge modules and interactive install packages are discussed.

Contents

| | |
|--|-----------|
| INTRODUCTION | 2 |
| DEPLOYING REPORTS IN VISUAL STUDIO .NET APPLICATIONS | 2 |
| OVERVIEW OF DEPLOYMENT SCENARIOS | 3 |
| DEPLOYING USING THE MERGE MODULES..... | 3 |
| <i>Report Files (.RPT).....</i> | <i>3</i> |
| <i>Common Language Runtime.....</i> | <i>3</i> |
| <i>Crystal Reports 10 Merge Modules</i> | <i>3</i> |
| CrystalReports10_NET_EmbeddedReporting.msm | 4 |
| CrystalReports10_NET_RemoteReporting.msm | 4 |
| CrystalReports10_NET_WebServiceReporting.msm | 4 |
| CrystalReports10_Maps.msm | 5 |
| DEPLOYING USING THE SERVER INSTALL (INTERACTIVE INSTALL) | 5 |
| USAGE SCENARIOS | 5 |
| ASP.NET Web Application – Single Server..... | 5 |
| Windows Desktop Application (Thick Client)..... | 6 |
| ASP.NET or Windows Application Viewing Report Web Services | 6 |
| ASP.NET or Windows Application using CE Embedded (RAS) | 6 |
| CRYSTAL REPORTS 10 KEYCODE | 7 |
| CUSTOMIZING THE DATABASE DRIVER INSTALLATION OPTIONS (OPTIONAL).. | 7 |
| DEPLOYING AN ASP.NET APPLICATION | 9 |
| DEPLOYING A WINDOWS APPLICATION..... | 10 |
| FINDING MORE INFORMATION | 11 |

Introduction

Crystal Reports for Visual Studio .NET extends the powerful reporting capability of Crystal Reports on the Microsoft .NET platform. You can use the integrated Crystal Reports Designer in Visual Studio .NET to create a new Crystal report or modify an existing Crystal report. You can then keep the report on a local computer, or publish it as a Report Web Service on a Web server. Depending on whether you develop a Windows or Web application, you can host the report with either the Windows Form Viewer or the Web Forms Viewer respectively.

In addition to building Windows and ASP.NET Web applications using CR for VS .NET, you can build enterprise solutions used to view reports from CE Embedded (RAS) 10 and Crystal Enterprise (CE) 10.

For additional information and updates about Crystal Reports 10, please visit one of the following sites:

<http://www.businessobjects.com/products/reporting/crystalreports/net/>

http://www.businessobjects.com/products/dev_zone/net/

Deploying Reports in Visual Studio .NET Applications

When reports are added to a .NET Windows or Web application, they are added by default as an “Embedded Resource” for the application. What this means is that the report will be compiled into the assembly’s manifest, and will not be loaded from a separate report file (.RPT).

This allows Visual Studio .NET applications to be distributed and deployed without having to distribute report files separately. The advantage of doing this is that applications do not have to deploy external report files and end users cannot modify them. The disadvantage is that if a report needed to be modified, the entire application needs to be recompiled and redeployed for the report changes to be saved.

To avoid reports being compiled into the assembly’s manifest as an embedded resource, change how the report is saved in the project. After adding a report to a project, highlight the report in the Solution Explorer window and from the Properties Window change the **Build Action** property from **Embedded Resource** to **None**. Building this project will not compile the report into its assembly and reports will have to be loaded from disk. Reports can be loaded by specifying a file path using the .Load() method of the **ReportDocument** object.

When deploying applications that do not embed report files, the report files must be manually added to the Setup projects. The advantage of keeping reports outside of the assembly is that reports can be easily modified and redeployed without having to recompile and redeploy the entire assembly. The disadvantage is that strongly-typed report objects cannot be used in the application and the reports must be loaded from disk.

Overview of Deployment Scenarios

There are two scenarios for deploying the Crystal Reports 10 runtime files to a deployment machine:

- Using merge modules - Merge modules are provided for those developers who want to include the Crystal Reports 10 runtime files along with the install package for the surrounding application.
- Using the server install (interactive install) - The server install is provided for those developers who simply want to deploy the files to a target machine as a standalone task.

If your application requires a RAS server, run the Crystal Enterprise Embedded Edition install on your deployed server.

| | |
|-------------|--|
| NOTE | Your RAS server is licensed for deployment only if you have purchased Crystal Reports Advanced or have separately purchased Crystal Enterprise Embedded Edition. |
|-------------|--|

Deploying Using the Merge Modules

The components to add to Setup Projects for distributing and deploying Crystal Reports 10 applications include:

- Report files (.rpt)
- Dependencies
- Crystal Reports 10 merge modules

| | |
|-------------|--|
| NOTE | The Crystal Reports 10 merge modules are located on our support site: http://support.businessobjects.com/mergemodules |
|-------------|--|

Report Files (.RPT)

Report files (.RPT) need to be distributed when a .NET application is deployed. How the report files are deployed, depend on the need and architecture of the application. Refer to the previous section, [Deploying Reports in Visual Studio .NET applications](#) for details on deploying report files.

Common Language Runtime

Since Crystal Reports for Visual Studio .NET is based on the .NET Framework, the framework must also be distributed and/or installed on the target computer. When a Setup Project is created, the framework is usually not automatically included.

Crystal Reports 10 Merge Modules

There are four different Crystal Reports merge modules to choose from. These merge modules are designed to fit into different core scenarios so in most

scenarios only **one** of the merge modules will be required in any single setup package.

Adding multiple merge modules may cause your application to not deploy properly. However, the one exception is with the Mapping merge modules, which can be used in combination with any of the other 3 merge modules

| | |
|-------------|---|
| NOTE | To download the Crystal Reports 10 merge modules go to http://support.businessobjects.com/search and search for cr10_net_merge_modules.zip. |
|-------------|---|

CrystalReports10_NET_EmbeddedReporting.msm

- This merge module contains the Windows Form and Web Forms viewers (CrystalReportViewer controls), .NET SDK assemblies, RAS SDK, report processing engine, database and export components.

| | |
|-------------|--|
| NOTE | This merge module requires a valid keycode to be set. See section Crystal Reports 10 Keycode for more information. |
|-------------|--|

- This merge module allows developers to select which database drivers are installed. By default, all database drivers are installed. For more information, refer to the [Customizing the Database Driver Installation Options](#) section.
- This merge module deploys all Crystal Reports runtime files on a single machine/tier.
 - Windows application where all report processing is within the application
 - Web application where all report processing is within the web application tier

CrystalReports10_NET_RemoteReporting.msm

- This merge module contains Windows Form and Web Forms viewers, .NET SDK assemblies, RAS SDK
- This merge module deploys Crystal Reports runtime to clients used to view/process reports from a remote server.
 - Windows or Web application processing reports on remote RAS server.

CrystalReports10_NET_WebServiceReporting.msm

- This merge module contains Windows Form and Web Forms viewers, .NET SDK assemblies
- This merge module deploys Crystal Reports runtime to clients viewing Report Web Services.
 - Windows or Web applications accessing report web services.

CrystalReports10_Maps.msm

- This merge module contains mapping runtime files.
- This merge module deploys runtime files for reports using Map objects.
- This merge module is used in combination with the other version 10 merge modules.

Deploying Using the Server Install (Interactive Install)

In addition to the merge modules, a self-contained setup package, commonly referred to as the Server Install (or Interactive Install), is provided for web server installations. The **CrystalReports10_NET_EmbeddedInstall.msi** file installs all Crystal Reports runtime files on the computer.

Typically, the server install is used to install the runtime files on web servers for ASP.NET Web applications using Crystal Reports. This allows the Crystal Reports runtime to be installed once per machine, without having to include the Crystal Reports runtime with each application deployment.

The server install can be used to install the Crystal Reports runtime for .NET Windows applications but the **CrystalReports10_NET_EmbeddedInstall.msi** file needs to be run on each client computer. It is recommended to include the **CrystalReports10_NET_EmbeddedReporting.msm** merge modules with the .NET Windows application's setup package so that the user only has to run a single setup.

In fact, the server install is using the **CrystalReports10_NET_EmbeddedReporting.msm** merge modules internally. The only difference is that the server install does not allow customization of which database drivers are installed; all database drivers will be installed.

Running the **CrystalReports10_NET_EmbeddedInstall.msi** will require a valid Crystal Reports 10 keycode. For more information, see the section [Crystal Reports 10 Keycode](#).

| | |
|-------------|---|
| NOTE | The Server Install will only work for applications developed using the Crystal Reports 10 Developer or Advanced editions. |
|-------------|---|

Usage Scenarios

The merge modules are used in four different core scenarios. The following describe each of these scenarios and when to use the merge modules.

ASP.NET Web Application – Single Server

In this scenario, the ASP.NET application is loading reports from the same web server as the application. This means that the reports are being processed by the Web application on a single machine.

This requires that the **CrystalReports10_NET_EmbeddedReporting.msm** merge module be added to the setup used to install the application.

Alternatively, you can use the **CrystalReports10_NET_ServerInstall.msi** to install the Crystal Reports runtime on the web server before installing the ASP.NET Web application.

Windows Desktop Application (Thick Client)

In this scenario, the .NET Windows application is loading reports from the same desktop computer as the Windows application. This means that the application is loading and processing reports, locally, on the same computer as the application.

This will require the **CrystalReports10_NET_EmbeddedReporting.msm** merge modules to be added to the setup to install the .NET Windows application.

Using the **CrystalReports10_NET_ServerInstall.msi** is not the best option because the application will be installed to many individual machines. This requires the user to run two separate installs, one for the server install and the other for the application.

ASP.NET or Windows Application Viewing Report Web Services

In this scenario, the entire solution is made of a “client” and a “server”. The “client” is a .NET Windows application or an ASP.NET Web application that uses the Crystal Reports viewer control to view a report. The “server” are the Report Web Services that will be viewed by the “client”.

In most cases, the “client” will be installed to one computer (client computer or web server if ASP.NET) and the “server” will be installed to a separate web server computer (to host the report web services). This will require two setup installers, one for the “client” and the other for the “server”.

Since the “client” is used to only view reports stored and processed on a different server, only minimal runtime files are required to view the reports. In the “client” setup, only the **CrystalReports10_NET_WebServiceReporting.msm** merge module is required to be included in this setup.

Since the “server” will be loading and processing reports on the web server, the “server” setup only requires the **CrystalReports10_NET_EmbeddedReporting.msm** merge module.

ASP.NET or Windows Application using CE Embedded (RAS)

In this scenario, a .NET Windows application or an ASP.NET Web application is used as a client to view reports that are processed by a remote server which hosts the CE Embedded (RAS) Server.

Since the reports are already being hosted by RAS, this does not require any Crystal Reports runtime files to be distributed on the server. (For information on how to install and configure CE Embedded, refer to the Crystal Enterprise product documentation.)

However, the client application still requires some runtime files to be installed on the client computer. The setup for this client will require the **CrystalReports10_NET_RemoteReporting.msm** merge module as part of the setup.

| | |
|-------------|---|
| NOTE | These deployment scenarios do not include deploying an ASP.NET solution using Crystal Enterprise. For information on deploying an ASP.NET & CE solution, refer to the Crystal Enterprise documentation. |
|-------------|---|

Crystal Reports 10 Keycode

A Crystal Reports keycode is used to validate the licensing for the application. Both the **CrystalReports10_NET_EmbeddedReporting.msm** merge module and the **CrystalReports10_NET_EmbeddedInstall.msi** server installer require a valid keycode to be specified.

The license key is the alphanumeric string you receive when registering Crystal Reports.

| | |
|-------------|---|
| NOTE | Please note that this is <i>not</i> the 10-digit registration number. The keycode will have the following pattern: xxxxx-xxxxxxx-xxxxxxx . |
|-------------|---|

The keycode is located by using the Crystal License Manager installed on the computer with Crystal Reports 10. Selecting the 'Crystal Reports 10 Tools' program group in the Start menu can access the Crystal License Manager.

Step by step instructions for creating a setup project and entering the keycode can be found in the following sections: [Deploying an ASP.NET Application](#) and [Deploying a Windows Application](#).

Customizing the Database Driver Installation Options (Optional)

Your deployment needs may require including or excluding specific Crystal database drivers from your application install. Specific Crystal Reports database drivers can be individually included or excluded from the **CrystalReports10_NET_EmbeddedReporting.msm** merge module.

To change an option for a specific database driver, select **CrystalReports10_NET_EmbeddedReporting.msm** from your setup project and expand the **MergeModuleProperties** property.

Each of the merge module options corresponds to a driver in the table listed below. Setting the value for a specific driver to **1** will cause the driver to be installed with your application. Setting the value to **0** will exclude the driver from the target machine when your application is being installed. By default, all database drivers are set to install with **CrystalReports10_NET_EmbeddedReporting.msm**.

| Name | Description |
|-----------------|---|
| act | Crystal Reports database driver for PC Databases |
| btrieve | Crystal Reports database driver for Pervasive Databases |
| cdo32 | Crystal Data Object database driver |
| crdb_ado | Crystal Reports database driver for Microsoft ActiveX Data Objects/OLE DB |
| crdb_adoplus | Crystal Reports database driver for Microsoft ADO.NET |
| crdb_cdo | Crystal Reports database driver for Crystal Data Object |
| crdb_com | Crystal Reports database driver for COM data provider |
| crdb_dao | Crystal Reports database driver for Microsoft Data Access Objects |
| crdb_dataset | Crystal Reports database driver for DataSet provider |
| crdb_fielddef | Crystal Reports database driver for Field Definitions |
| crdb_filesystem | Crystal Reports database driver for File System data. |
| crdb_javabeans | Crystal Reports database driver for Java data |
| crdb_odbc | Crystal Reports database driver for ODBC |
| crdb_oracle | Crystal Reports database driver for Oracle |
| crdb_query | Crystal Reports database driver for Query Data |
| crddt32 | Crystal Reports Data Definition Tool |
| ibm_db2 | Crystal Reports database driver for DB2 Server |
| informix | Crystal Reports database driver for Informix Online Server |
| p2bbde | Crystal Reports database driver for IDAPI |
| p2bxbse | Crystal Reports database driver for xBase |
| p2ixbse_dll | Crystal Reports physical database DLL for xBase |
| p2sacl | Crystal Reports database driver for Public Folder ACL |
| p2sevt | Crystal Reports database driver for NT Event Log |
| p2sexsr | Crystal Reports database driver for Exchange Server Admin |
| p2smapi | Crystal Reports database driver for Exchange Folders and Address Book |
| p2smsiis | Crystal Reports database driver for IIS Log Files |
| p2soutlk | Crystal Reports database driver for Outlook Data |
| p2srepl | Crystal Reports database driver for Exchange Public Folder Replica |
| p2strack | Crystal Reports database driver for Exchange Server Message Tracking Logs |
| p2swblg | Crystal Reports database driver for Web Log Files |
| querybuilder | Crystal Query Builder |
| sybase | Crystal Reports database driver for Sybase |

Deploying an ASP.NET Application

The following procedures assume that you already have a Web application or Web services application ready for deployment. To deploy a Web application using the Crystal Reports merge modules:

1. In Visual Studio .NET, open your Web application, go to the **View** menu, and select **Solution Explorer**. Right-click your Web application solution and from the pop-up menu, select **Add** then click **New Project**.
2. The **Add New Project** dialog box appears. Select **Setup and Deployment Projects** in the left pane and select **Web Setup Project** in the right pane. Click **OK**.
3. In the Solution Explorer, right click the **Web Setup Project** and select **Add** then click **Project Output**.
4. The **Add Project Output Group** dialog box appears. Select **Primary Output** and **Content Files** to add to your Web Setup project.

| | |
|-------------|--|
| NOTE | When the Primary Output is added to the project, several Crystal Reports detected dependencies will be added. Select all of the Crystal Reports dependencies and right-click on the highlighted files and select ' Exclude ' so that these files are not installed on the target machine. |
|-------------|--|

5. In the Solution Explorer, right-click the Web Setup project and select **Add** then click **Merge Module**.
6. The **Add Modules** dialog box appears. Select the appropriate merge modules (See section above titled [Crystal Reports 10 Merge Modules](#)) and add them to your project. Click **OK**.

*** Steps 7 and 8 are for
CrystalReports10_NET_EmbeddedReporting.msm only ***

7. From your Setup project, select the Crystal Reports merge module (you just added) to display the properties.
8. Expand the **MergeModuleProperties** and enter a valid license key the in **Keycode Properties** box.

The license key is the alphanumeric string you receive when registering Crystal Reports. This is **mandatory** when you deploy a Crystal Reports for Visual Studio .NET application using **CrystalReports10_NET_EmbeddedReporting.msm**.

For a description of the other configuration options, please see the section above titled [Customizing the Database Driver Installation](#).

| | |
|-------------|--|
| NOTE | A license key will be given to you once you have registered Crystal Reports for Visual Studio .NET. You can also find the license key by opening VS .NET and going to the 'Help' menu and selecting 'About...'. Under the 'Installed application list' is the license key for Crystal Reports 10 for Visual Studio .NET. |
|-------------|--|

9. To build your Web Setup project, highlight your deployment package (websetup1) in the Solution Explorer, right-click and select **Build**. You are now ready to deploy your ASP.NET Web application.

Deploying a Windows Application

The following procedures assume that you already have a Windows application ready for deployment. To deploy a Windows application using the Crystal Reports merge modules:

1. In VS .NET, open your Windows application and go to the **View** menu and select **Solution Explorer**. Right click your Windows application solution and from the pop up menu, select **Add** then click **New Project**
2. The **Add New Project** dialog box appears. Select **Setup and Deployment Projects** in the left pane and select **Setup Project** in the right pane. Click **OK**.
3. In the Solution Explorer, right click the new setup project and select **Add** then click **Project Output...**
4. The **Add Project Output Group** dialog box appears. Select **Primary Output** to be added to your Setup project.

| | |
|-------------|---|
| NOTE | When the Primary Output is added to the project, several Crystal Reports detected dependencies will be added. Select all of the Crystal Reports dependencies and right-click on the highlighted files and select 'Exclude' so that these files are not installed on the target machine. |
|-------------|---|

5. In the Solution Explorer, right click the Setup project and select **Add** then click **Merge Module...**
6. The **Add Modules** dialog box appears. Select the appropriate merge modules (See section above titled [Crystal Reports 10 Merge Modules](#)) and add them to your project. Click **OK**

*** Steps 7 and 8 are for
CrystalReports10_NET_EmbeddedReporting.msm only ***

7. From your Setup project, select the Crystal Reports merge module you just added to display the properties.
8. Expand the **MergeModuleProperties** and enter a valid license key the in **Keycode Properties** box. The license key is the alphanumeric string you receive when registering Crystal Reports. Please note that this is *not* the 10-digit registration number. This is **mandatory** whenever you deploy a Crystal Reports for Visual Studio .NET application using CrystalReports10_NET_EmbeddedReporting.msm.

For a description of the other configuration options, please see the section above titled [Customizing the Database Driver Installation](#).

| | |
|-------------|--|
| NOTE | A license key will be given to you once you have registered Crystal Reports for Visual Studio .NET. You can also find the license key by opening VS .NET and going to the 'Help' menu and selecting 'About...'. Under the 'Installed application list' is the license key for Crystal Reports for Visual Studio .NET. |
|-------------|--|

9. To build your Setup project, right-click your deployment package (setup1) in the Solution Explorer, and select **Build**. You are now ready to deploy your Windows application.

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

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.