

Crystal Reports and Visual Studio .NET

Application Deployment

Overview

This document discusses the options for deploying .NET applications built using either the bundled version of Crystal Reports for Visual Studio .NET or the standalone version of Crystal Reports 9 for use in both Visual Studio 2002 and Visual Studio 2003. Both merge modules and interactive install packages are discussed.

Contents

INTRODUCTION	2
INCLUDING REPORTS IN .NET APPLICATIONS	2
OVERVIEW OF DEPLOYMENT METHODS	2
DEPLOYING USING THE MERGE MODULES	3
<i>Report Files (.RPT)</i>	<i>3</i>
<i>Crystal Reports Merge Modules</i>	<i>3</i>
Crystal Reports 9 Full Version with VS 2002 or VS 2003	4
Crystal Reports Bundled Version with VS 2002 or VS 2003	5
<i>Runtime</i>	<i>6</i>
.NET Framework	6
ADO.NET	6
DEPLOYING USING THE INTERACTIVE INSTALL	7
DEPLOYING AN ASP.NET WEB APPLICATION	7
DEPLOYING A WINDOWS APPLICATION	9
ADDITIONAL RESOURCES	10
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	10

Introduction

Crystal Reports for Visual Studio .NET (CR for VS .NET) extends the powerful reporting capability of Crystal Reports on the Microsoft .NET platform. You can use the Crystal Report Designer in Visual Studio .NET (VS .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.

Including Reports in .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).

Having reports as an embedded resource allows .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 prevent reports from being compiled into the assembly's manifest as an embedded resource, use the following steps:

1. After adding a report to a project, highlight the report in the **Solution Explorer** window.
2. From the **Properties** windows change the **Build Action** property from **Embedded Resource** to **None**.

Using the above steps will not compile the report into the assembly's manifest, therefore, reports will have to be loaded from disk. Reports can be loaded from disk 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 Methods

There are two methods for deploying the Crystal Reports 9 runtime files to a deployment computer:

- Using merge modules

- Using an interactive install (also known as ‘redistributable’ or ‘server install’)

If you have the bundled version of Crystal Reports 9, then you are limited to using the merge modules. However, if you have the full version of Crystal Reports 9, then you can also use the interactive install.

NOTE	To determine if you have full or bundled version of Crystal Reports: 1. Inside the Visual Studio .NET IDE, click on the Help Menu and select the About option. 2. Under the installed product list look for the Crystal Reports entry. • If you see “Crystal Reports 9 for Visual Studio .NET”, then you are using the full version of Crystal Reports • If you see “Crystal Reports for Visual Studio .NET”, then you are using the bundled version of Crystal Reports
-------------	---

Both scenarios install the same files, however the method of deployment is different. Merge modules are provided for those developers who want to include the Crystal Reports runtime files along with the install package for the surrounding application. The interactive install is provided for those developers who simply want to deploy the files to a server machine as a standalone task.

Deploying Using the Merge Modules

There are a few components to add to setup projects when distributing and deploying Crystal Reports 9 applications.

- Report Files (.RPT)
- Crystal Reports Merge Modules
- Dependencies

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, [Including Reports in .NET Applications](#) for details on deploying report files.

Crystal Reports Merge Modules

The merge modules that need to be added to the setup project will vary depending on which version of Crystal Reports you are using as well as the version of Visual Studio. To determine which version you are using, use the following steps:

1. Inside the Visual Studio .NET IDE, click on the **Help** Menu and select the **About** option.
2. Under the installed product list look for the Crystal Reports entry.
 - If you see “Crystal Reports 9 for Visual Studio .NET”, then you are using the **Full Version** of Crystal Reports

- If you see “Crystal Reports for Visual Studio .NET”, then you are using the **Bundled Version** of Crystal Reports

3. At the top of the **About** window, look for the line containing the phrase “Microsoft Development Environment”.

- If you see “Microsoft Development Environment 2002”, then you are using **Visual Studio 2002**
- If you see “Microsoft Development Environment 2003”, then you are using **Visual Studio 2003**

Depending on the version being used, you will need to use certain merge modules.

Crystal Reports 9 Full Version with VS 2002 or VS 2003

If you are using the full version of Crystal Reports 9 with Visual Studio 2002 or 2003, download the merge modules from the Crystal Technical Support web site at the following URL:

<http://support.crystaldecisions.com/search>

Once on the web site, search for “cr9netmergemodules” to find the merge modules for Crystal Reports 9 and VS .NET corresponding to your language.

There are potentially four merge modules that need to be included in a setup project to deploy reports.

- reportengine.msm
- crnetruntime.msm
- mapping.msm
- license.msm

Reportengine.msm: This merge module is used to install the core files that comprise the Crystal Reports engine. This merge module is always required for Crystal Reports 9 .NET applications.

Crnetruntime.msm: This merge module is used to install the Crystal Reports 9 .NET managed components, such as:

- CrystalDecisions.CrystalReports.Engine.dll
- CrystalDecisions.Web.dll
- CrystalDecisions.Windows.Forms.dll

Crnetruntime.msm is required for all application using any of the managed .NET components.

Mapping.msm: This merge module is used to install the files required to support displaying geographic maps inside reports. This merge module is only required if geographic mapping is included in your reports.

License.msm: This merge module is used to specify a license keycode in order to determine the licensing of the report engine. This merge module is always required for .NET applications.

When license.msm is added to a setup project, it exposes a **LicenseKey** property that accepts a keycode. The license key is emailed when the product is registered. This key is also available from the VS .NET **Help | About...** dialog, under the installed products list.

The 19-digit license key must be entered into the **LicenseKey** property for the license.msm merge module.

NOTE	If the license key is not set for the license.msm merge module, an error will occur when building the project. If the setup is installed on a target computer without specifying the correct license key for the merge module, various "keycode.dll" errors will occur. The three most common are: "Invalid or Missing KeycodeV2.dll" - See kbase c2010681 "Job Failed Because a Free License Could not be Obtained" – See kbase c2012716 "Err Msg: "Cannot find keycode2.dll or invalid keycode" appears in VS .NET – See kbase c2011205
-------------	---

Crystal Reports Bundled Version with VS 2002 or VS 2003

CR for VS .NET provides merge modules in setup projects to make deployment a simple task. The merge modules are installed to the following folder, by default:

C:\Program Files\Common Files\Merge Modules

There are four merge modules that need to be included in a setup project include:

VS .NET 2002 Applications	VS .NET 2003 Applications
Managed.msm	Crystal_Managed2003.msm
Database_Access.msm	Crystal_Database_Access2003.msm
Database_Access_enu.msm	Crystal_Database_Access2003_enu.msm
RegWiz.msm	Crystal_RegWiz2003.msm

Managed.msm / Crystal_Managed2003.msm: This merge module is used to install all the CR for VS .NET managed components, such as:

- CrystalDecisions.CrystalReports.Engine.dll
- CrystalDecisions.Web.dll
- CrystalDecisions.Windows.Forms.dll

Database_access.msm / Crystal_Database_Access2003.msm: This merge module is used to install all database drivers the report uses to connect to various

types of data sources. This merge module also installs export destination and format drivers, which are used to save report in different file formats, such as RTF and HTML. In fact, this merge module is used to install all non-managed runtime components for CR for VS .NET, including charting components.

Database_access_enu.msm / Crystal_Database_Access2003_enu.msm: This merge module is used to install language specific (localized) components. Some of the localized charting and exporting (like PDF) components are installed by this component. Each language the product is released in will have a specific version of this merge module.

RegWiz.msm / Crystal_RegWiz2003.msm: This merge module is used to configure registration and licensing information on deployed computers (server and/or client). When building setup projects using this merge module, it is important to provide the license key for it.

When RegWiz.msm / Crystal_RegWiz2003.msm is added to a setup project, it exposes a **LicenseKey** property that accepts a keycode. The license key is emailed when the product is registered. This key is also available from the VS .NET **Help | About...** dialog, under the installed products list.

The 19-digit license key must be entered into the **LicenseKey** property for the regwiz.msm / Crystal_RegWiz2003 merge module.

NOTE	If the license key is not set for the RegWiz.msm / Crystal_RegWiz2003.msm merge module, an error will occur when building the project. If the setup is installed on a target computer without specifying the correct license key for the merge module, various "keycode.dll" errors will occur. The three most common are: "Invalid or Missing KeycodeV2.dll" - See kbase c2010681 "Job Failed Because a Free License Could not be Obtained" – See kbase c2012716 "Err Msg: "Cannot find keycode2.dll or invalid keycode" appears in VS .NET – See kbase c2011205
-------------	---

Runtime

.NET Framework

Since CR for VS .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, usually the framework is not automatically included.

ADO.NET

If any of the report files are based on ADO.NET dataset objects, you will need to include the following two merge modules.

- For VS .NET 2002, use VC_CRT.msm, and VC_STL.msm
- For VS .NET 2003, use VC_User_CRT71_RTL_X86_---.msm and VC_User_STL71_RTL_X86_---.msm

These two merge modules are required because the database driver, Crdb_adoplus.dll, is dependent on the following files that the merge modules install:

- For VS .NET 2002 - Msvcr70.dll and Msvcp70.dll

- For VS .NET 2003 – Msvcr71.dll and Msvcp71.dll

NOTE	The files from these merge modules must be available to the application. This can be either in the system path or in the application folder. It is recommended that they be installed in the application folder. If the files are not accessible to the application, runtime errors will occur when trying to process reports.
-------------	--

Deploying Using the Interactive Install

The interactive install will only work with the full version of Crystal Reports 9 Developer or Advanced editions.

NOTE	To determine if you have full or bundled version of Crystal Reports: 1. Inside the Visual Studio .NET IDE, click on the Help Menu and select the About option. 2. Under the installed product list look for the Crystal Reports entry. • If you see "Crystal Reports 9 for Visual Studio .NET", then you are using the full version of Crystal Reports • If you see "Crystal Reports for Visual Studio .NET", then you are using the bundled version of Crystal Reports
-------------	---

When deploying a web application, or deploying multiple Windows applications to the same computer, it is recommended that the Crystal dependencies be installed separately. This stand-alone install package can be found on the web at the following URL:

<http://support.crystaldecisions.com/search>

Once on the web site, search for the file name "cr9netredist" without the quotes and find the redistribution package for Crystal Reports 9 for VS .NET corresponding to your language.

Run the install (.MSI) file on the computer you are deploying to and enter your Crystal Reports 9 keycode when asked. You can find the keycode by clicking on **About Microsoft Development Environment** located in the **Help** menu of Visual Studio. The keycode can be found in the **Installed Products** listing.

Deploying an ASP.NET Web Application

The following procedures assume that you already have a web application or web services application ready for deployment. To deploy an ASP.NET Web application using the CR merge modules:

1. In VS .NET, open your web application and go to the **View** menu and select **Solution Explorer**. From the **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 be added to your Web Setup project.

When the **Primary Output** is added to the project, some detected dependencies will be added.

- If you are using the bundled version of CR 9, add:
dotnetfxredist_x86_enu.msm, and **managed.msm**
- If you are using the full version of CR 9, add:
dotnetfxredist_x86_enu.msm and **dotnetcrystalreports.msm**

In both cases, the **dotnetfxredist_x86_enu.msm** dependency should be **excluded** if it is not already.

NOTE	If you are using the full version of CR 9: The dotnetcrystalreports.msm should be excluded because it will conflict with components in the other merge modules. Right-click on this item in the Solution Explorer and select Exclude.
-------------	---

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

NOTE	If you deploy a web application that uses an ADO.NET Dataset, you also need to include the two additional merge modules listed in the Runtime section above.
-------------	---

7. From your WebSetup project, select the licensing merge module to display the properties.
8. Expand the **MergeModuleProperties** and enter a valid license key the in **License Key** Properties box. The license key is the 19-digit 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.

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 Web Setup project, highlight and right-click your deployment package (websetup1) in the Solution Explorer 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 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 Windows application 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.

When the **Primary Output** is added to the project, some detected dependencies will be added.

- If you are using the bundled version of CR 9, add:
dotnetfxredist_x86_enu.msm, and **managed.msm**
- If you are using the full version of CR 9, add:
dotnetfxredist_x86_enu.msm and **dotnetcrystalreports.msm**

In both cases, the **dotnetfxredist_x86_enu.msm** dependency should be **excluded** if it is not already.

NOTE	If you are using the full version of CR 9: The dotnetcrystalreports.msm should be excluded because it will conflict with components in the other merge modules. Right-click on this item in the Solution Explorer and select Exclude .
-------------	--

5. In the Solution Explorer, right-click the WebSetup 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 Merge Modules](#)) and add them to your project. Click **OK**.

NOTE	If you deploy a Windows application that uses an ADO.NET Dataset, you also need to include the two additional merge modules listed in the Runtime section above.
-------------	--

7. From your Setup project, select the licensing merge module to display the properties.
8. Expand the **MergeModuleProperties** and enter a valid license key the in **License Key** Properties box. The license key is the 19-digit 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.

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

Additional Resources

For additional .NET information and resources, please visit one of the following sites:

.NET Product site: <http://www.crystaldecisions.com/net>

Developer Zone: <http://www.crystaldecisions.com/getinthezone>

Support site search: <http://support.crystaldecisions.com/search>

Listing of sample applications:

<http://support.crystaldecisions.com/fix/samplescr.asp>

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

<http://support.crystaldecisions.com/>

Email Support:

<http://support.crystaldecisions.com/support/answers.asp>

Telephone Support:

<http://www.crystaldecisions.com/contact/support.asp>