

Crystal Reports 9

Application Deployment

Overview

This document discusses the options for deploying applications built with the Crystal Reports 9 .NET components. Both merge modules and a interactive install package are discussed.

Contents

INTRODUCTION	2
DEPLOYING REPORTS IN .NET APPLICATIONS	2
OVERVIEW OF DEPLOYMENT SCENARIOS	2
DEPLOYING USING THE MERGE MODULES	3
<i>Report Files (.RPT).....</i>	<i>3</i>
<i>Crystal Reports Merge Modules</i>	<i>3</i>
<i>Runtime</i>	<i>4</i>
DEPLOYING A WEB/WEB SERVICES APPLICATION	5
DEPLOYING A WINDOWS APPLICATION.....	7
DEPLOYING TO A CLIENT/SERVER COMPUTER.....	8
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	9

Introduction

Crystal Reports 9 extends the powerful reporting capability of Crystal Reports for Visual Studio .NET. You can use the Crystal Report 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 Forms Viewer or the Web Forms Viewer respectively.

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

<http://www.crystaldecisions.com/net>

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

Deploying Reports in .NET Applications

When reports are added to a .NET Windows or Web applications, 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 .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 will need 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 added to the project. After adding a report to a project, highlight the report in the Solution Explorer window and from the Properties Windows 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.

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 9 runtime files to a deployment machine: using merge modules or using an interactive install. Both install the same files however it 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.

This document will discuss deploying Crystal Reports 9 via the merge modules.

Deploying using the Merge Modules

There are a few components to add to Setup projects for 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 architect of the application. Refer to the previous section, 'Deploying Reports in .NET applications' for details on deploying report files.

Crystal Reports Merge Modules

The merge modules Crystal Reports 9 are available for download from the Crystal Care Technical Support web site at the following URL:

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

Once on the web site, search for the merge modules by selecting Crystal Reports from "Select a product," 9 from "Select a product version," and Product update from "Select a file type," and then click Go.

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

The **reportengine.msm** merge module is used to install the core files that comprise the Crystal Reports engine. This merge modules is always required for Crystal Reports 9 .NET applications.

The **crnetruntime.msm** merge module is used to install the Crystal Reports 9 .NET managed components, such as CrystalDecisions.CrystalReports.Engine.DLL, CrystalDecisions.Web.DLL, and CrystalDecisions.Windows.Forms.DLL. This merge module is required for all application using any of the managed .NET components.

The **mapping.msm** 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.

The **license.msm** merge module is used to specify a license keycode in order to determine the scalability 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.

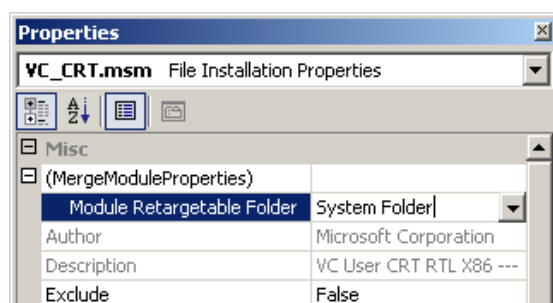
If the license key is not set for the license.msm merge module, an error will occur when building the project. If the setup is installed on a target computer without specifying the license key for the merge module, various “keycode.dll” errors will occur.

Runtime

Since Crystal Report 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 merge modules will be automatically included.

There are two other merge modules that may need to be included depending on the report(s). If any of the report files are based on ADO.NET dataset objects, the **VC_CRT** and **VC_STL** merge modules need to be included in the setup project. These two merge modules are required because the database driver, crdb_adoplus.dll, is dependent on files installed by these merge modules.

Important: When these two merge modules are added, there is a ‘MergeModuleProperties’ property for the file. If you expand that, one of the properties is “Module Retargetable Folder” – this is set to Default. Click on the drop down list to select a different folder from the default. Select the ‘Browse’ option and it will list the folders available on the target machine. A system folder will need to be added before you can set this as the new target folder. This is shown in the screenshot.



Deploying a Web/Web Services Application

The following procedures assume that you already have a web application or web services application ready for deployment. To deploy a Web/Web Service Application:

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.

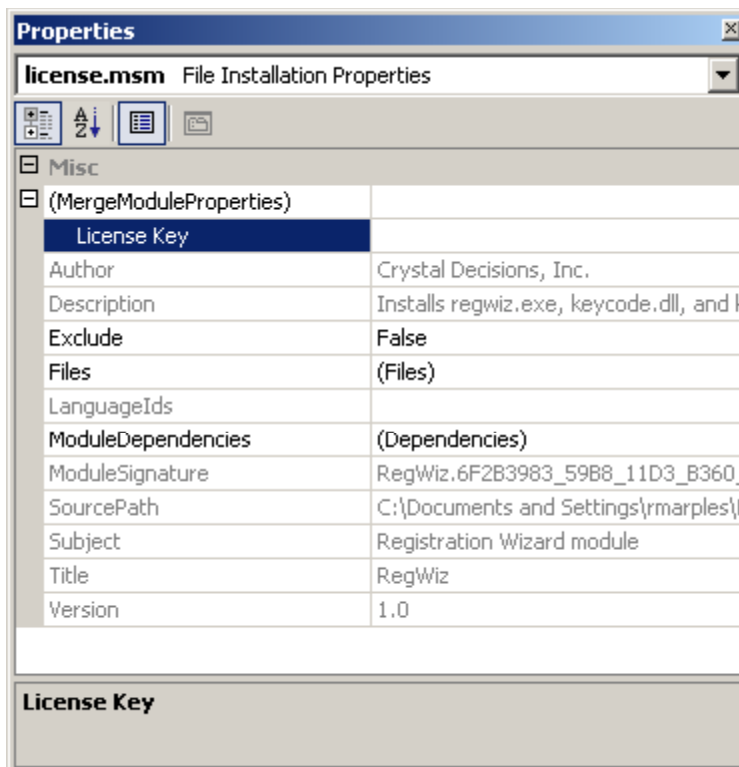
When the **Primary Output** is added to the project, the dependencies **dotnetfxredist_x86_enu.msm**, and **dotnetcrystalreports.msm** will be added automatically.

5. The **dotnetcrystalreports.msm** should be excluded since this is an internal merge module. Right-click on this item in the **Solution Explorer** and select **Exclude**.
6. In the Solution Explorer, right click the WebSetup project and select **Add** then click **Merge Module...**
7. The Add Modules dialog box appears. Select the merge modules **reportengine.msm**, **crnetruntime.msm**, **license.msm**, and optionally **license.msm** to add to your WebSetup project. Click **OK**.

NOTE	If you deploy a Web Application that uses an ADO.NET Dataset, you also need to include the VC_CRT , and VC_STL merge modules. See 'Runtime' section for more information.
-------------	---

8. From your WebSetup project, select the **license.msm** to display the properties.

9. Expand the **MergeModuleProperties** and enter a valid license key the in **License Key** Properties box.



10. This is **mandatory** whenever you deploy a Crystal Reports for Visual Studio .NET application.

NOTE	A license keycode 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.
-------------	---

11. To build your Web Setup application, highlight your deployment package (websetup1) in the Solution Explorer, right-click and select **Build**. You are now ready to deploy your web/web services application.

Deploying a Windows Application

The following procedures assume that you already have a Windows application ready for deployment. To deploy a Windows Application:

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 add to your Setup project.

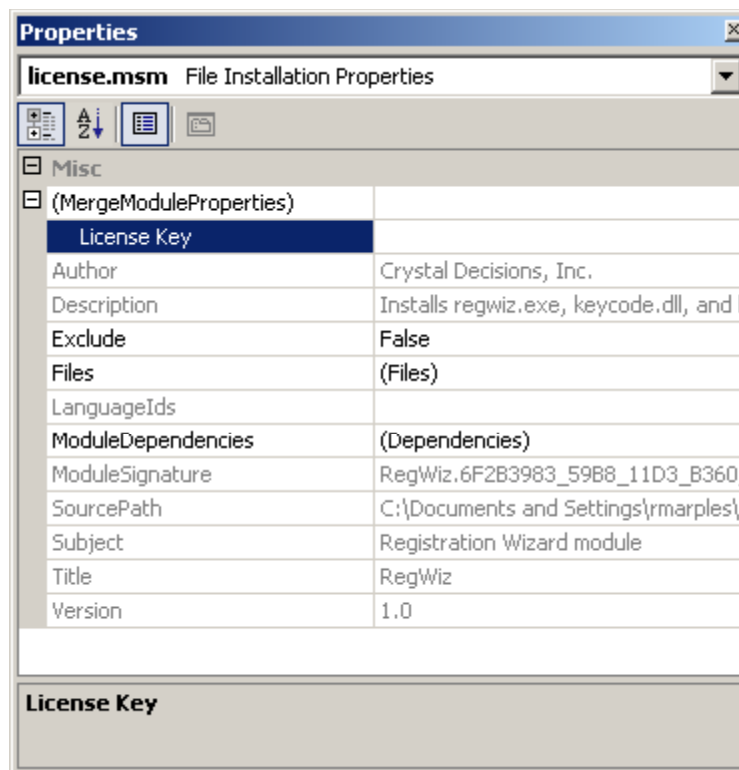
When the **Primary Output** is added to the project, the dependencies **dotnetfxredist_x86_enu.msm**, and **dotnetcrystalreports.msm** are added automatically.

5. The **dotnetcrystalreports.msm** should be excluded since this is an internal merge module. Right-click on this item in the **Solution Explorer** and select **Exclude**.
6. In the Solution Explorer, right click the Windows application and select **Add** then click **Merge Module...**
7. The **Add Modules** dialog box appears. Select the merge modules **reportengine.msm**, **crnetruntime.msm**, **license.msm** and optionally **mapping.msm** to add to your Setup project. Click **Open**.

NOTE	If you deploy a Windows Application that uses an ADO.NET Dataset, you also need to include the VC_CRT , and VC_STL merge modules. See 'Runtime' section for more information.
-------------	---

8. From your Setup project, select the **license.msm** to display the properties

9. Expand the **MergeModuleProperties** and enter a valid license key the in **License Key** Properties box.



10. 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.
-------------	---

11. To build your Setup application, highlight your deployment package in the Solution Explorer, right-click and select **Build**. You are now ready to deploy your Windows application.

Deploying to a Client/Server Computer

When deploying to a client or server computer, always ensure that the computer already has the .NET Framework installed. The .NET Framework can be found on the Visual Studio 7 Windows Components Update CD or from the Microsoft's web site at www.microsoft.com.

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>