

Crystal Reports for Visual Studio .NET

Application Deployment

Overview

This document discusses how to create an application deployment project to allow you to seamlessly deploy Crystal Report for Visual Studio .NET on a client or server computer.

Contents

INTRODUCTION	2
DEPLOYING REPORTS IN .NET APPLICATIONS	2
DEPLOYING CR FOR VS .NET.....	2
<i>Report Files (.RPT).....</i>	<i>3</i>
<i>Merge Modules</i>	<i>3</i>
<i>Runtime</i>	<i>4</i>
DEPLOYING A WEB/WEB SERVICES APPLICATION	4
DEPLOYING A WINDOWS APPLICATION.....	7
DEPLOYING TO A CLIENT/SERVER COMPUTER.....	11
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	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 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.

Deploying CR for VS .NET

There are a few components to add to Setup projects for distributing and deploying .NET applications using Crystal Reports for Visual Studio .NET (CR for VS .NET).

- Report Files (.RPT)
- Merge Modules
- Runtime

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.

Merge Modules

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 to deploy reports.

- Managed.MSM
- Database_Access.MSM
- Database_Access_enu.MSM
- REGWIZ.MSM

The **managed.msm** merge module is used to install all the CR for VS .NET managed components, such as CrystalDecisions.CrystalReports.Engine.DLL, CrystalDecisions.Web.DLL, and CrystalDecisions.Windows.Forms.DLL.

The **database_access.msm** 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 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.

In order to track licensing information, **regwiz.msm** is used to configure registration information on deployed computers (server or client). When building setup projects using this merge module, it is important to provide the license key for it.

When regwiz.msm is added to a setup project, it exposes a LicenseKey property that must be set before building the setup project. The 19-digit 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 license key must be entered into the LicenseKey property for the regwiz.msm merge module.

If the license key is not set for the regwiz.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 "keycodev2.dll" errors will occur.

For more information, go to <http://support.crystaldecisions.com/kbase> and search for knowledge base articles: c2010681 and c2011205.

Runtime

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 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 the files, MSVCR70.DLL and MSVCP70.DLL (which the merge modules install).

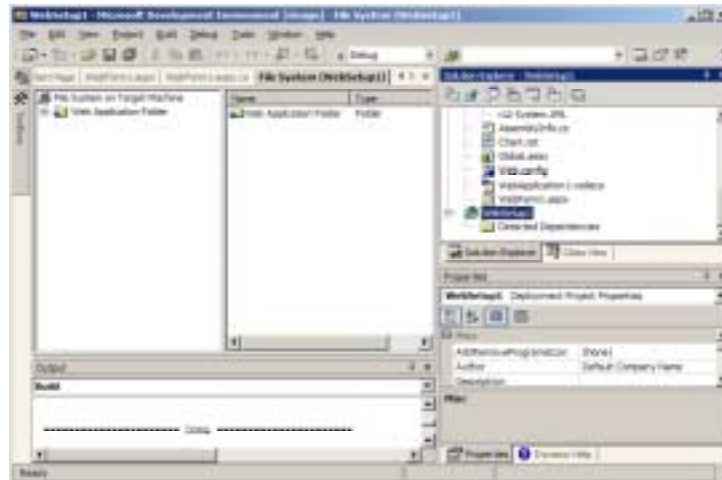
NOTE	The files from these merge modules must be installed in the application folder. For example, if the executable for a Windows application is not in the same folder as these files, runtime errors will occur when trying to process reports. For an ASP.NET Web application, these files must be installed in the application's root folder (where the .ASPX pages are).
-------------	--

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:

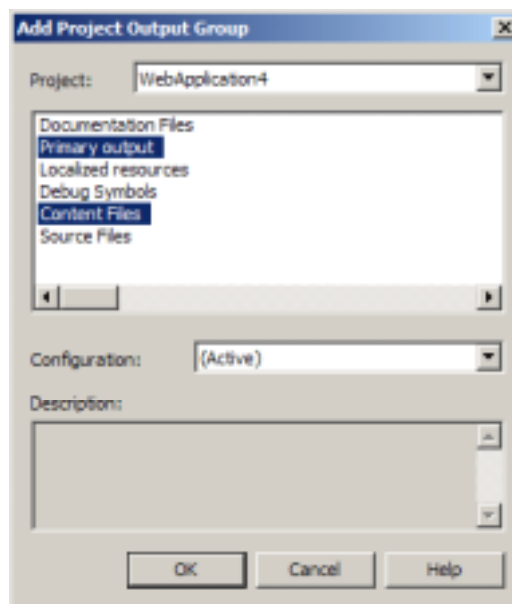
1. In VS .NET, open your web application and go to the **View** menu and select **Solution Explorer**. From the **Solution Explorer**, right click your web application solution and from the pop up menu, select **Add** then click **New Project**
2. The **Add New Project** dialog box appears. Select **Setup and Deployment**
3. Projects in the left pane and select **Web Setup Project** in the right pane. Click **OK**.

4. In the Solution Explorer, right click the **Web Setup Project** and select **Add** then click **Project Output...**

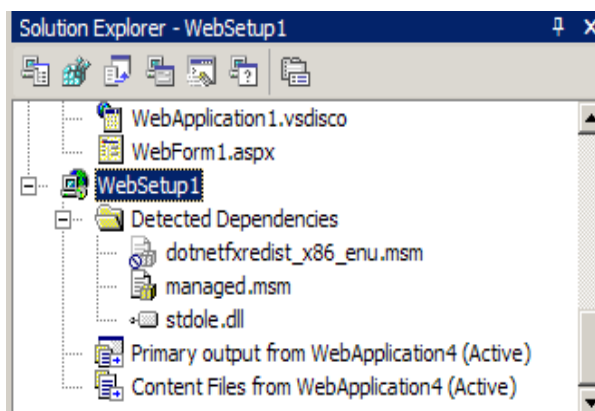


5. The **Add Project Output Group** dialog box appears. Select **Primary Output** and **Content Files** to be added to your Web Setup project.

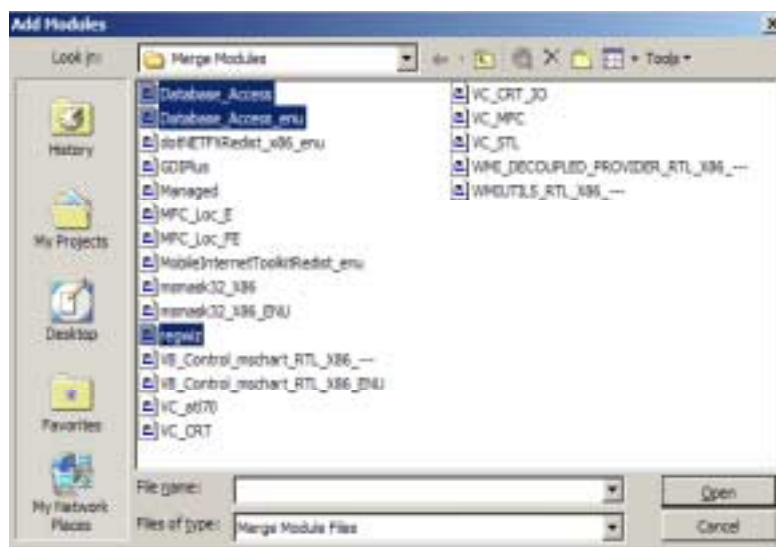
When the **Primary Output** is added to the project, the dependencies **dotnetfxredis_x86_enu.msm**, and **managed.msm** will be added automatically.



6. In the Solution Explorer, right click the WebSetup project and select **Add** then click **Merge Module...**

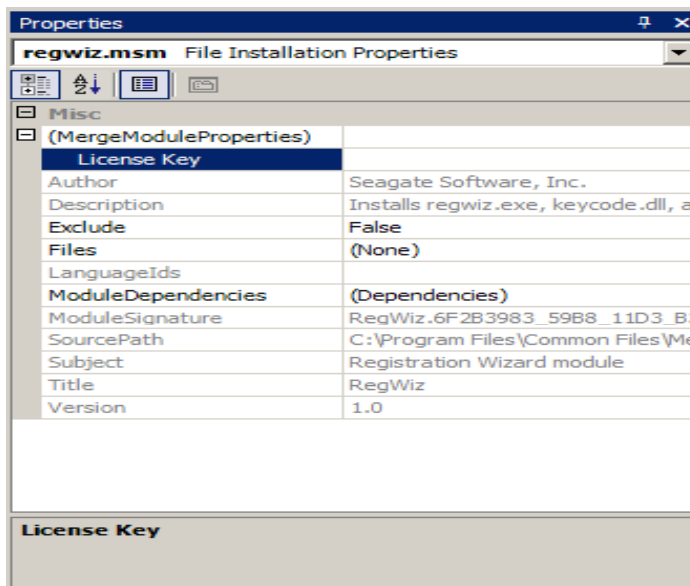


7. The Add Modules dialog box appears. Select the merge modules **Database_Access**, **Database_Access_enu** and **regwiz** to add to your WebSetup project. Click **OK**.

**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 the '[Runtime](#)' section for more information.

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



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

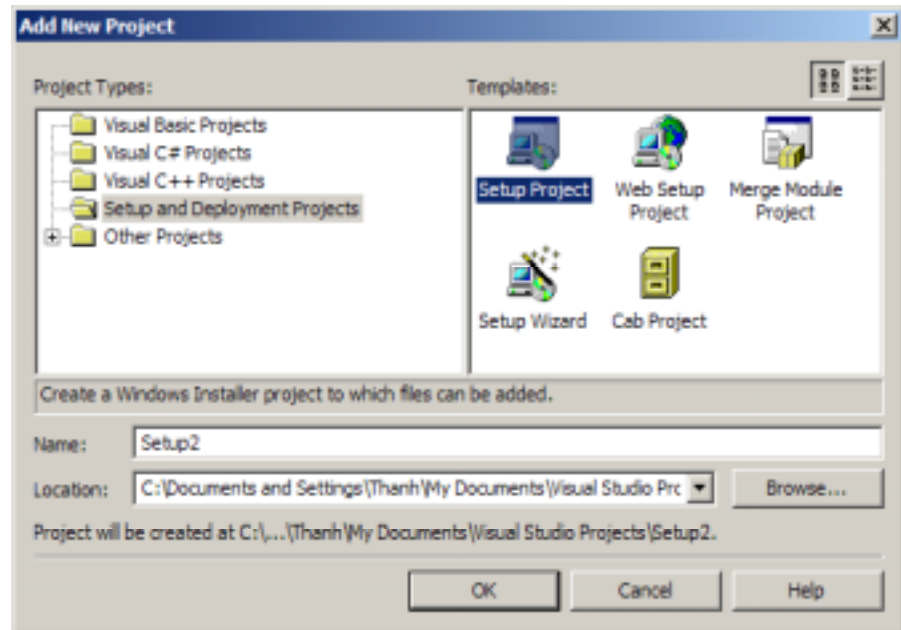
NOTE	A license key will be given to you once you have registered Crystal Reports for Visual Studio .NET. You can also find the license key by opening VS .NET and going to the 'Help' menu and selecting 'About...'. Under the 'Installed application list' is the license key for Crystal Reports for Visual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

10. 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.

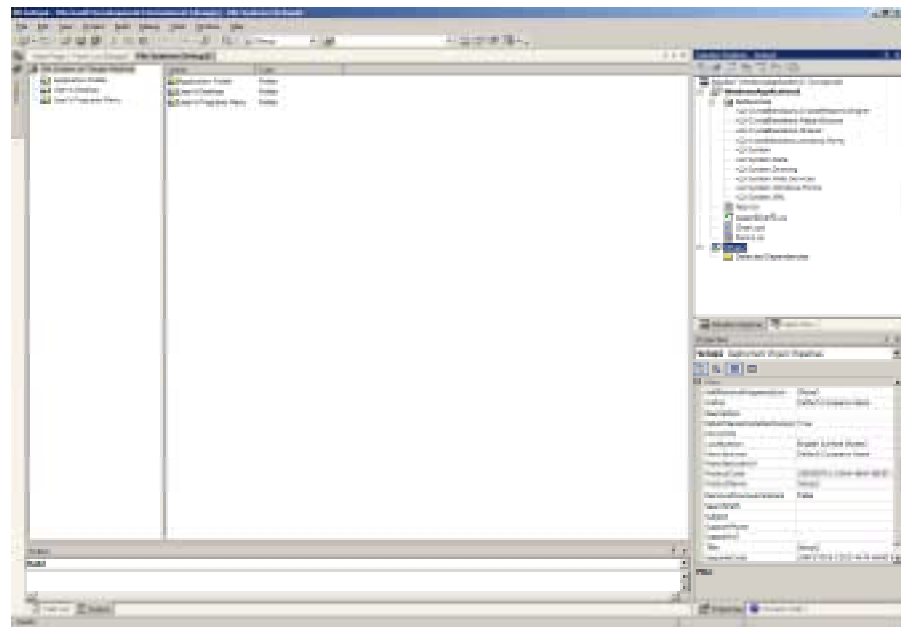
Deploying a Windows Application

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**.

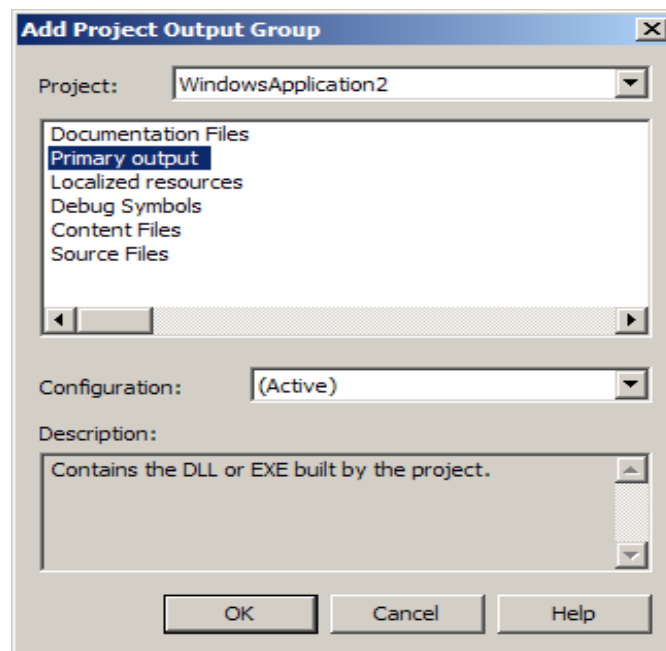


3. In the Solution Explorer, right click the Windows application and select **Add** then click **Project Output...**

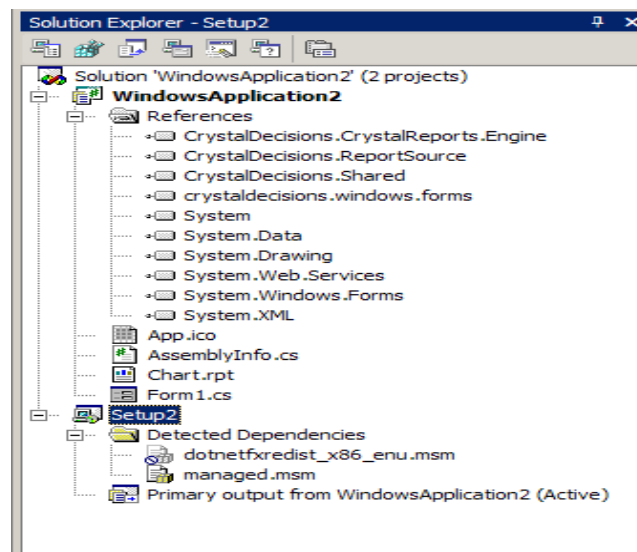


4. The **Add Project Output Group** dialog box appears. Select **Primary Output** to add to your Setup project.

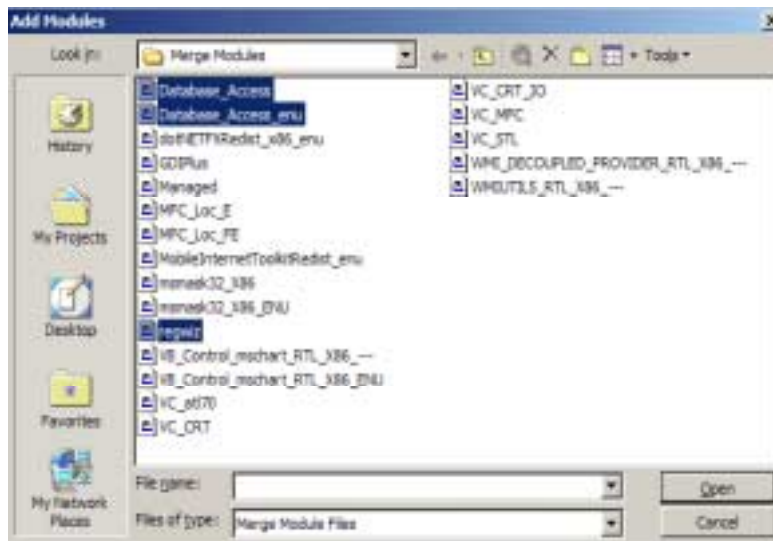
When the **Primary Output** is added to the project, the dependencies **dotnetfxredis_x86_enu.msm**, and **managed.msm** are added automatically.



5. . In the Solution Explorer, right click the Windows application and select **Add** then click **Merge Module...**

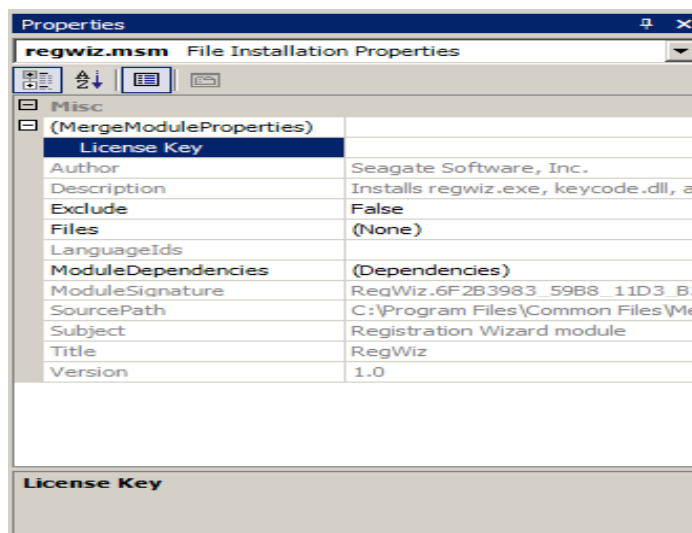


6. The **Add Modules** dialog box appears. Select the merge modules **Database_Access**, **Database_Access_enu** and **regwiz** to add to your Setup project. Click **Open**.

**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 the '[Runtime](#)' section for more information.

7. From your Setup project, select the **Regwiz.msm** to display the regwiz properties



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

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 application, highlight your deployment package (setup2) 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>