

Report Designer Component 9

Creating a RDC Deployment Package

Overview

An application designed using Crystal Decisions' Report Designer Component (RDC) requires that multiple DLLs are present on a client computer for the application to install and run successfully. Although most of the DLLs are Crystal DLLs there are some required Microsoft DLLs as well.

This document discusses the process of creating a RDC deployment package to distribute a RDC application.

Contents

INTRODUCTION	2
RUNTIME FILES	2
<i>Merge modules</i>	3
<i>Runtime Help file</i>	3
<i>Methods for creating a RDC Deployment Package</i>	3
METHOD 1 - USING THE CRYSTAL REPORTS MERGE MODULES	4
Creating a new Microsoft Visual Studio Installer package.....	4
Add Microsoft's Merge Modules.....	5
Add Crystal Decisions' Merge Module.....	5
METHOD 2: CREATING A VISUAL BASIC INSTALLER PACKAGE	6
<i>Dependency files for Craxdrt9.dll</i>	6
Modifying Craxdrt9.dep and Craxddrt9.dep.....	6
Alerts.....	7
U2dapp.dll and U2dpost.dll.....	7
Unicows.dll.....	8
<i>Create a new Visual Basic Installer Package</i>	8
METHOD 3: USING THE RUNTIME HELP FILE TO CREATE A DEPLOYMENT PACKAGE	10
Creating a new Microsoft Visual Studio Installer package.....	10
Add Microsoft's Merge Modules.....	11
Use the Runtime.chm file.....	11
ADDING REPORT CREATION API FUNCTIONALITY	12
<i>Adding the License Key through code</i>	12
ADDING EXTRA FILES	13
BUILDING THE INSTALLER PACKAGE	15
<i>To launch an Installer Package (.msi) file</i>	16

FINDING MORE INFORMATION	16
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	17

Introduction

This document describes the steps for creating a successful deployment package using the freely available Microsoft Visual Studio Installer 1.1 available for download from Microsoft's website at:

<http://msdn.microsoft.com/vstudio/downloads/tools/vsi11/default.asp>

Microsoft's Visual Studio Installer 1.1 comes bundled with a number of Microsoft merge modules. These merge modules include all of the required installation routines for many of the core Microsoft DLLs.

NOTE	The steps outlined in this document apply for most Installer software that supports merge modules. Please refer to Platforms.txt on the root drive of your Crystal Reports CD for a listing of supported Installer software.
-------------	---

Runtime Files

The Crystal Reports runtime files required for distribution with your application are dependent on the following:

- The method used to access the Crystal Report Engine.
- The data source used for the reports.
- The exporting options you want available to end-users.
- Any additional components you may be using such as charts, maps, additional formula language functions, and so on.

When using the RDC as the method to access the Crystal Report Engine, here is a listing of the core runtime files needed to open a report:

Crystal DLLs:

- Crqe.dll (Crystal Query Engine)
- Ufmanager.dll (Manager for loading UFLs)
- Craxdrt9.dll (Crystal Reports 9 ActiveX Designer Runtime support)
- Crviewer.dll (Crystal Reports viewer. This DLL is only needed if the application previews the report)

Other DLLs

- Atl.dll

NOTE	There are two versions of Atl.dll—one for Windows NT/2000 and above and another for Windows 9x. These files are not cross-platform compatible so it is important to distribute the proper
-------------	--

	<p>version of Atl.dll with your application to avoid error registering the above files.</p> <p>The Windows 9x Atl.dll (ANSI) can be found on the Crystal Reports CD at \Windows\System32\Redist\MS\System\Ansi\Atl.dll.</p> <p>The Windows NT/2000 Atl.dll (UNICODE) can be found on the Crystal Reports CD at \Windows\System32\Redist\MS\System\Atl.dll.</p>
--	--

- Msvcp60.dll
- Msvcr7.dll
- Riched20.dll
- Unicows.dll (only if deploying to Windows 9x)

These are the required runtime DLLs to just open a report. Additional runtime files are required for database connectivity, previewing the report and exporting.

Crystal Decisions provides two resources to assist you in gathering these additional runtime files:

- Merge modules
- Runtime help file

Merge modules

Use merge modules when deploying Crystal Reports 9 applications. Using these merge modules ensure that all of the required Crystal DLLs are added to the deployment package.

A merge module package is available that includes 5 merge modules files (.msm) and 1 Readme.txt. These merge modules were not included with the release of Crystal Reports 9.0 but they can be found on our support site by going to <http://support.crystaldecisions.com/downloads> and searching for:

Cr9rdcmmergemodules.zip

Runtime Help file

The second resource is our Runtime help file (Runtime.chm). This compiled help file contains step-by-step instructions to help you determine which DLLs your application needs. This help file is installed by default to the following folder:

C:\Program Files\Crystal Decisions\Crystal Reports 9\Developer Files\Help\En

Methods for creating a RDC Deployment Package

There are three methods for creating a RDC deployment package. All three methods use the Microsoft Visual Studio Installer 1.1. These three methods are:

- Using the Report Designer Component merge modules
- Creating a Visual Basic Installer Package
- Using the Runtime help file to create a deployment package

NOTE	<p>Microsoft DLLs are required in all three methods. These Microsoft DLLs must be correctly installed for the RDC application to work.</p> <p>If you require further information regarding the distribution of any of the Microsoft DLL's listed in this document please contact Microsoft Technical Support.</p>
-------------	---

Method 1 - Using the Crystal Reports Merge Modules

This method describes how to create a RDC deployment package using Crystal Decisions' and Microsoft merge modules.

Creating a new Microsoft Visual Studio Installer package

To use the merge modules from Crystal Decisions and Microsoft, you need to first open Microsoft Visual Studio Installer and create an **Empty Installer Project**.

To open Microsoft Visual Studio Installer and create an **Empty Installer Project**:

1. Go to the **Start menu** and select **Programs**. Click **Microsoft Visual Studio 6.0** and select **Microsoft Visual Studio 6.0 Enterprise Tools**.
2. Click **Visual Studio Installer**.
3. The **New Project** dialog box appears. Expand the **Visual Studio** node, and click **Visual Studio Installer Projects**.
4. Select the **Empty Installer** icon and then click **OK**.



NOTE	<p>You can also run the Visual Studio Installer by launching Microsoft Visual J++ or Microsoft Visual InterDev.</p>
-------------	---

Add Microsoft's Merge Modules

1. After creating the **Empty Installer Project**, go to the **Project** menu and select **Add Merge Modules**.
2. In the **Browse for Merge Modules File** dialog box, browse to **C:\Program Files\Microsoft Visual Studio\Common\Tools\VSInst\BuildRes**
3. Choose the following merge modules and click **Open**:
 - Atl.msm
 - Comcat.msm
 - Comctl32.msm
 - Mdac.msm
 - Msvbvm60.msm
 - Msvcp60.msm
 - Msvcr70.msm

These 7 merge modules are now listed under the **Files** folder in the **Project Explorer**

NOTE	<p>The Mdac.msm file is not always required, however including this file is strongly recommended as it determines the client's version of MDAC.</p> <p>This MSM file does not actually install MDAC but will inform the user that they need to upgrade before the installation can proceed.</p>
-------------	---

Add Crystal Decisions' Merge Module

1. After adding Microsoft's merge modules, go to the **Project** menu and select **Add Merge Modules**.
2. In the **Browse for Merge Modules File** dialog box, browse to the directory where you downloaded the Crystal Reports 9 merge module (**cr9rdcmmergemodules.zip**)
3. Select the following required merge modules:
 - License.msm
 - Reportengine.msm
 - Rdcruntime.msm.

4. Add any additional files to your applications such as reports, physical database and custom DLLs.

For information on how to add additional files, refer to the [Adding Extra Files](#) section in this document.

5. Make any additional changes to the installation package and then **build** the Installer Package (.msi) file.

For information on how to build the Installer Package (.msi) file, refer to the [Building the Installer Package](#) section in this document.

After building the Installer Package you can distribute it to your client computers.

NOTE	<p>There are 5 Crystal Decisions Merge Modules available. These include:</p> <ul style="list-style-type: none"> • Reportengine.msm – Contains the core files that comprise the Crystal Reports engine. This merge module is always required. • Rdcruntime.msm - Contains the Report Designer Component (RDC) runtime files and is required by all RDC applications using the Crystal Reports ActiveX Designer Runtime library or the Crystal Report ActiveX Viewer control. • Rdcdesigntime.msm - This contains the Embeddable Report Designer Control and it's associated runtime files. This merge module is only required if the Embeddable Designer Control (crdesignerctrl.dll) is used in the application. • Mapping.msm - This contains the files required for displaying geographic maps inside Crystal Reports. This merge module is only required if reports are using mapping. • License.msm - This merge module is used to apply the Report Creation API license keycode during installation. It exposes a custom property called License Key that must be populated with the keycode found on the Crystal Reports 9 CD envelope. For information about Report Creation API, please refer to the Adding Report Creation API Functionality section in this document. <p>License.msm is a configurable merge module. It is designed to prompt for a license key when it is added to an MSI-based installation package. Not all installers* are designed to work with configurable merge modules, nevertheless license.msm is a required module and should always be included. Even if your installer does not prompt for a keycode when including license.msm, the installation should still install without error when it is included.</p> <p>* InstallShield 8, and Microsoft's Visual Studio .NET installers can take advantage of configurable merge modules. Wise 4.0, Visual Studio Installer 1.1 and older versions of InstallShield are not able to take full advantage of configurable merge modules.</p>
-------------	--

METHOD 2: Creating a Visual Basic Installer Package

This method uses the Visual Basic (VB) Installer Package Wizard (found in Microsoft Visual Studio Installer) to deploy a RDC application.

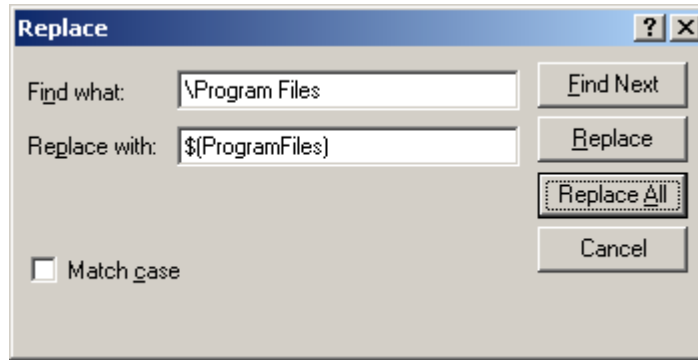
Dependency files for Craxdrt9.dll

Before using the VB Installer Package Wizard, some changes are required for **Craxdrt9.dep** and **Craxddrt9.dep** (the dependency files for **Craxdrt9.dll**).

Modifying Craxdrt9.dep and Craxddrt9.dep

In order for **Craxdrt9.dep** and **Craxddrt9.dep** to successfully work with the VB Installer Package Wizard, modify **Craxdrt9.dep** and **Craxddrt9.dep** using the following steps:

1. Open **Craxdrt9.dep** in Notepad. **Craxdrt9.dep** is installed by default to
C:\Program Files\Crystal Decisions\Report Designer Component
2. Select **Replace** from the **Edit** menu.
3. In the **Find What** text box type: **'\Program Files'**.
4. In the **"Replace With"** text box type: **'\$(ProgramFiles)'**.
5. Select **Replace All** and then save this file.



- Repeat steps 1 to 5 but with **Craxddrt9.dep**. **Craxddrt9.dep** is installed by default to the following folder:

C:\Program Files\Common Files\Crystal Decisions\2.0\bin

Alerts

The VB Installer Package Wizard will look for every DLL listed in the dependency files (**Craxdrt9.dep** and **Craxddrt9.dep**) and attempt to add them to your package.

The VB Installer Package Wizard may issue an alert for each of the following files, as they may not be on your computer.

- U2dapp.dll
- U2dpost.dll
- Unicows.dll

U2dapp.dll and U2dpost.dll

These Crystal DLLs handle the exporting destination to the Application and Exchange Folders. Alerts for these files may appear in the VB Installer Package Wizard as these files are listed in the **Craxdrt9.dep** file and may not be on your computer.

You can either choose to comment out these DLLs from **Craxdrt9.dep** or install these DLLs to your computer. If you wish to comment out the DLL follow these steps:

- Open **Craxdrt9.dep** in **Notepad**. **Craxdrt9.dep** is installed to the following folder by default:

C:\Program Files\Crystal Decisions\Report Designer Component

- Search for the line:

Uses20=\Program Files\common files\Crystal Decisions\2.0\bin\u2dpost.dll

- Place a semi-colon in front of this line so that it reads:

;Uses20=\Program Files\common files\Crystal Decisions\2.0\bin\u2dpost.dll

4. Search for the line:

Uses21=\Program files\common files\Crystal Decisions\2.0\bin\u2dapp.dll

5. Place a semi-colon in front of this line so that it reads:

;\Uses21=\Program files\common files\Crystal Decisions\2.0\bin\u2dapp.dll

6. Repeat steps 1 to 5 for **Craxddrt9.dep** file. **Craxddrt9.dep** is installed by default to the following folder:

C:\Program Files\Common Files\Crystal Decisions\2.0\bin

NOTE	<p>If you chose a Typical installation of Crystal Reports 9 then the only Export Destination DLL that is installed is the Disk file destination (u2ddisk.dll).</p> <p>You may choose a Complete or Custom Installation method if you require more Export Destinations. Any extra Export Destination will need to be added separately.</p> <p>For more information, refer to the Adding Extra Files section in this document.</p>
-------------	--

Unicows.dll

Unicows.dll is required if you wish to deploy to any **Windows 9x** operating systems. Due to licensing restrictions, you must get Unicows.dll from the Microsoft web site at:

<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/psdkredist.htm>

Select "**Microsoft Layer for Unicode on Windows 95/98/ME Systems (MSLU) version 1.0**".

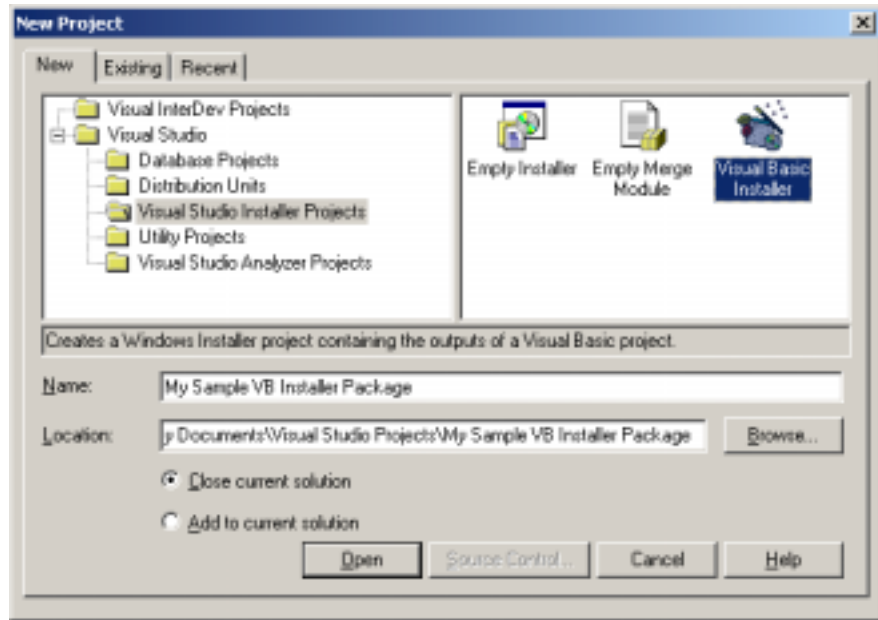
In order for the dependency files to find Unicows.dll, extract Unicows.dll to the following location:

C:\Program Files\Common Files\Crystal Decisions\2.0\bin

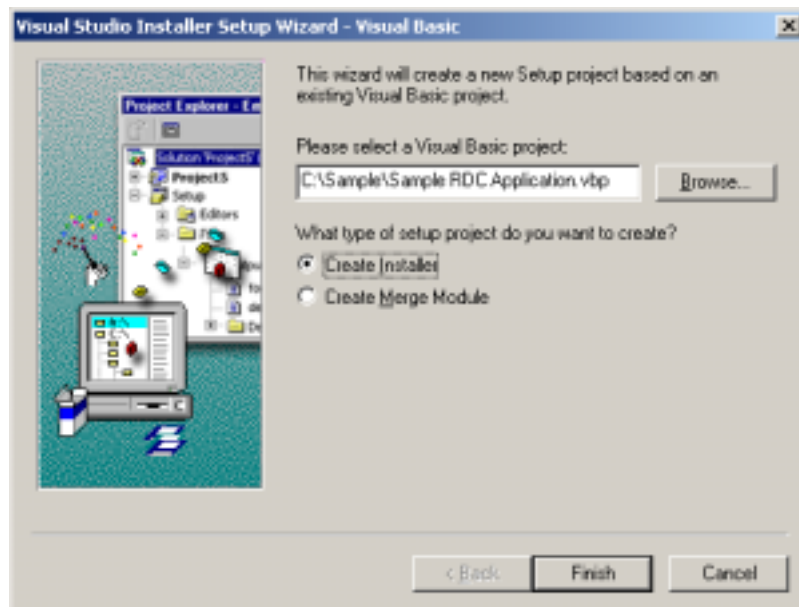
Create a new Visual Basic Installer Package

To create a new Microsoft Visual Basic Installer Package, use the following steps:

1. Go to the **Start menu** and select **Programs**. Click **Microsoft Visual Studio 6.0** and select **Microsoft Visual Studio 6.0 Enterprise Tools**.
2. Click **Visual Studio Installer**.
3. The **New Project** dialog box appears. Expand the **Visual Studio** node, and click **Visual Studio Installer Projects**.
4. Select the **Visual Basic Installation** icon and then click **OK**.



5. Browse to your Visual Basic Project and select **Finish**. This creates a new installation package.



6. Add any additional files to your applications such as reports, physical database and custom DLLs.

For information on how to add additional files, refer to the [Adding Extra Files](#) section in this document.

7. Make any additional changes to the installation package and then **build** the Installer Package (.msi) file.

For information on how to build the Installer Package (.msi) file, refer to the [Building the Installer Package](#) section in this document.

After building the Installer Package you can distribute it to your client computers.

METHOD 3: Using the Runtime Help File to Create a Deployment Package

Using the Runtime help file to create a RDC deployment package allows for the most control over the files being deployed. However, this method is not automated and must be done manually.

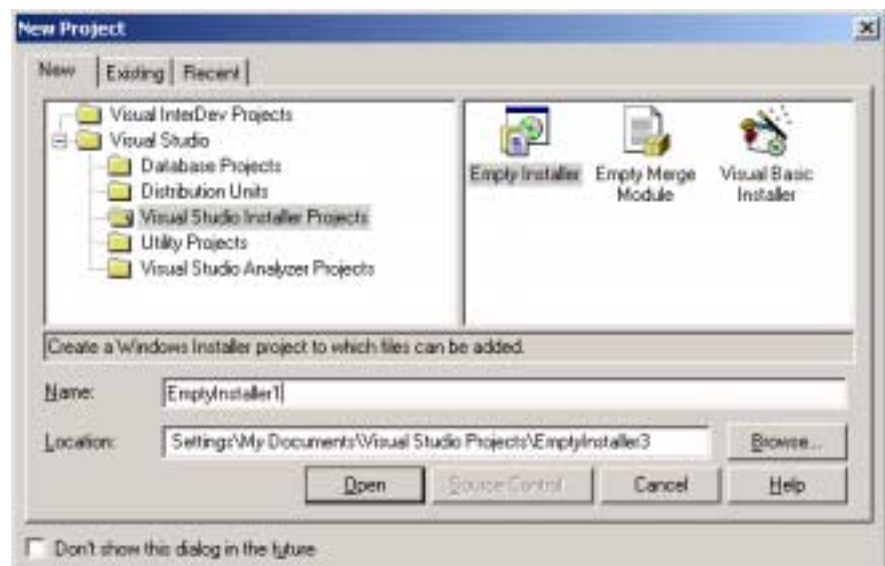
This method uses the Microsoft merge modules to add the required Microsoft DLLs to the RDC deployment package. You then need to manually select and add the Crystal runtime files to your RDC deployment.

Creating a new Microsoft Visual Studio Installer package

To use the merge modules from Microsoft and manually add the Crystal runtime files, you need to first open Microsoft Visual Studio Installer and create an **Empty Installer Project**.

To open Microsoft Visual Studio Installer and create an **Empty Installer Project**:

1. Go to the **Start menu** and select **Programs**. Click **Microsoft Visual Studio 6.0** and select **Microsoft Visual Studio 6.0 Enterprise Tools**.
2. Click **Visual Studio Installer**.
3. The **New Project** dialog box appears. Expand the **Visual Studio** node, and click **Visual Studio Installer Projects**.
4. Select the **Empty Installer** icon and then click **OK**.



NOTE

You can also run Visual Studio Installer by launching Microsoft Visual J++ or Microsoft Visual InterDev.

Add Microsoft's Merge Modules

1. After creating the **Empty Installer**, go to the **Project** menu and select **Add Merge Modules**.
2. In the **Browse for Merge Modules File** dialog box, browse to **C:\Program Files\Microsoft Visual Studio\Common\Tools\VSInst\BuildRes**
3. Choose the following merge modules and click **Open**:
 - Atl.msm
 - Comcat.msm
 - Comctl32.msm
 - Mdac.msm
 - Msvbvm60.msm
 - Msvcp60.msm
 - Msvcr.msm

These 7 merge modules are now listed under the **Files** folder in the **Project Explorer**

NOTE	<p>The Mdac.msm file is not always required, however including this file is strongly recommended as it determines the client's version of MDAC.</p> <p>This MSM file does not actually install MDAC but will inform the user that they need to upgrade before the installation can proceed.</p>
-------------	---

Use the Runtime.chm file

The Crystal Reports runtime files required for distribution with your application are dependent on the following:

- The method used to access the Crystal Report Engine.
- The data source used for the reports.
- The exporting options you want available to end-users.
- Any additional components you may be using such as charts, maps, additional formula language functions, and so on.

Use the Runtime Help file (Runtime.chm) to determine which files need to be added to your VB Installation Package.

By default, Runtime.chm is installed to **C:\Program Files\Crystal Decisions\Crystal Reports 9\Developer Files\Help\En**

NOTE	<p>For further information about runtime file requirements for the Report Designer Component (RDC) version 9, search for knowledge base article c2012421 at:</p> <p>http://support.crystaldecisions.com/kbase</p>
-------------	---

After you determine which runtime files to include in your VB Installer Package, manually add these runtime files using the instructions from the [Adding Extra Files](#) section in this document.

Make any additional changes to the installation package and then **build** the Installer Package (.msi) file.

For information on how to build the Installer Package (.msi) file, refer to the [Building the Installer Package](#) section in this document.

After building the Installer Package you can distribute it to your client computers.

Adding Report Creation API Functionality

If your application uses any of Crystal Decision's Runtime Required objects or methods, your application requires the 19 digit Report Creation API (RCAPI) license key.

- If you are using InstallShield to create the deployment package, then upon adding the license.msm merge module, you will be prompted to insert the required license key.
- You will need to add this license key at runtime using code if you use:
 - Microsoft's Visual Studio Installer,
 - Wise Installer,
 - Microsoft Visual Basic Package and Deployment Wizard
 - Crystal Runtime Help file method to create your deployment package.

NOTE	You may choose to add the license key through code regardless, as it provides protection against exposing your license key to your customers.
-------------	---

Adding the License Key through code

To add the license key through code you will need to set the **SetLicenseKeycode** method.

Applying your RCAPi license key to the **SetLicenseKeycode** method of the **Application** object will assign Report Creation permissions to your application.

The **SetLicenseKeycode** method is a hidden member of the Application object so Visual Basic's IntelliSense may not display it if you have chose to not display Hidden Members.

To set **SetLicenseKeycode**, refer to the following sample code:

```
OApplication.SetLicenseKeycode ("XXXXXX-XXXXXXX-XXXXXXX")
```

This line of code should be the very first method called by your **Application** object. For example:

```
Dim Application as New CRAXDRT.Application
```

```
Dim Report As CRAXDRT.Report
Application.SetLicenseKeycode ("XXXXX-XXXXXXX-XXXXXXX")
Set Report = Application.NewReport
```

NOTE	<p>If you have added the Crystal Reports 9 Designer to your Visual Basic application then you can call the method as follows:</p> <pre>Dim Report As New CrystalReport1 Report.Application.SetLicenseKeycode ("XXXXX-XXXXXXX-XXXXXXX")</pre>
-------------	--

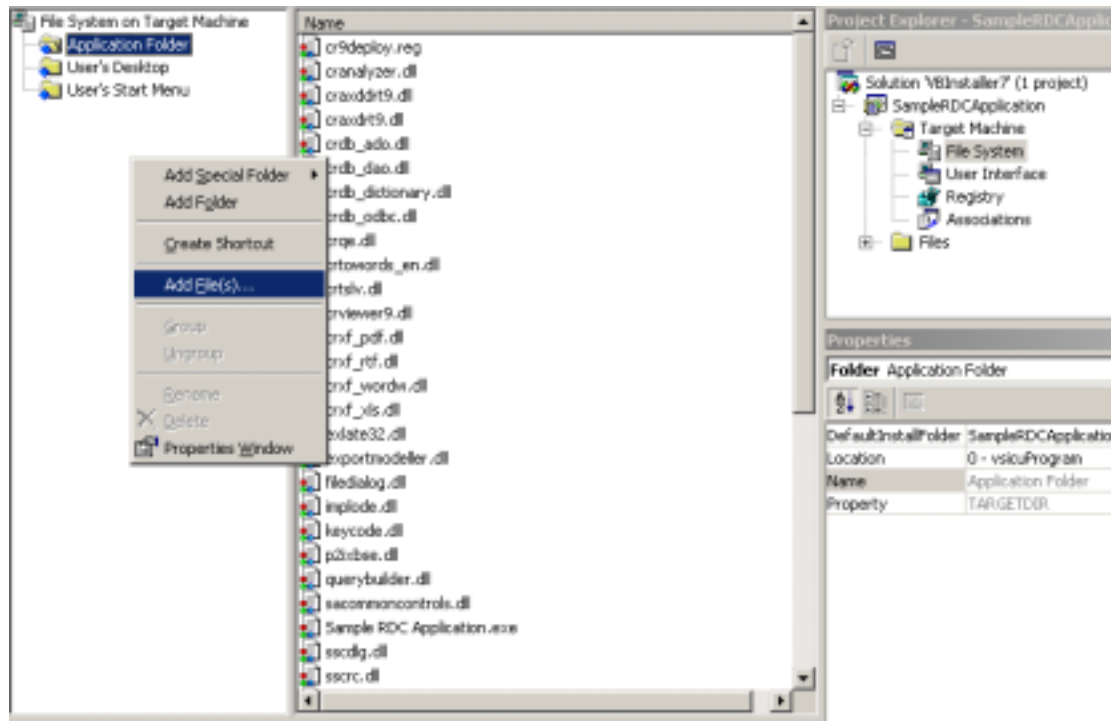
Adding Extra Files

To manually add files to your Installer Package, use the following steps:

1. Go to the **Start** menu and select **Programs**. Click **Microsoft Visual Studio 6.0** and select **Microsoft Visual Studio 6.0 Enterprise Tools**.
2. Click **Visual Studio Installer**.
3. Open the solution containing your Visual Studio Installer project.
4. In the **Project Explorer**, expand the **Target Computer** node under your installer project.
5. Double-click **File System** in the **Target Computer** node.
6. In the **File System** editor, right-click the folder you want to add files to. You may choose to add another folder (such as Windows System) if you choose.

The **File System** editor in Visual Studio Installer gives you a way to configure your application files on the target computer while you add them to the installer project.

7. Select **Add File(s)** from the context menu.

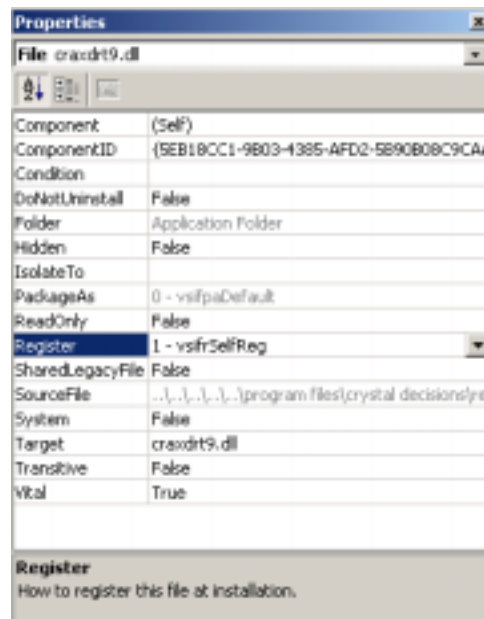


8. In the **Browse for Files** dialog box, navigate to and select the files you want to add. Click **Open**.

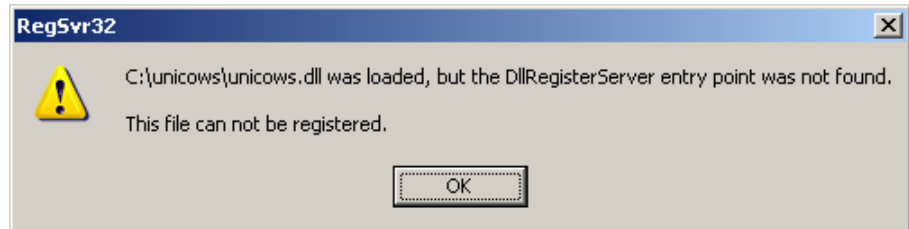
The File System editor displays the selected files you added to the folder. The files are also listed in the installer project Files node in the Project Explorer.

9. If the added file needs to be registered right-click on the file and choose **Properties Window**.

This will display the Property Window for the specific file. Select the property **Register** and change the value to **1-vsifrSelfReg**.



If you are unsure if a DLL needs to be registered, try registering it on the development computer. If a similar error message like the following appears, then the DLL does not require registration.

**NOTE**

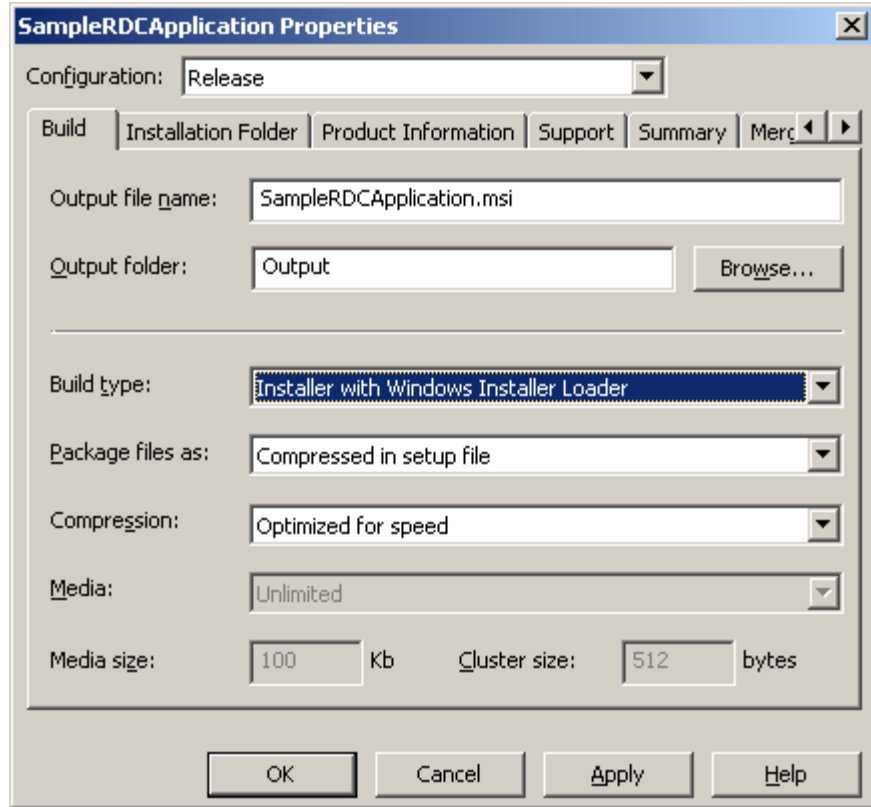
None of Crystal Database, Export Destinations or Export Formats DLLs requires registration.

Building the Installer Package

After you configure all elements of an application's installation in your installer roject (by specifying folder and file locations, file associations, registry actions, and so on), you must build the project into an Installer Package (.msi) file. You can then distribute the .msi file to users who want to install your application.

To build an Installer Package (.msi) file:

1. Open the solution containing your Visual Studio Installer project.
2. In the **Project Explorer**, select your installer project.
3. Under the **Build** menu, change the **Build Configuration** from **Debug** to **Release**. You may run the build in debug mode if you want to ensure it compiles.
4. In the **Project Explorer**, right-click your project and select **Properties**. In the Properties dialog box, under the **Build** tab, ensure the **Build type** value is set to either:
 - Installer
 - Installer with Windows Installer Loader



5. Change the **Output Folder** value to the folder where you want the .msi file to be created. Click **OK**.
6. In the **Project Explorer**, right-click your project and click **Build**.

To launch an Installer Package (.msi) file

For development and debugging purposes, the best way to launch your Installer Package (.msi) file is from the Microsoft development environment. For example:

1. Open the solution containing your Visual Studio Installer project.
2. In the **Project Explorer** window, right-click the installer project you want to launch.
3. Select **Launch Installer** from the **Project** menu.

You may also double-click the .msi file generated to start the installation

For more information on customizing the installation package please refer to the Visual Studio Installer Help guide

Finding More Information

Search for the following knowledge Base articles at <http://support.crystaldecisions.com/kbase>:

C2012421 - Runtime File Requirements for the Report Designer Component version 9

C2011871 - Where can you find Unicows.dll?

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>