

Report Designer Component 10

Creating an RDC Deployment Package

Overview

An application designed using Business Objects' 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 Reports 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>	<i>3</i>
<i>Customizing the Database Driver Installation Options (Optional)</i>	<i>4</i>
<i>Runtime Help file</i>	<i>5</i>
METHODS FOR CREATING A RDC DEPLOYMENT PACKAGE	5
METHOD 1 - USING THE RDC MERGE MODULES	5
Creating a new Microsoft Visual Studio Installer package	5
Adding RDC Merge Modules	6
METHOD 2: CREATING A VISUAL BASIC INSTALLER PACKAGE	7
<i>Dependency files for Craxdrt.dll.....</i>	<i>7</i>
Modifying Craxdrt.dep and Craxddrt.dep	7
Alerts.....	8
U2dapp.dll and U2dpost.dll	8
Create a new Visual Basic Installer Package	9
METHOD 3: USING THE RUNTIME HELP FILE TO CREATE A DEPLOYMENT PACKAGE	10
Creating a new Microsoft Visual Studio Installer package	10
Use the Runtime.chm file	11
ADDING REPORT CREATION API FUNCTIONALITY	12
Adding the License Key through code.....	12
ADDING EXTRA FILES.....	13
BUILDING THE INSTALLER PACKAGE	14
To launch an Installer Package (.msi) file	15
CRYSTAL REPORTS 10 KEYCODE	16
FINDING MORE INFORMATION	16

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 Docs\ folder 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 Reports 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 and additional formula language functions.

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

Crystal Reports DLLs:

- Crqe.dll (Crystal Reports Query Engine)
- Ufmanager.dll (Manager for loading UFLs)
- Craxdrt.dll (Crystal Reports ActiveX Designer Runtime 10)
- Crviewer.dll (Crystal Reports viewer. This DLL is only needed if the application previews the report)

Other DLLs

- Atl.dll

NOTE	<p>There are two versions of Atl.dll—one for Windows NT/2000 and above and another for Windows 9x.</p> <p>These files are not cross-platform compatible so it is important to distribute the proper version of Atl.dll with your application to avoid error registering the above files.</p> <p>Due to licensing restrictions, you must get ATL from Microsoft. Please see the MS KB article for more info: http://support.microsoft.com/default.aspx?scid=kb;en-us;259403</p>
-------------	---

- Msvcp60.dll
- Msvcr7.dll
- Riched20.dll

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

Business Objects provides two resources to assist you in gathering these additional runtime files:

- Merge modules
- Runtime help file

Merge modules

Using merge modules is the recommended method when deploying Crystal Reports 10 applications. Using these merge modules ensure that all of the required Crystal Reports DLLs are added to the deployment package.

There are five merge modules files for Crystal Reports 10 RDC deployment. They include:

NOTE	<p>These merge modules were not included with the release of Crystal Reports 10, but they can be found by searching for "cr10_rdc_merge_modules.zip" on our support site at:</p> <p>http://support.businessobjects.com/search</p>
-------------	--

- **CrystalReports10_rdc_reportengine.msm** - This merge module contains the Crystal Reports database and exporting DLLs. Files in this module include (but are not limited to) Crqe.dll, Crdb_ado.dll, Crdb_odbc.dll, Crxf_pdf.dll, Crxf_xls.dll, U2ddisk.dll and U2dapp.dll.

This merge module is now configurable in Crystal Reports 10. Installation IDEs that support configurable merge modules (such as the VS .NET installer, Wise and InstallShield 8+) allow you to choose which Crystal Reports database drivers to install to your client's computer.
- **CrystalReports10_rdc_runtime.msm** - This merge module contains the core RDC engine required for all RDC applications. Files in this module include (but are not limited to) Craxdrt.dll, Crviewer.dll, Cselexpt.ocx, Sviewhlp.dll, Swebrs.dll.
- **CrystalReports10_rdc_license.msm** - This required merge module is configurable and requires a valid Crystal Reports 10 keycode. . See section [Crystal Reports 10 Keycode](#) for more information.
- **CrystalReports10_rdc_designtime.msm (optional)** - Include this merge module for applications using the Report Wizard component, or the Embeddable Designer Control. These components may require additional licensing.
- **CrystalReports10_maps.msm (optional)** - Include this merge module if the report contains any maps.

NOTE	<p>CrystalReports10_rdc_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, however, CrystalReports10_rdc_license.msm is a required module and should always be included.</p> <p>Even if your installer does not prompt for a keycode when including CrystalReports10_rdc_license.msm, the installation should still install without error when it is included.</p> <p>* InstallShield 8, and Microsoft Visual Studio .NET installer 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. To supply the keycode with an installer that does not support configurable merge modules you can use a tool like ORCA, or add a Command Line Install to your package.</p>
-------------	--

Customizing the Database Driver Installation Options (Optional)

Your deployment 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_rdc_reportengine.msm** merge module.

When adding **CrystalReports10_rdc_reportengine.msm** to your setup project a prompt with a list of database drivers appears.

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_rdc_reportengine.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_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
p2xbse	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

Runtime Help file

The second resource is the Crystal Reports Runtime help file (Runtime.chm). This compiled help file contains 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 10\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 include:

- Using the Crystal Reports merge modules
- Creating a Visual Basic Installer Package
- Using the Runtime help file to create a deployment package

NOTE

Microsoft DLLs are required in all three methods. These Microsoft DLLs must be correctly installed for the RDC application to work.

If you require further information regarding the distribution of any of the Microsoft DLL's listed in this document please contact Microsoft Technical Support.

Method 1 - Using the RDC Merge Modules

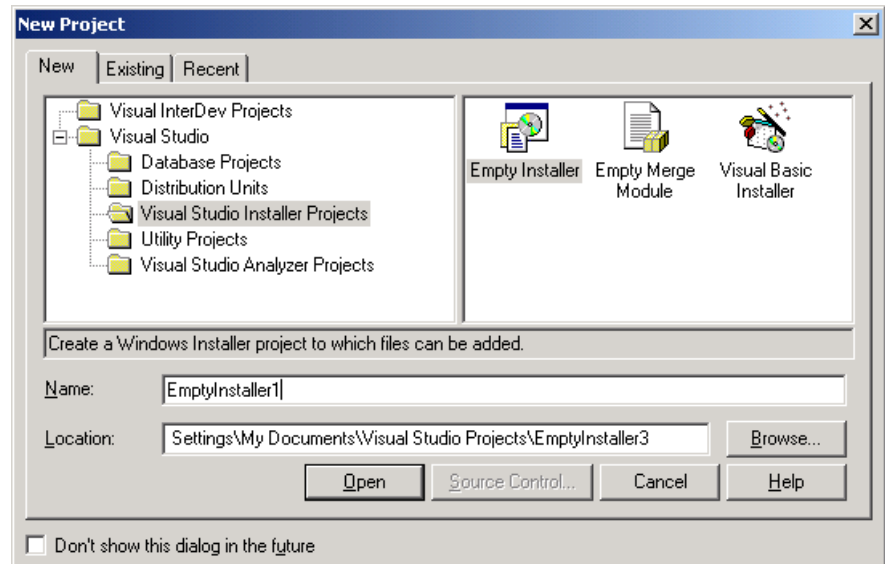
This method describes how to create a RDC deployment package using the RDC merge modules and the Microsoft merge modules.

Creating a new Microsoft Visual Studio Installer package

To use the merge modules from Business Objects 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**

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

Adding RDC Merge Modules

1. After adding Microsoft's merge modules, go to the Project menu and select Add Merge Modules.

In the Browse for Merge Modules File dialog box, browse to the folder where you downloaded the RDC merge modules (cr10_rdc_merge_modules.zip)

2. Select the following required merge modules:
 - CrystalReports10_rdc_license.msm
 - CrystalReports10_rdc_reportengine.msm
 - CrystalReports10_rdc_runtime.msm

3. 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.

4. 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 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 Craxdr.t.dll

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

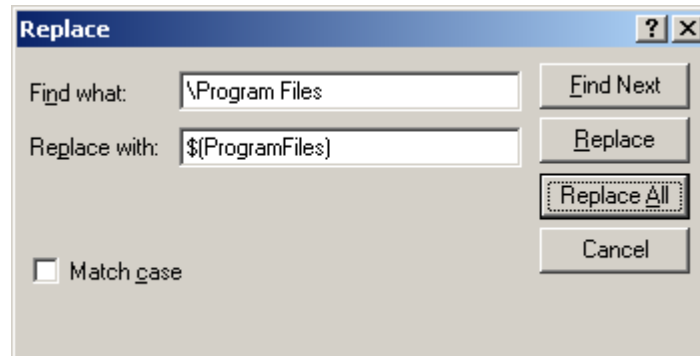
Modifying Craxdr.t.dep and Craxddrt.dep

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

1. Open **Craxdr.t.dep** in Notepad. **Craxdr.t.dep** is installed by default to:

C:\Program Files\Common Files\Crystal Decisions\2.5\bin

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.



6. Place a semi-colon in front of the following lines (to comment them out) in **Craxdr.t.dep**:

```
Uses1=$(ProgramFiles)\Common Files\Crystal Decisions\2.5\bin\Unicows.dll
```

```
[unicows.dll]
```

```
Dest=$(CommonFiles)\Crystal Decisions\2.5\bin
```

```
Uses1=
```

7. Search for the string 'Uses31'. Below this line, add the following line:

Uses32=\$(System32)\cxlibw-1-6.dll

8. Repeat steps 1 to 7 but with **Craxddrt.dep**. **Craxddrt.dep** is installed by default to the following folder:

C:\Program Files\Common Files\Crystal Decisions\2.5\bin

Place a semi-colon in front of the following line (to comment it out) in Craxddrt.dep:

Uses11=C:\$(ProgramFiles)\Common Files\Crystal Decisions\2.5\bin\ssrcr.dll

Alerts

The VB Installer Package Wizard will look for every DLL listed in the dependency files (**Craxdrtd.dep** and **Craxddrt.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 Reports 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 **Craxdrtd.dep** file and may not be on your computer.

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

1. Open **Craxdrtd.dep** in Notepad. **Craxdrtd.dep** is installed to the following folder by default:

C:\Program Files\Common Files\Crystal Decisions\2.5\bin

2. Place a semi-colon in front of the following lines:

Uses20=\$(ProgramFiles)\Common Files\Crystal Decisions\2.5\bin\u2dpost.dll
Uses21=\$(Program Files)\Common Files\Crystal Decisions\2.5\bin\u2dapp.dll

3. Repeat steps 1 and 2 for **Craxddrt.dep** file. **Craxddrt.dep** is installed by default to the following folder:

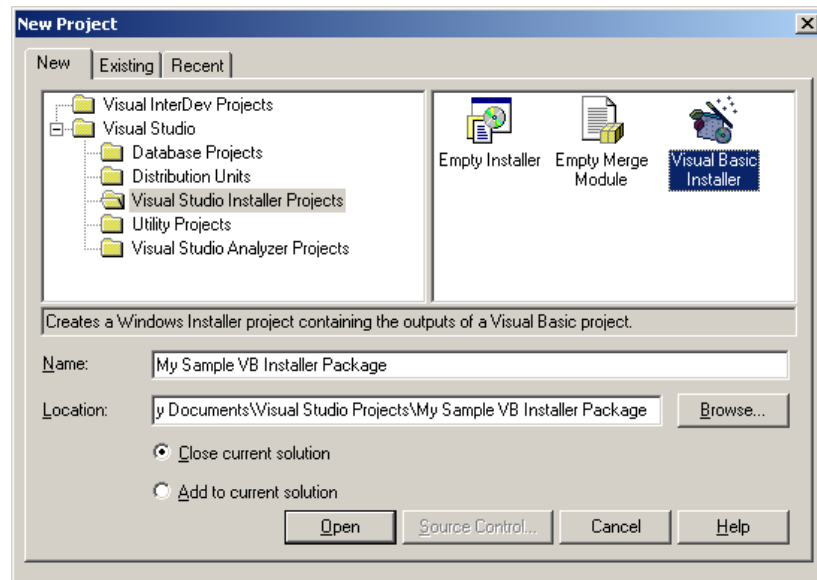
C:\Program Files\Common Files\Crystal Decisions\2.5\bin

NOTE	<p>If you chose a Typical installation of Crystal Reports 10 then the only Export Destination DLL that is installed to the Disk file destination is 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>
-------------	--

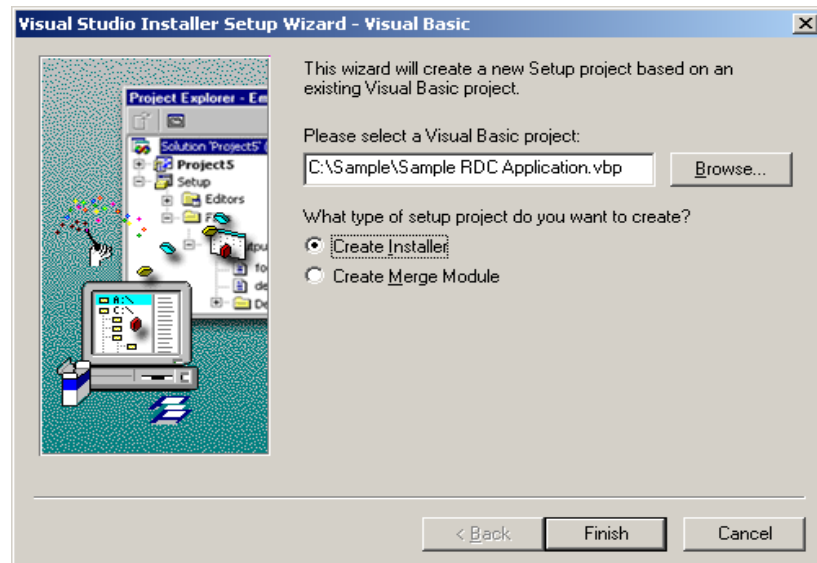
Creating a new Visual Basic Installer Package

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.



5. 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.

6. 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 an RDC deployment package allows for the most control over the files being deployed. However, this method is not automated and must be done manually.

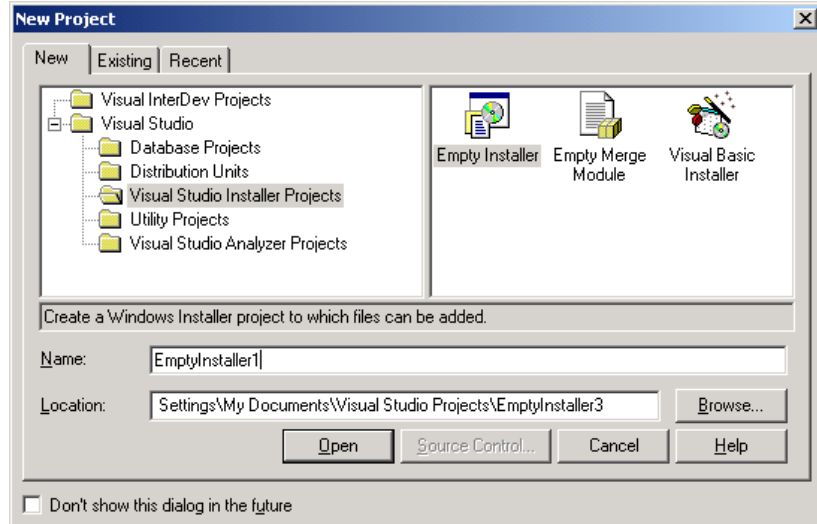
This method uses the Microsoft merge modules to add the required Microsoft DLLs to the RDC deployment package. You then need to manually select and add the Crystal Reports runtime files to your RDC deployment.

Creating a new Microsoft Visual Studio Installer package

To use the merge modules from Microsoft and manually add the Crystal Reports 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.

Using 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 Reports 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 the **C:\Program Files\Crystal Decisions\Crystal Reports 10\Developer Files\Help\En** folder.

NOTE

An updated Runtime.chm is available on the Business Objects support site. To obtain this updated file, refer to knowledge base article c2014956 located at:

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

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 Report Creation API (RCAPI) calls, a 19-digit RCAPI license key is required. This license key is available with Crystal Reports 10 Advanced Edition.
- If you are using InstallShield to create the deployment package, then upon adding the CrystalReports10_rdc_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 Visual Studio Installer
 - Wise Installer
 - Microsoft Visual Basic Package and Deployment Wizard
 - Crystal Reports Runtime Help file method to create your deployment package

NOTE	<p>You may choose to add the license key through code regardless, as it provides protection against exposing your license key to your customers.</p> <p>For more information about licensing, go to: http://www.businessobjects.com/products/reporting/crystalreports/licensing</p>
-------------	---

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 chosen to not display Hidden Members.

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

```
Application.SetLicenseKeycode ("XXXXXX-XXXXXXXX-XXXXXXXX")
```

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 ("XXXXXX-XXXXXXXX-XXXXXXXX")
Set Report = Application.NewReport
```

NOTE	<p>If you have added the Crystal Reports 10 Designer to your Visual Basic application then you can call the method as follows:</p> <pre>Dim Report As New CrystalReport1 Report.Application.SetLicenseKeycode ("XXXXXX-XXXXXXX-XXXXXXX")</pre>
-------------	--

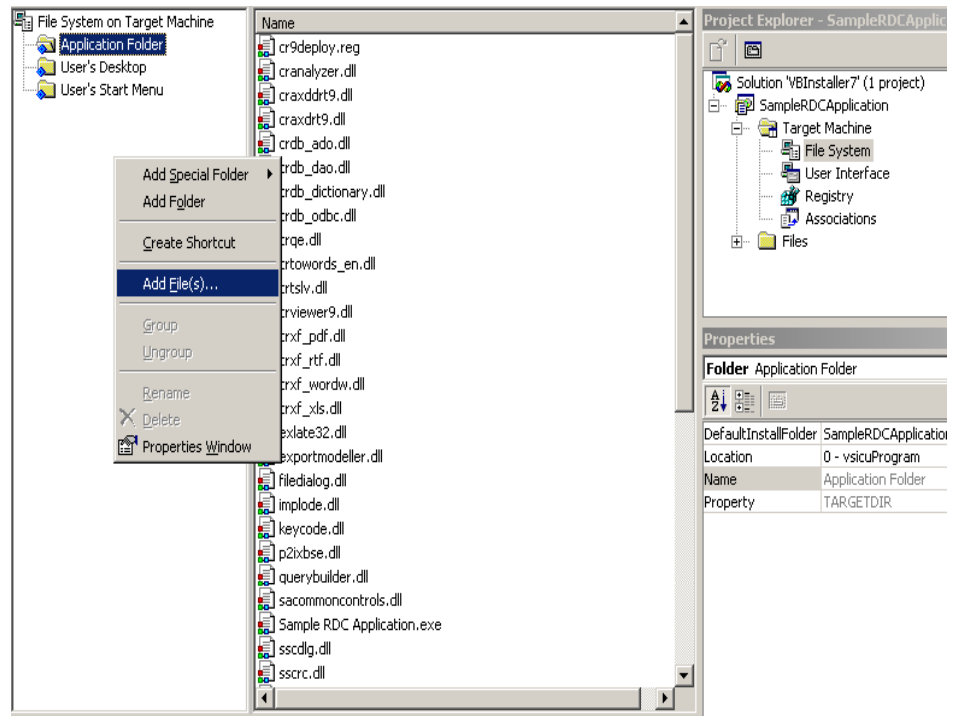
Adding Extra Files

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.

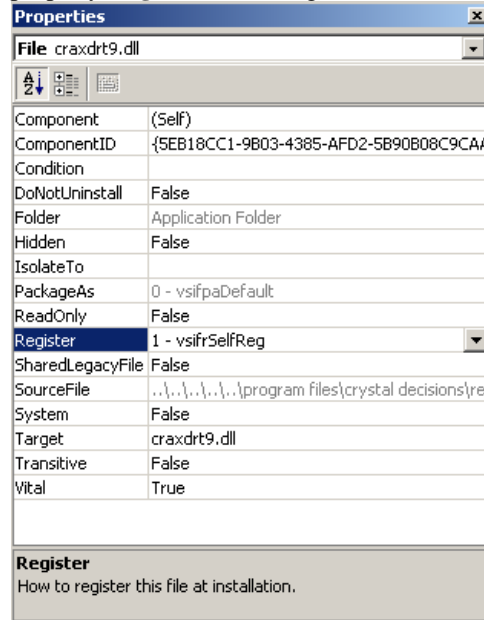


8. In the **Browse for Files** dialog box, navigate to and select the files you want to add. Click **Open**.

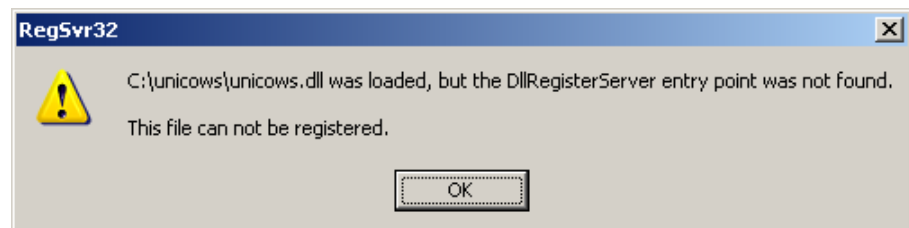
The File System editor displays the selected files you added to the folder. The files are also listed in the installer project Files node in the Project Explorer.

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 Reports Database, Export Destinations or Export Formats DLLs requires registration.

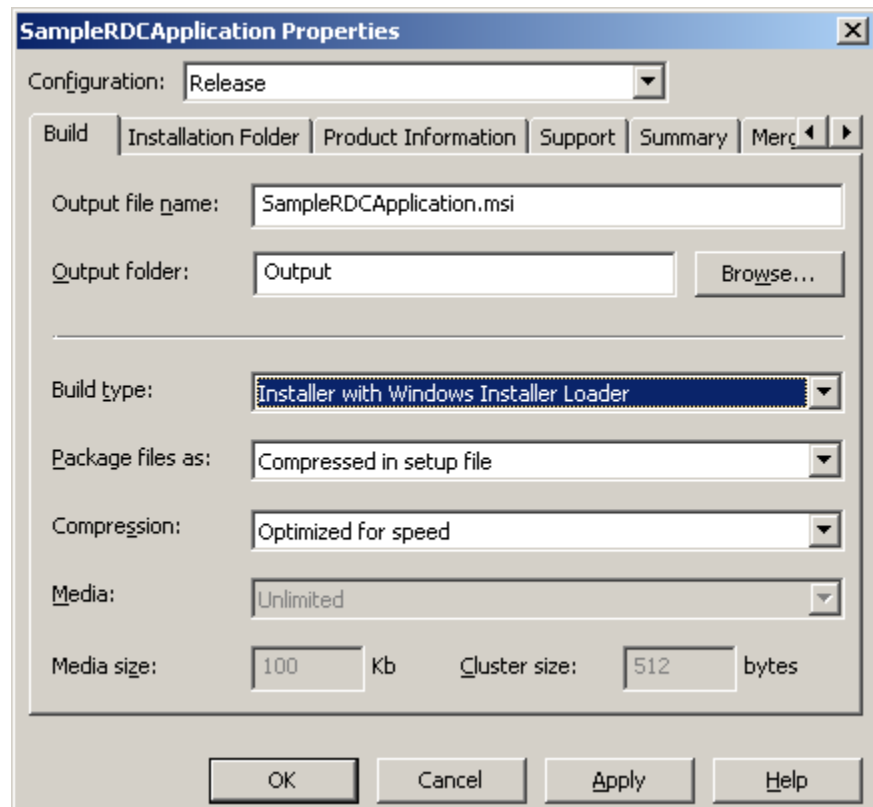
Building the Installer Package

After you configure all elements of an application's installation in your installer project (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**.

Launching 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

Crystal Reports 10 Keycode

A Crystal Reports keycode is used to validate the licensing for the application. The CrystalReports10_rdc_license.msm merge module requires 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. To access the Crystal License Manager, select the 'Crystal Reports 10 Tools' program group in the Start menu.

Finding More Information

For more information, search for the following articles at <http://support.businessobjects.com/search>

- c2014897 - Dependency reference to Cxlibw-1-6.dll is missing in RDC.dep files in CR 10
- c2014817 - Where to find the merge modules for Crystal Reports 10
- c2014956 - Crystal Reports Developer Help files are out of date in Crystal Reports 10

► 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