Report Designer Component 10

Creating an RDC Deployment Package with InstallShield 8

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 an RDC deployment package with InstallShield 8 to distribute an RDC application.

Contents

Introduction

This document describes the steps for creating an RDC deployment package using InstallShield 8.

InstallShield 8 comes bundled with a number of 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 InstallShield 8 (http://www.installshield.com)
	Please refer to **Platforms.txt** on the **Docs** 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 customers.

- Any additional components you may be using such as charts, maps, and additional formula language functions.

When using the RDC 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)

- 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

- Msvcp60.dll

- Msvcrt.dll

- Riched20.dll

- 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 version of Atl.dll with your application to avoid error registering the above files.
	Due to licensing restrictions, you must get ATL from Microsoft. For more information, go to http://support.microsoft.com and download knowledge base article 259403.

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:

- Runtime Help file

- Merge modules

Runtime Help file

This compiled help file (runtime.chm) 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 10\Developer Files\Help\En

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 DLLs are added to the deployment package.

A merge module package is available that includes 5 merge modules files (.msm) and 1 Readme file.

These merge modules were not included with the release of Crystal Reports 10.0 but they can be found downloaded by visiting http://support.businessobjects.com/search and downloading **cr10_rdc_merge_modules.zip.**

NOTE	There are 5 Report Designer Component merge modules available. These include:
	• **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+) will allow you to choose which of the 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_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.
	• **CrystalReports10_rdc_license.msm** - This required merge module is configurable but requires a valid Crystal Reports 10 keycode. However, if no Report Creation API calls or components are used in the application, a keycode is not required. 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. * 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.

Methods for creating an RDC Deployment Package

There are two methods for creating an RDC deployment package with InstallShield. These two methods are:

• Using the RDC merge modules

• Using the Runtime Help file

NOTE	Microsoft DLLs are required in both 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 DLLs listed in this document, please contact Microsoft Technical Support.

Method 1 - Using the RDC Merge Modules

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

Creating a new InstallShield 8 package

To use the RDC and Microsoft merge modules, open InstallShield and create a **Basic MSI Project** using the following steps:

1. Add the following files to your **c:\Program Files\Common Files\Merge Modules** folder. By default, InstallShield will look to this folder for merge modules to display. If you don't put the files here you will have to add your folder to the list of folders InstallShield checks for merge modules.

    *   CrystalReports10_rdc_reportengine.msm

    *   CrystalReports10_rdc_license.msm

    *   CrystalReports10_rdc_designtime.msm

    *   CrystalReports10_rdc_runtime.msm

    *   CrystalReports10_rdc_maps.msm

2. Go to the **Start** menu and click **Programs.** Click **InstallShield** and then click **Developer.**

3. Go to the **File** menu and click **New** then click **Basic MSI Project.** The **New Project** dialog box appears.

4. Click the **Browse** button to choose a folder location and filename for your project.

5. Set up general features of the MSI project.

6. Under **Application Data** click **Redistributables**. On the right side you can now select some of the RDC merge modules to add to the project.

Add the RDC Merge Modules

After adding Microsoft's merge modules you can begin adding the RDC merge modules.

1. Expand the **Application Data** folder and then click the **Redistributables** icon.

2. On the right hand side you will see our merge modules listed as:

    *   License (CrystalReports10_rdc_license.msm)

    *   Mapx Merge Module (CrystalReports10_maps.msm)

    *   Rdcdesigntime (CrystalReports10_rdc_designtime.msm)

    *   Rdcruntime (CrystalReports10_rdc_runtime.msm)

    *   Reportengine (CrystalReports10_rdc_reportengine.msm)

☐ 🪆 ADO Data Control 6.0
☐ 🪆 ATL 3.0
☐ 🪆 ATL 7.0
☐ 🪆 DAO 3.50
☐ 🪆 DAO 3.60
☐ 🪆 DCOM 95
☐ 🪆 InstallShield Scripting Engine 8.0 (See Comment for usage)
☐ 🪆 InstallShield Update Service
☑ 🪆 license
☑ 🪆 mapx Merge Module
☐ 🪆 MFC 6.0
☐ 🪆 MFC 7.0
☐ 🪆 MFC European Resources 7.0
☐ 🪆 MFC Far Eastern Resources 7.0
☐ 🪆 MFC Unicode 6.0

☐ 🪆 Microsoft Windows Common Controls-2 (6.0)
☐ 🪆 Microsoft Windows Common Controls-3 (6.0)
☐ 🪆 Microsoft Winsock Control 6.0
☐ 🪆 OLE DB 2.1
☑ 🪆 rdcdesigntime
☑ 🪆 rdcruntime
☑ 🪆 reportengine
☐ 🪆 VBScriptRuntime
☐ 🪆 VC User ATL71 RTL X86 ---
☐ 🪆 VC User CRT71 RTL X86 ---
☐ 🪆 VC User MFC71 Loc RTL X86 ---
☐ 🪆 VC User MFC71 RTL X86 ---
☐ 🪆 VC User STL71 RTL X86 ---
☐ 🪆 Visual J# .NET Redistributable Merge Module
☐ 🔵 Windows CE

NOTE	If you don't see the merge modules listed above when you click on the **Redistributables** icon, then the Crystal Reports RDC 10 merge modules are not in the **c:\Program Files\Common Files\Merge Modules** folder.
	Copy the merge modules to the **c:\Program Files\Common Files\Merge Modules** folder, then close and re-open your InstallShield project.

3. Add any additional files to your applications such as reports, physical database, your .exe 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: Using the Runtime Help file

Using the Runtime Help file to create an RDC deployment package allows you to control what files are 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 InstallShield 8 package

To use the merge modules from Microsoft and manually add the Crystal runtime files, you need to first open InstallShield 8 and create a **Basic MSI Project**.

To open InstallShield 8 and create a **Basic MSI Project**:

1. Go to the **Start** menu and click **Programs.** Click **InstallShield** and then click **Developer.**

2. Go to the **File** menu and click **New** then click **Basic MSI Project.** The **New Project** dialog box appears.

3. Click the **Browse** button to choose a folder location and filename for your project.

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 customers.

•   Any additional components you may be using such as charts, maps, and additional formula language functions.

Use the Runtime Help file (Runtime.chm) to determine which files need to be added to your InstallShield project.

By default, Runtime.chm is installed to **C:\Program Files\Crystal Decisions\Crystal Reports 10\Developer Files\Help\En**

NOTE	There is an updated Runtime.chm file available on the Business Objects support site. For more information, search for knowledge Base articles c2014956 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 customer 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. Crystal Reports Advanced Edition includes this keycode.

- 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 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, 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 the **SetLicenseKeycode** method, use the following sample code:

```
Application.SetLicenseKeycode ("XXXXX-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
```

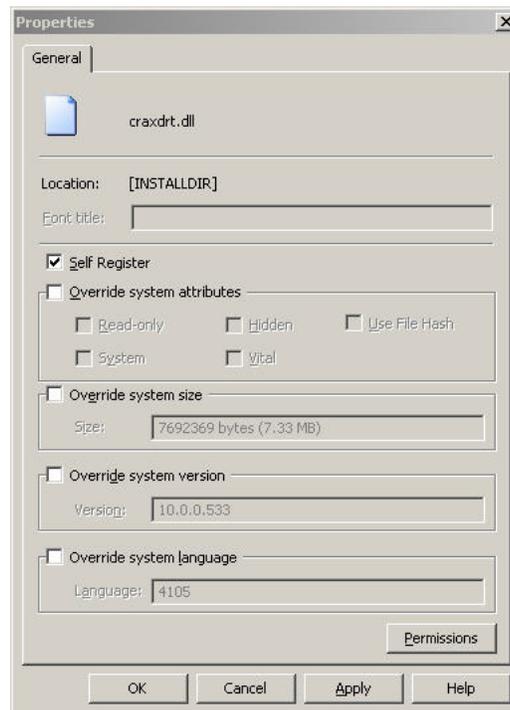| NOTE	If you have added the Crystal Reports 10 Designer to your Visual Basic application then you can call the **SetLicenseKeycode** method as follows:
	
	`Dim Report As New CrystalReport1`
	
	`Report.Application.SetLicenseKeycode ("XXXXX-XXXXXXX-XXXXXXX")`

Adding Extra Files

To manually add files to your InstallShield 8 project, use the following steps:

1. Open your InstallShield (.ism) project.

2. Expand the **Application Data** folder and then click **Files and Folders**.

3. Browse your folders for the files and folders you want to add, then drag and drop the files and folders to the bottom frame

4. If the added file needs to be registered right-click on the file and choose **Properties**. In the Property Window, select **Self Register**.

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

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.

RegSvr32

C:\unicows\unicows.dll was loaded, but the DllRegisterServer entry point was not found.

This file can not be registered.

OK

Properties

General

craxdrt.dll

Location: [INSTALLDIR]

Font title:

☑ Self Register
☐ Override system attributes
   ☐ Read-only   ☐ Hidden   ☐ Use File Hash
   ☐ System   ☐ Vital
☐ Override system size
   Size: 7692369 bytes (7.33 MB)
☐ Override system version
   Version: 10.0.0.533
☐ Override system language
   Language: 4105

Permissions

OK   Cancel   Apply   Help

Building the Installer Package

After you configure all elements of an application's installation in your InstallShield 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 your InstallShield project and go to the **Build** menu and click **Release Wizard.**

2. Give the configuration a name (or choose a previous configuration name), then click **Next.**

3. Give the release a name (or choose a previous release), then click **Next.**

4. Set your Filter settings and click **Next.**

5. Set the languages and click **Next**.

6. Choose your media type and then click **Next**.

7. Select **Compress all files** if you want all the files compressed into an .msi file.

8. Choose whether you want to include the setup launcher.

9. Keep clicking **Next** until you are ready to click **Finish** and build the release.

For more information on customizing the installation package please refer to the InstallShield 8 Help guide.

Finding More Information

For more information, search for the following knowledge base 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