

Crystal Reports Support for Delphi – History and Present



Applies to:

Crystal Reports 9.2 to Crystal Reports 2008 (12.0.x).

For more information, visit the [Business Objects homepage](#).

Summary

Crystal Reports SDKs have supported Delphi programmers since the days of the Crystal Reports 2.0 Windows Print Engine APIs. Since then, Delphi support has evolved to the use of the Crystal Reports Report Designer Component (RDC), Crystal Reports Visual Component Library (VCL) and finally Crystal Reports Assemblies for .NET (CR for .NET). This article discusses current options for Delphi developers when using Crystal Reports.

Authors: Ludek Uher, Jonathan Parminter, Trevor Dubinsky

Company: SAP

Created on: 14 May 2009

Author Bio

Ludek Uher, Jonathan Parminter, and Trevor Dubinsky are Senior Engineers with Technical Customer Assurance, SAP BusinessObjects. They specialize in the SDKs supplied with Crystal Reports and BusinessObjects Enterprise.

Table of Contents

Crystal Reports Windows Print Engine API SDK	3
Crystal Reports Report Designer Component (RDC) SDK	3
Crystal Reports RDC Runtime Distribution	3
Crystal Reports Visual Component Library for Delphi SDK	4
Crystal Reports VCL for Delphi Runtime Distribution	4
Crystal Reports .NET SDK	5
Technical Support Disclaimer	5
Creating the First Web Application	6
Directing Delphi 2007 to Use Crystal Reports 2008 Assemblies for Visual Studio .NET	6
Setting up Directories/Conditionals	7
Referencing the Crystal Reports 2008 DHTML Viewer for .NET	7
Displaying the first report in the Crystal Reports DHTML Viewer	8
Converting Applications Using Strongly Typed Reports	9
Distributing the Crystal Reports Runtime for the .NET Framework	10
Related Content	10
Copyright	11

Crystal Reports Windows Print Engine API SDK

The Crystal Reports Print Engine WIN API SDK was retired in version 9 of Crystal Reports. There are no header files available for any latter version of Crystal Reports, thus persisting with development using the Print Engine APIs is not recommended and no further discussion of this technology follows.

Crystal Reports Report Designer Component (RDC) SDK

The Crystal Reports RDC (craxdrt.dll) was retired in version 11.5 (XI release 2) of Crystal Reports. See the blog [Report Designer Component - Past, Present & Future](#).

Note that Patch Support End of Life for Crystal Reports XI release 2 is June 30, 2009, and Technical Support End of Life is June 30, 2010. For more details, see the [Product End of Life Dates](#).

For sample RDC applications select **Samples** in the **Software Type** drop down on the [downloads](#) page, then search for Delphi.

Crystal Reports RDC Runtime Distribution

Since version 9 of Crystal Reports, the RDC component can only be deployed by use of merge modules. Each version of Crystal Reports has specific merge modules and it is imperative that the correct merge modules are used. Locations of the RDC merge modules along with additional runtime help for the RDC are noted in this [Wiki](#).

Crystal Reports Visual Component Library for Delphi SDK

Crystal Reports Visual Component Library for Delphi was retired in version 11.0 (XI Release 1) of Crystal Reports. For developers who prefer to use the VCL, it is possible to update the VCL supplied by SAP BusinessObjects for Crystal Reports XI to work with more recent versions of Crystal Reports such as Crystal Reports XI Release 2 and Crystal Reports 2008. Please refer to the article [Updating the Crystal Reports VCL](#).

Version 11.0 of the VCL can be downloaded from [here](#).

The Crystal Reports VCL has not been updated since version 11.0 of Crystal Reports and thus new functionality in latter versions of Crystal Reports may not work. In particular, starting with Crystal Reports XI release 2, dynamic and cascading parameters will be an issue and in Crystal Reports 2008, reports using Flash Objects and Xcelsius will most certainly fail. For more details, see Note 1219099, “VCL component does not support prompting for dynamic/cascading parameter fields” and Note 1323599, “Is it possible to use reports created in Crystal Reports 2008 in previous versions of Crystal Reports?” in our [Notes database](#).

Note: There are no plans to update the VCL.

For sample VCL applications select **Samples** in the **Software Type** drop down on the [downloads](#) page, then search for Delphi.

Crystal Reports VCL for Delphi Runtime Distribution

There are no merge modules for the VCL. The recommended way of distributing the Crystal Reports runtime, when using the VCL, is to use either the RDC merge modules, or the merge modules for the Crystal Reports assemblies for VS .NET. Note, that the RDC merge modules do not include the crpe32.dll which is the report engine and is thus integral to correct functioning of your deployed application. It is up to the developer to include the correct crpe32.dll in the deployment project and to ensure that the crpe32.dll is installed into the correct Crystal Reports bin directory. See the [Crystal Reports Report Designer Component Runtime Distribution - Versions pre 8.0.x to 12.0](#) wiki for correct RDC merge modules.

The wiki [Crystal Reports for Visual Studio .NET Runtime Distribution - Versions 9.1 to 12.0](#) lists the correct merge modules for Crystal Reports assemblies for VS .NET.

Crystal Reports .NET SDK

Crystal Reports 2008 can be used in Delphi 2007 for .NET. If necessary, an [evaluation version of Crystal Reports 2008](#) can be downloaded.

Note that it should be possible, using the instructions below, to use Crystal Reports version 11.5 (XI Release 2) with Delphi 2007. As Delphi 2007 uses framework 2.0, Crystal Reports assemblies of version 11.5.3700.0 would have to be used. However, the use of Crystal Reports XI Release 2 has not been attempted as part of this article.

This article does not discuss Crystal Reports in Delphi Chrome.

Technical Support Disclaimer

Delphi 2007 is not on the list of [supported platforms](#) for any version of Crystal Reports.

Since Delphi 2007 is not on the supported platforms list, no issues with the Crystal Reports SDK can be escalated for a resolution with R&D, unless these are duplicated in a supported environment. E.g.; Microsoft Visual Studio .NET. The steps below have been performed successfully on two test systems. This however is not meant to be a guarantee, implied or otherwise, that your implementation will be successful. Any feedback regarding this article is welcome via the [Crystal Reports Development – Other forum](#).

Creating the First Web Application

Following are the steps needed for adding Crystal Reports 2008 to the Delphi 2007 development environment and for creating the first simple web forms project which displays a report in the Crystal Reports DHTML viewer. At the end of this section, a number of further resources are also provided, including Crystal Reports runtime distribution for .NET.

Directing Delphi 2007 to Use Crystal Reports 2008 Assemblies for Visual Studio .NET

The approach described here will update the Borland Delphi 2007 IDE so that all future attempts to load Crystal Reports assemblies will result in the correct references being used. Specifically, the assemblies will be of version 12.0.2000.0

Ensure that the Delphi 2007 IDE closed. Browse to <installation directory>\Program Files\CodeGear\RAD Studio\5.0\bin and open **bds.exe.config** in a text editor. By default, a majority of this file is enclosed between these two tags:

```
<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
...
</assemblyBinding>
```

Add the following entries prior to the </assemblyBinding> tag, but not enclosed within another assembly's <dependentAssembly> tag.

```
<dependentAssembly>
  <assemblyIdentity name="CrystalDecisions.Web"
    publicKeyToken="692fbea5521e1304" culture="neutral" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="CrystalDecisions.Shared"
    publicKeyToken="692fbea5521e1304" culture="neutral" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="CrystalDecisions.ReportSource"
    publicKeyToken="692fbea5521e1304" culture="neutral" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="CrystalDecisions.CrystalReports.Engine"
    publicKeyToken="692fbea5521e1304" culture="neutral" />
</dependentAssembly>
```

Save and close bds.exe.config.

CAUTION Incorrect changes to the **bds.exe.config** file can cause the Borland Developer Studio to not start up correctly and cause access violations. It is strongly recommended to keep a backup copy of the original file before making any changes.

Setting up Directories/Conditionals

As Crystal Reports uses files which are located in several directories, it is necessary to specify which directories Delphi 2007 IDE should notice. To setup Directories/Conditionals go to the Project menu in the Delphi 2007 IDE. Select **Options > Directories/Conditionals**. In the search path, specify the following:

- C:\Program Files\Business Objects\BusinessObjects Enterprise 12.0\win32_x86;C:\Program Files\Business Objects\Common\4.0\managed\dotnet2;
- C:\Program Files\Business Objects\Common\4.0\managed;
- C:\Program Files\Business Objects\BusinessObjects Enterprise 12.0\win32_x86\managed

Referencing the Crystal Reports 2008 DHTML Viewer for .NET

This section provides a process to create a simple Delphi 2007 Web Forms application to preview a Crystal Reports 2008 report.

1. Create a new Web Forms Application by following these menu selections: File | New ASP .Net Web Application – Delphi for .NET.
2. To get the correct version of the viewer to appear on the Tool Palette we'll need to manually select an assembly by browsing to the correct Crystal Reports directory.
3. On the "Installed .NET Components" form, change the text next to Category from "General" to "Crystal Reports". Click the **select an assembly...** button and browse to the C:\Program Files\Business Objects\Common\4.0\managed\dotnet2 folder. Select the **CrystalDecisions.Web.dll**. Close the Installed .NET Components dialog.
4. Under the Component menu select **Installed .NET Components...** Highlight the .NET Components tab and scroll down to the Crystal Reports Viewer assemblies. Place a check mark at the **CrystalDecisions.Web** namespace.
5. Go to the design view of the web form. The Tool Palette has "Crystal Reports" as one of the headings. Double-click **CrystalReportViewer** to place the viewer on your web form.

When the Crystal reports DHTML viewer is added to the form, it will automatically add the following references to the project:

- CrystalDecisions.CrystalReports.Engine.dll
- CrystalDecisions.ReportSource.dll
- CrystalDecisions.Shared.dll
- CrystalDecisions.Web.dll.

All of these assemblies should be version 12.0.2000.0 and be loaded from C:\Program Files\Business Objects\Common\4.0\managed\dotnet2 folder. Confirm this is the case by highlighting one of the assemblies under the. In the Object Inspector the version number and full assembly path are shown.

Displaying the first report in the Crystal Reports DHTML Viewer

Add `CrystalDecisions.CrystalReports.Engine`, `CrystalDecisions.Shared`, `CrystalDecisions.Web`, and `CrystalDecisions.ReportSource` to the “uses” clause of the form.

Use the following code with a saved data report to display a report using the Crystal Reports viewer SDK:

```
procedure TDefault.Page_Load(sender: System.Object; e: System.EventArgs);
begin
  // TODO: Put user code to initialize the page here
  // This will display the report using the Crystal Reports viewer API
  CrystalReportViewer1.ReportSource := ('c:\crystal\deleteme.rpt');
end;
```

Run the project. Once the project is successfully compiled, a report should come up in the Crystal Reports DHTML viewer.

To use the Crystal Reports engine SDK, use the following code:

```
procedure TDefault.Page_Load(sender: System.Object; e: System.EventArgs);
var crReportDocument : CrystalDecisions.CrystalReports.Engine.ReportDocument;
begin
  // TODO: Put user code to initialize the page here
  // This will display the report using the Crystal Reports engine API

  crReportDocument:= CrystalDecisions.CrystalReports.Engine.ReportDocument.Create();
  crReportDocument.Load('c:\crystal\deleteme.rpt');
  CrystalReportViewer1.ReportSource := crReportDocument;
end;
```

For more help with coding, download the [Crystal Reports 2008 .Net SDK Developer Guide](#) and [Crystal Reports 2008 .Net API Reference](#).

There are also three windows sample projects written in Delphi 2005 available for download. Two of these projects use strongly typed reports. It is not possible to create strongly typed reports in Delphi 2007 and Delphi 2005 projects using strongly typed reports must be modified and use the `.Load` method of the Crystal Reports engine SDK or the `.ReportSource` method of the viewer SDK.

Converting Applications Using Strongly Typed Reports

The following Delphi 2005 sample applications are available for download:

[Passing parameters using Delphi for .NET](#)

[Connecting to a database using Delphi in .NET](#)

[Exporting to PDF using Delphi for .NET](#)

The instructions below detail how to convert applications using strongly typed reports.

1. Open the project in Delphi 2007.
2. Using the Object Inspector, check that all the Crystal Reports Assemblies have been updated to version 12.0.2000.0.
3. Remove the assembly CrystalKeyCodeLib.dll from the project.
4. Remove from the project the Report1Unit.pas file.
5. Delete Report1Unit from the uses clause.
6. Go to the Project menu in the Delphi 2007 IDE. Select Options and then Directories/Conditionals. In the search path, remove the path;

`c:\windows\microsoft.net\framework\v1.1.4322;`

and ensure that these paths have been added:

`C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727;`

`C:\Program Files\Business Objects\BusinessObjects Enterprise 12.0\win32_x86;`

`C:\Program Files\Business Objects\Common\4.0\managed\dotnet2;`

`C:\Program Files\Business Objects\Common\4.0\managed;`

`C:\Program Files\Business Objects\BusinessObjects Enterprise 12.0\win32_x86\managed`

7. Find the “strict private” clause and delete or comment out the following line of code:

```
CrystalReport1 : ReportUnit1.Report1;
```

8. Add the following line of code to the “strict private” clause:

```
CrystalReport1 : CrystalDecisions.CrystalReports.Engine.ReportDocument;
```

Find the clause:

```
Procedure TForm.InitializeComponent;
```

And delete or comment out the following line:

```
Self.CrystalReport1 := Report1Unit.Report1.Create;
```

9. Add the following of code to Procedure TForm.InitializeComponent;

```
Self.CrystalReport1 :=
```

```
CrystalDecisions.CrystalReports.Engine.ReportDocument.Create;
```

```
Self.CrystalReport1.Load :=('c:\test\mytestreport.rpt')
```

Distributing the Crystal Reports Runtime for the .NET Framework

Runtime distributions of the Crystal Reports runtime can only be achieved via the use of Crystal Reports merge modules or .msi files. As the Crystal Reports for VS .NET components rely on COM Interop for the backend report processing engine, copying runtime files or XCopy deployments do not work. For more information regarding deployment of Crystal Reports 2008 runtime see the [Crystal Reports for Visual Studio .NET Runtime Distribution - Versions 9.1 to 12.0](#) wiki.

Related Content

[Crystal Reports For Visual Studio 2005 Walkthroughs](#)

[Getting Started with the Crystal Report Designer Component in Borland Delphi 5](#)

[Migrating Applications from the RDC to the .NET assemblies](#)

[Getting Started with Crystal Reports .NET SDK](#)

[Notes database](#)

[Articles](#)

[Downloads](#)

For more information, visit the [Business Objects homepage](#).

Copyright

© Copyright 2009 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.