

Crystal Reports for Visual Studio .NET

Using Server File Report Web Services

Overview

This document discusses using Server File Report web services (SFR web services) in Crystal Reports for Visual Studio .NET. This document reviews the components that makes up the SFR web services and examines how to use these components.

Contents

| | |
|---|-----------|
| INTRODUCTION | 2 |
| WHY USE SERVER FILE REPORT WEB SERVICES? | 2 |
| BASIC ARCHITECTURE | 2 |
| SFR COMPONENTS | 4 |
| <i>The Server File Report Web Services</i> | <i>5</i> |
| ServerFileReportManager.ASMX | 5 |
| ServerFileReportService.ASMX | 6 |
| <i>Classes Used to Access SFR Web Services</i> | <i>7</i> |
| ServerFileReportManagerProxy | 7 |
| ReportManagerRequest | 7 |
| ReportManagerResponse | 8 |
| ServerFileReport | 8 |
| LIMITATIONS | 9 |
| Clients to SFR Web Services | 9 |
| Runtime Customizability of Server File Reports | 9 |
| Support for ADO.NET Datasets | 10 |
| OTHER RESOURCES | 10 |
| CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT | 10 |

Introduction

One of the main features of Microsoft's .NET initiative are web services. Since Crystal Reports for Visual Studio .NET (CR for VS .NET) is tightly integrated into Visual Studio .NET and the .NET Framework, it only made sense to provide web services for reports.

CR for VS .NET provides different methods of using web services for reports, but this document focuses on using Server File Report web services (SFR web services).

Why Use Server File Report Web Services?

Today's applications require some kind of reporting functionality built into the system. Some systems only provide a handful of reports while other systems need to provide many reports. Smaller and older applications can distribute client applications as well as reports to each desktop. However, larger and distributed systems require reports to be centralized which adds complexity to the overall system. With .NET, it becomes easier to utilize reports in a distributed system.

SFR web services allow you to store reports for remote access using .NET web service features (such as SOAP or XML). Using SFR web services allows reports to be hosted on a centralized web server so they can be accessed by any type of Windows or web client.

The client in this case would use one of the Crystal Report Viewer controls that CR for VS .NET installs. The Windows Form Viewer and Web Forms Viewer controls should be on the client computer that consumes the SFR web services.

In Visual Studio .NET, you can view which folders and reports the SFR web services will use by opening the **Server Explorer**. Expand the node that shows the local server name, and you will find a **Crystal Services** node. Expand the **Crystal Services** node, and you will see a node called **Server Files**. Under the Server Files node is a listing of folders and reports.

NOTE

If there is no listing under the Server Files node, please go to <http://support.crystaldecisions.com/kbase> and search for knowledge base article c2010774.

Basic Architecture

Before getting into the actual components that make up the SFR web services, it is important to see the bigger picture. The bigger picture is how SFR web services fit into a distributed .NET system.

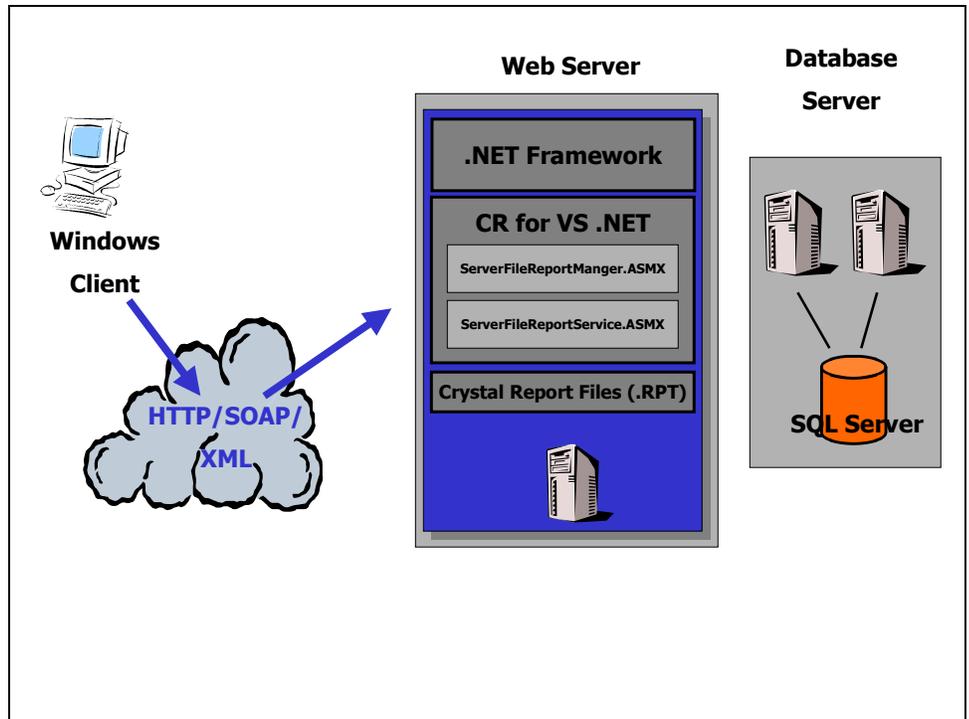


Diagram 1- Windows application

The main component of the overall architecture is the web server. The web server hosts the SFR web services and the physical report files. Since the web server hosts the SFR web services, the .NET Framework as well as the Crystal Reports components, must be installed on the web server.

The data source for the reports could be on the same computer as the web server or hosted on another server on the same network. In any case, the reports can be based on a secured data source, which may or may not be on the same server.

The clients for viewing these Server File Reports can be either Windows or web based applications. Diagram 1 illustrated a scenario for a Windows application.

You could design a web application (see diagram 2) similar to a Windows application where you only have one web server that handles requests from the client, as well as host the SFR web services. Or you can distribute and isolate the SFR web services from the web browser that handles client request.

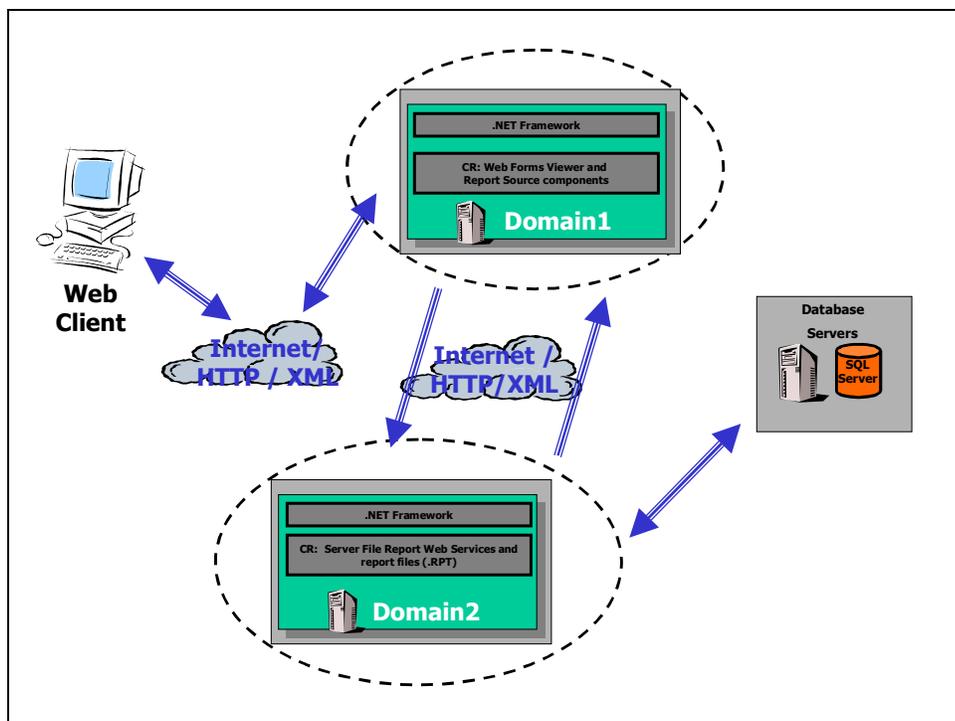


Diagram 2 – Web application

The web client would request a web page (.ASPX) to be loaded from the first web server (Domain 1). This web server is responsible to handle client requests for web pages. This web server would only require Crystal Reports components such as the Web Forms Viewer server side control, but the SFR web services are not required.

When the Web Forms Viewer is set to a Server File Report, it will make calls to the second web server (Domain 2) to requests specific reports. All this communication is done using SOAP and XML built into the .NET Framework. The required communication between the web servers and the SFR web services are done is set up for you.

SFR Components

The SFR web services are not made up of many components. The two components that actually make up the SFR web services are:

- ServerFileReportManager.ASMX
- ServerFileReportService.ASMX

There are also classes in the CrystalReports.Shared namespace that are used to access and use the web services. These include:

- ServerFileReportManagerProxy
- ReportManagerRequest

- ReportManagerResponse
- ServerFileReport

The Server File Report Web Services

Use the two SFR web services to get a list of objects (from the web server hosting these web services) and to view specific reports in the CR for VS .NET viewers (Windows Form Viewer or Web Forms Viewer).

When these web services are invoked, they look for a parent (or root) folder to provide reports. A configuration file called **crystalserverfile.config** provides this root folder. By default the **crystalserverfile.config** file is not created until the first time the web services are accessed. This file is found in the folder **<c:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Config>**.

Since these are web services, there are security settings that you must first configure before this **crystalserverfile.config** is created. You must give the ASPNET account full permissions to the folder, **<c:\Program Files\Microsoft Visual Studio .NET\Crystal Reports>**.

When the **crystalserverfile.config** file is created, it has the following XML content:

```
<?xml version="1.0" encoding="utf-8"?>
<ServerFileConfig
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RootDirectory>C:\Program Files\Microsoft Visual Studio
.NET\Crystal Reports\Samples\Reports</RootDirectory>
</ServerFileConfig>
```

NOTE

You will need to modify the path specified by the **<RootDirectory>** tag to the folder that contains the reports on your web server.

ServerFileReportManager.ASMX

This web service is used to retrieve a list of objects. This list can be a list of folders and/or reports from the folder specified by the **crystalserverfile.config** file.

This web service exposes two web methods:

- ListChildObjects
- TestListChildObjects

The **ListChildObjects** is the only web method used for this web service. This web method takes a [ReportManagerRequest](#) object and returns a [ReportManagerResponse](#) object that has a list of folders and/or reports from the path given in the **crystalserverfile.config** file.

The list is in a form of URI's (ReportURI) that defines a path (relative to the path defined in the `crystalserverfile.config` file) and whether the object is a FOLDER or a REPORT.

The **TestListChildObjects** web method tests the web service to make sure it works correctly. If you view the web service in a web browser (such as <http://localhost/crystalreportwebformviewer/serverfilereportmanager.asmx>) you will get the list of web methods.

Click on the `TestListChildObjects` link to take you to another page where you can enter a path to a folder, and click on the `Invoke` button. This will load a new web browser and display the XML that is returned. You will see the URI (Unique Resource Identifier) information returned by the web method. The XML information that is returned by the `TestListChildObjects` web method is what the `ListChildObjects` web method should also return.

ServerFileReportService.ASMX

The CR for VS .NET viewers (Windows and Web Forms Viewer) uses this web service to access and view Server File Reports. This web service exposes the following web methods:

- DrillGraph
- Refresh
- FindText
- GetPage
- GetTotaller
- TestReport
- GetGroupLevelData
- FindGroup
- GetLastPageNumber
- GetReportInfo
- Export
- TestGetPage
- TestGetTotaller

It is important to know that the Windows Form and Web Forms Viewers (the client for these web services) will make these web method calls to load a report. You cannot create your own client and invoke these web methods to build a custom client.

Classes Used to Access SFR Web Services

Although there are multiple classes that are used with the SFR web services, there are four main classes that can be used to access and invoke the SFR web services from an application.

These four classes are exposed by the **CrystalDecisions.Shared** namespace. The client application that is using these classes must add a reference to the CrystalDecisions.Shared.DLL assembly. The four main classes include:

- ServerFileReportManagerProxy
- ReportManagerRequest
- ReportManagerResponse
- ServerFileReport

ServerFileReportManagerProxy

This class is used to represent the ServerFileReportManager.ASMX web service. When an instance of this class is created, set the **Url** property to a URL location of the web service. For example:

```
Dim crServerFileReportManagerProxy As
ServerFileReportManagerProxy

crServerFileReportManagerProxy = New
ServerFileReportManagerProxy ()

With crServerFileReportManagerProxy
    .Url =
"http://localhost:80/crystalreportwebformviewer/serverfiler
eportmanager.aspx"
End With
```

Besides the Url property, this class exposes the ListChildObjects method. This method accepts a [ReportManagerRequest](#) object as an argument and returns a [ReportManagerResponse](#) object. The ListChildObjects method is essentially the same as the web method of the ServerFileReportManager web service.

ReportManagerRequest

This class is used to define the information that will be used by the ListChildObjects web method of the [ServerFileReportManager](#) web service.

This class exposes two members:

- ExtraData
- ParentUri

Both of these members must be set before invoking the `ListChildObjects` method. The values for these members are extracted from the [ServerFileReport](#) class. The following sample code shows how to use this class:

```
crReportManagerRequest = New ReportManagerRequest()  
With crReportManagerRequest  
    .ExtraData = crServerFileReport.GetExtraData()  
    .ParentUri = crServerFileReport.ToUri()  
End With
```

ReportManagerResponse

This class is used as the return value of the `ListChildObjects` method of the [ServerFileReportManagerProxy](#) class. It is used to get a collection of URI (ReportUris) that are returned by the `ListChildObjects` method. It also returns any error information that may have occurred.

The ReportUris returned by this class contain the FOLDER and REPORT returned by the `ListChildObjects` method. Once this collection is retrieved, the collection can be indexed to get specific URI information. For example:

```
Dim crReportManagerResponse As ReportManagerResponse  
crReportManagerResponse =  
crServerFileReportManagerProxy.ListChildObjects(crReportMan  
agerRequest)
```

ServerFileReport

As one of the main classes, it is used extensively with SFR web services. At the most basic level, all you need to specify are values for the **ReportPath** and **WebServiceUrl** properties to access a report. For example:

```
crServerFileReport = New ServerFileReport()  
With crServerFileReport  
    .ReportPath = node.Tag  
    .WebServiceUrl =  
"http://localhost:80/crystalreportwebformviewer/serverfiler  
eportservice.asmx"  
End With
```

This class also exposes two other methods that are used to extract information.

- `GetExtraData`
- `ToUri`

The `GetExtraData()` and `ToUri()` methods are used to extract information for the [ReportManagerRequest](#) class.

When the `ListChildObjects` method returns a [ReportManagerResponse](#) object, the `ServerFileReport` class is used to extract the information from the `ReportManagerResponse` object. For example:

```
For i = 0 To crReportManagerResponse.ReportUri.Count - 1
    crServerFileReport =
    ServerFileReport.FromUri (crReportManagerResponse.ReportUri
    .Item(i))

crServerFileReport.SetExtraData (crReportManagerResponse.ExtraData)
Next i
```

Once a ServerFileReport object is defined for a specific report, it can be set to the **ReportSource** property of either the Windows or Web Form Viewer. The viewer will use the information in the ServerFileReport object to access and invoke the ServerFileReportService web service to view that particular report. For example:

```
crServerFileReport = New ServerFileReport()
With crServerFileReport
    .ReportPath = ""
    .WebServiceUrl =
    "http://localhost:80/crystalreportwebformviewer/serverfiler
    eportservice.aspx"
End With
CrystalReportViewer1.ReportSource = crServerFileReport
```

Limitations

The following list limitations with the SFR web services:

Clients to SFR Web Services

The only clients that can exist for SFR web services are the two runtime report viewers that are provided by Crystal Reports for VS .NET.

- Windows Form Viewer
- Web Forms Viewer

The Windows Form Viewer is for use in rich Windows client applications. It is a Windows Form control and offer rich report viewing.

The Web Forms Viewer is an ASP.NET server side control. It is for use in Web applications to offer zero-client report viewing capabilities.

Developing custom clients for SFR web services is not exposed nor supported.

Runtime Customizability of Server File Reports

When using SFR web services to access and view reports in clients, there are report functionality limitations.

The CrystalDecisions.CrystalReports.Shared namespace exposes a ReportDocument class that provides rich runtime report customizability. With the ReportDocument, you have access to runtime features such as (but not limited to):

- Modifying formula fields
- Passing ADO.NET Datasets
- Modifying report objects
- Setting text for Text objects

Unfortunately, the SFR web services do not allow you to access the abovementioned features.

The viewer that is bound to the Server File Report object exposes an object model that can be used to access report features but it is not as rich as the ReportDocument class. Some of the runtime features that are available from the viewer's object model include:

- Database logon
- Parameter fields
- Record selection formula

Support for ADO.NET Datasets

SFR web services have limited support for ADO.NET datasets. A report based on an ADO.NET dataset cannot be used as a Server File Report object since you cannot pass the dataset object to the Server File Report object. The viewer's object model does not expose a member to accept a dataset object.

Other Resources

You can download sample applications demonstrating using the SFR web services by going to <http://support.crystaldecisions.com/downloads> and searching for the following files:

- Csharp_win_serverfilereports.exe
- Vbnet_win_serverfilereports.exe

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>