ComponentOne

SSRSDocumentSource for WinForms

ComponentOne, a division of GrapeCity
201 South Highland Avenue, Third Floor
Pittsburgh, PA 15206 USA

Website: http://www.componentone.com
Sales: sales@componentone.com
Telephone: 1.800.858.2739 or 1.412.681.4343 (Pittsburgh, PA USA Office)

Trademarks

Warranty

ComponentOne warrants that the media on which the software is delivered is free from defects in material and workmanship, assuming normal use, for a period of 90 days from the date of purchase. If a defect occurs during this time, you may return the defective media to ComponentOne, along with a dated proof of purchase, and ComponentOne will replace it at no charge. After 90 days, you can obtain a replacement for the defective media by sending it and a check for $2 5 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original media on which the software is delivered is set forth here, ComponentOne makes no other warranties, express or implied. Every attempt has been made to ensure that the information contained in this manual is correct as of the time it was written. ComponentOne is not responsible for any errors or omissions. ComponentOne's liability is limited to the amount you paid for the product. ComponentOne is not liable for any special, consequential, or other damages for any reason.

Copying and Distribution

Table of Contents

SSRSDocumentSource for WinForms

SSRSDocumentSource for WinForms, allows you to load an SSRS report into a C1PrintPreviewControl for viewing and exporting to a variety of formats including Adobe PDF, Microsoft Word and Excel.

Key Features

- **Full-Featured Preview**

  C1PrintPreviewControl control provides a ready-to-use full-featured UI with thumbnail and outline views, text search, and predefined toolbars right out of the box.

- **Connect to and View Reports**

  Connect to and view reports from SQL Server Reporting Services (SSRS).

- **Zoom and Navigation**

  Navigate and zoom pages in the report by mouse.

- **Export and Save Reports**

  Export and save reports locally as Adobe PDF (.pdf), Microsoft Word (.doc), Microsoft Excel (.xls), Web archive (.mhtml), CSV (.csv), Tiff Image (.tiff), Bmp image (.bmp), Enhanced Metafile (.emf), and Gif image (.gif).

- **Printing Reports**

  Print reports with support for custom page settings and paginated print preview.

- **Thumbnail Support**

  View all report pages as easy-to-access page thumbnails.

Help with WinForms Edition

Getting Started

For information on installing **ComponentOne Studio WinForms Edition**, licensing, technical support, namespaces and creating a project with the control, please visit Getting Started with WinForms Edition.

Run Time File

SSRSDocumentSource Quick Start

C1SsrsDocumentSource allows you to specify the location or name of the SSRS report to be viewed. It will then enable the C1PrintPreviewControl to display an SSRS report.

It is not a visual component so it can be added onto a form but it appears in the component tray.

> **Note**: You must have access to the SSRS report server and have at least one report on that server before you start this topic.

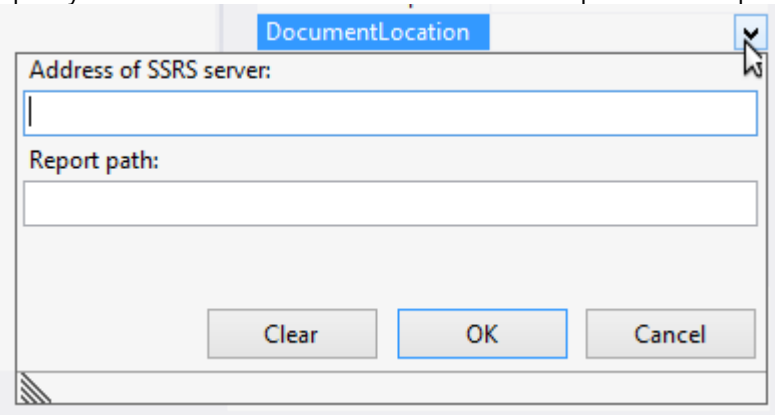
To preview or export an SSRS report, using the C1PrintPreviewControl, create a new WinForms application, using .NET framework 4.0 or above and complete the following steps:

In the Designer

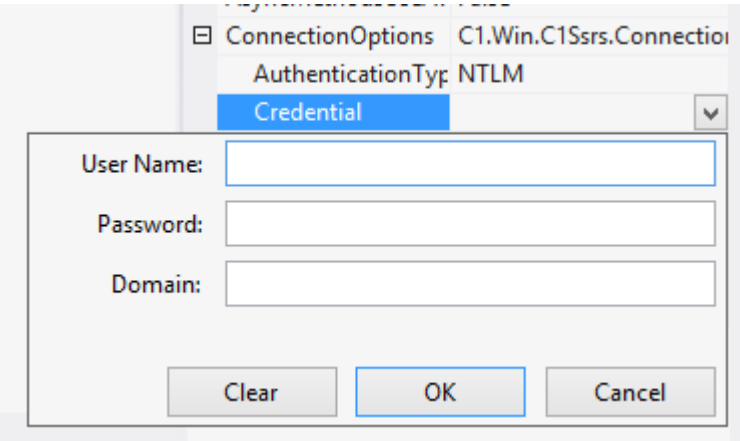
1. From the toolbox, add the C1SSRSDocumentSource component onto the form. It appears in the component tray.

   > If you cannot find the component in the toolbox, right click and select **choose items**. The **Choose Toolbox Items** dialog appears. Locate the component in the **C1.Win.C1Document.dll** and add it to the toolbox.

2. Add the **C1PrintPreviewControl** onto the form, and set its height and width as desired.
3. Right click the C1PrintPreviewControl and select **Properties** to open the properties window.
4. Set the SsrsPaginated property to true to view a paginated report. Or set it to false to view the report in non-paginated mode.
5. Click the C1SSRSDocumentSource component, then click the smart tag to open the **C1SSRSDocumentSource Tasks Menu**.
6. Check the checkbox next to the **C1PrintPreviewControl**. This assigns the C1SSRSDocumentSource component to the C1PrintPreviewControl's **Document** property.
7. Right click the C1SSRSDocumentSource and select **Properties** to open the Properties window.
8. Click the drop-down arrow next to the DocumentLocation property.
9. Specify the address of the SSRS server and the full path to the report in the following form and click **OK**.



10. Expand the **ConnectionOptions** property group.
11. Click the drop-down arrow next to the Credential property.
12. Specify the User Name, Password and Domain in the following form and click **OK**.



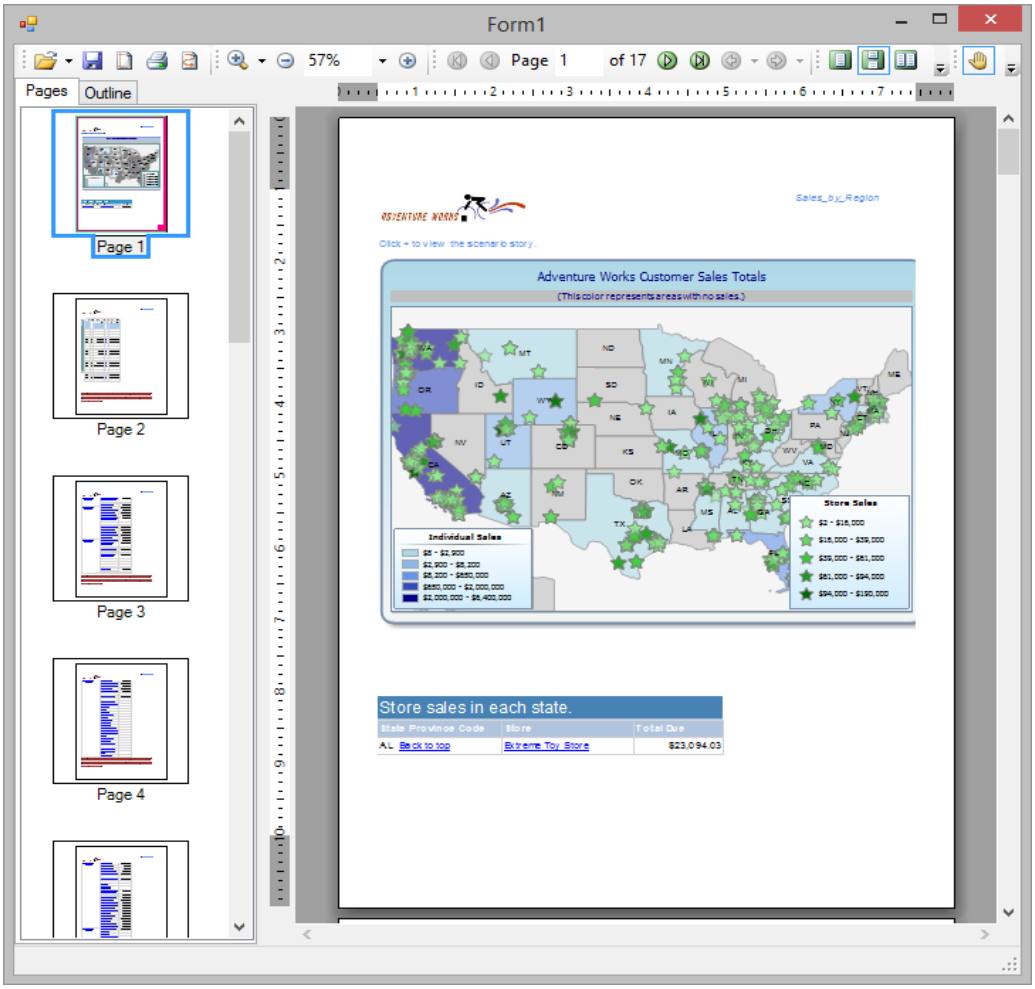
Note: Ensure that the following DLLs are added to the project references.

- C1.Win.C1Document.dll
- C1.Win.dll
- C1.Win.C1ImportServices.dll
- C1.Win.C1Report.dll
- C1.C1Report.dll
- C1.C1Zip.dll

You can also set the document location and network credentials in Code. See Set DocumentLocation in Code and Set Credentials in Code for more information.

What You've Accomplished

You may be asked to set some parameters, based on which you see the specified SSRS report in the preview control.



You can now export it to a number of supported external formats.

Set Credentials in Code

You can specify the username and password to access your SSRS report by setting the Credential property in C1SSRSDocumentSource's SecurityError event. This event will be fired and will allow you to provide the required credentials and retry the operation, when access to the SSRS server is denied due to lack of credentials.

In the Designer

1. Open the C1SSRSDocumentSource's **Events** window.
2. Double click the SecurityError event. This creates an empty handler for that event in your code.

In Code

1. In code view, modify the `c1SSRSDocumentSource1_SecurityError` event handler as shown below:

C#

```
  private void c1SSRSDocumentSource1_SecurityError(object sender,
C1.Win.C1Document.SecurityErrorEventArgs e)
  {
    var ds = (C1.Win.C1Document.C1SSRSDocumentSource)sender;
    ds.Credential = new System.Net.NetworkCredential("myUserId", "myPassword");
    e.Retry = true;
  }
```

Visual Basic

```
    Private Sub C1SSRSDocumentSource1_SecurityError(sender As Object, e As
C1.Win.C1Document.SecurityErrorEventArgs) Handles
C1SSRSDocumentSource1.SecurityError
        Dim ds = DirectCast(sender, C1.Win.C1Document.C1SSRSDocumentSource)
        ds.Credential = New System.Net.NetworkCredential("myUserId",
"myPassword")
        e.Retry = True
    End Sub
```

In the code above, replace "myUserId" and "myPassword" with valid credentials for your SSRS report server. When your form loads, the C1SSRSDocumentSource component will try to access the specified report.

> **Note**: Using improper credentials will throw a security error. So, you must specify a handler that will allow you to provide valid credentials and retry the operation.

2. Run the application.

Set DocumentLocation in Code

You can set the location of the SSRS document in code using in the **Form_Load** event. See SSRSDocumentSource Quick Start to know how to set document location using the designer.

In Code

C#

```
c1SSRSDocumentSource1.DocumentLocation = new
C1.Win.C1Document.SSRSReportLocation("http://ssrs.YourReportLocation.com/ReportServer",
"/Adventure Works/Sales");
```

Visual Basic

```
c1SSRSDocumentSource1.DocumentLocation = New
C1.Win.C1Document.SSRSReportLocation("http://ssrs.YourReportLocation.com/ReportServer",
"/Adventure Works/Sales")
```

Display Document State and Page Count

This topic demonstrates how to use the PageCount and State properties of the C1SsrsDocumentSource component to display the state of the document source and number of pages already generated. Create an application as shown in the Quick Start and complete the following steps.

In the Designer

1. Place two labels onto the form.
2. Set the **Name** of the first label as **tbReport**.
3. Set the **Name** of the second label as **tbState**.
4. Set the **Text** property for the first label as **Report: None**. This label will display the location of the report.
5. Set the **Text** property for the second label as **State: Unknown**. This label will display the status of the report and the number of pages generated.

In Code

Add the following code to the **Form_Load** event, to get the report location, state and page count into the labels.

C#

```
SSRSReportLocation rl = c1SSRSDocumentSource1.DocumentLocation as SSRSReportLocation;
if (rl != null)
        tbReport.Text = C1.Win.C1Ssrs.ReportSession.Combine(rl.ReportServer,
rl.ReportPath);
else
    tbReport.Text = "None";
switch (c1SSRSDocumentSource1.BusyState)
{
    case C1DocumentSourceBusyState.Generating:
    case C1DocumentSourceBusyState.Ready:
    case C1DocumentSourceBusyState.Exporting:
    case C1DocumentSourceBusyState.Printing:
        tbState.Text = string.Format("{0}, Pages: {1}",
c1SSRSDocumentSource1.BusyState, c1SSRSDocumentSource1.PageCount);
        break;
    default:
        tbState.Text = c1SSRSDocumentSource1.BusyState.ToString();
        break;
}
```

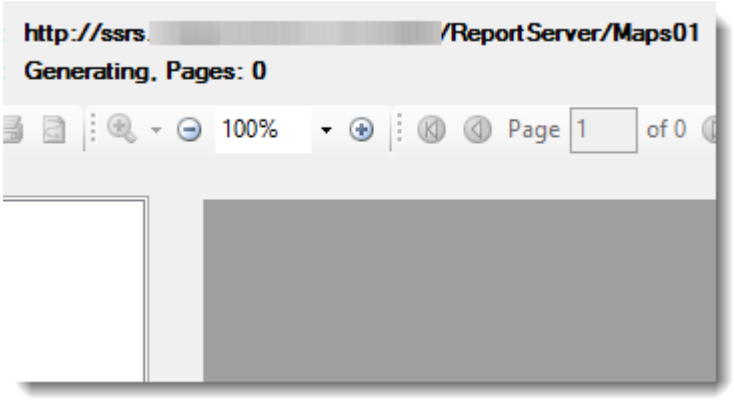
Visual Basic

```
Dim rl As C1.Win.C1Document.SSRSReportLocation =
TryCast(C1SSRSDocumentSource1.DocumentLocation, SSRSReportLocation)
If rl IsNot Nothing Then
        tbReport.Text = C1.Win.C1Ssrs.ReportSession.Combine(rl.ReportServer,
rl.ReportPath)
Else
    tbReport.Text = "None"
End If
Select Case C1SSRSDocumentSource1.BusyState
    Case C1DocumentSourceBusyState.Exporting, C1DocumentSourceBusyState.Generating,
```

```
C1DocumentSourceBusyState.Ready, C1DocumentSourceBusyState.Printing
                tbState.Text = String.Format("{0}, Pages: {1}",
C1SSRSDocumentSource1.BusyState, C1SSRSDocumentSource1.PageCount)
            Exit Select
    Case Else
            tbState.Text = C1SSRSDocumentSource1.BusyState.ToString()
            Exit Select
End Select
```

What You've Accomplished

When you run your project, notice that the labels that you placed on form display the report location, status and number of pages generated.



Important Properties and Methods

The following pages contain information about the important properties and methods of the C1SSRSDocumentSource control.

Important Methods

Most of the C1SSRSDocumentSource methods that are used to generate and execute queries in reports, have two versions: synchronous and asynchronous.

For example:

- C1SSRSDocumentSource.Generate - synchronous, blocking version of the method.
- C1SSRSDocumentSource.GenerateAsync - asynchronous, non-blocking version of the same method.

Some important C1SSRSDocumentSource methods are:

Property	Description
Generate	Generate the report. C1SSRSDocumentSource.Generate() can be used immediately after defining all necessary properties like DocumentLocation, ConnectionOptions etc. Note that all GenerateXXX methods are accessible only in C1SSRSDocumentSource, in C1DocumentSource they are internal.
GenerateAsync	
ValidateParameters	Validate the current parameter values, refresh their valid values' lists if the values are valid. Note that parameter values are now applied automatically when a report generation starts.
ValidateParametersAsync	
Export	Export the report into one of the supported external formats such as PDF.
ExportAsync	
Clear	Clears the generated report.

Important Properties

Some important C1SSRSDocumentSource properties are:

Property	Description
DocumentLocation	Gets or sets the location of the SSRS report, which includes the server address (e.g. http://ssrs.YourReportLocation.com/ReportServer) and the report path (e.g. /Adventure Works/Sales).
BusyState	Gets the current busy state of the current C1.Win.C1Document.C1DocumentSource. See the C1DocumentSourceBusyState enum for the list of busy states.
IsBusy	Gets the value indicating whether the current C1.Win.C1Document.C1DocumentSource is busy.
IsDisposed	Gets a value indicating whether this C1.Win.C1Document.C1DocumentSource is disposed and can not be longer used.

PageCount	Gets the number of already generated pages.
HasParameters	Gets a value indicating whether any parameters have been specified.
Parameters	Gets a collection of parameters used to generate content. The C1SSRSDocumentSource.ValidateParameters() method has to be used to validate the current parameter values, and refresh their valid values' lists if the values are valid. The values of parameters are applied automatically before generating.
PageLayout	Gets or sets the PageLayout object specifying the page layout to use when generating a paginated document. For SSRS reports, the following PageLayout properties are ignored: PageHeader, PageFooter, Watermark.
ConnectionOptions	Gets the ConnectionOptions object containing options used when connecting to SSRS.
Credential	Gets or sets NetworkCredential object defining credentials used in the SSRS connection. This is a shortcut to ConnectionOptions.Credential.
Dirty	Gets a value indicating whether the current content is dirty and does not correspond to the current values of DocumentLocation, Paginated etc.
AsyncMethodsUseAwaitPattern	Gets or sets a value indicating whether calls to the async methods of the current document source expect the async/await pattern to be used by the calling code. If this property is true, then the completed events (such as OpenCompleted, GenerateCompleted and others) do not fire.
IsCurrentActionCancellable	Gets a value indicating whether the current action can be cancelled.