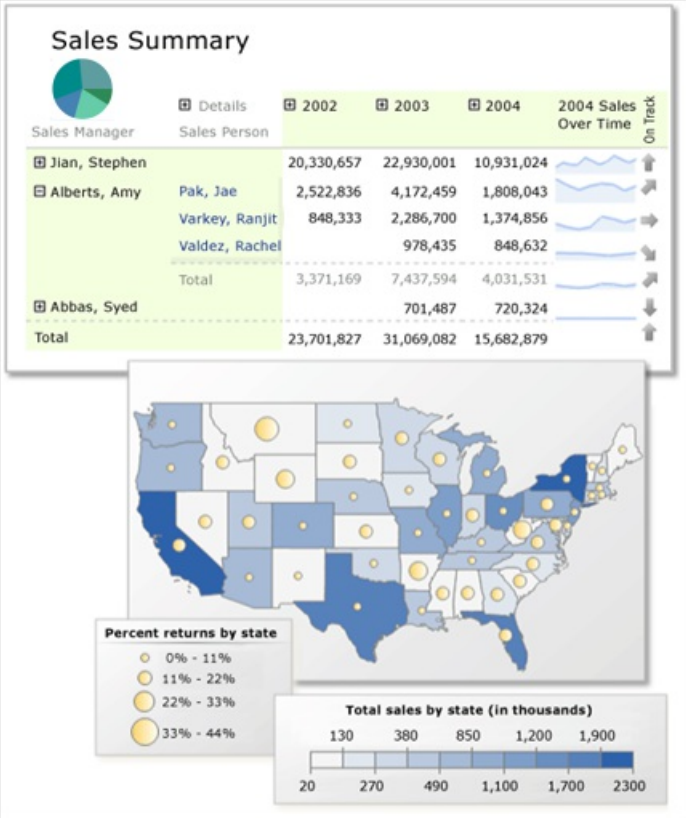
Table of Contents

Report Builder in SQL Server 2016

3/31/2017 • 4 min to read • Edit Online

Report Builder is a tool for authoring paginated reports, for business users who prefer to work in a stand-alone environment instead of using Report Designer in Visual Studio. When you design a paginated report, you're creating a report definition that specifies where to get the data, which data to get, and how to display the data. When you run the report, the report processor takes the report definition you have specified, retrieves the data, and combines it with the report layout to generate the report. You can preview your report in Report Builder and publish your report to a Reporting Services report server in native mode or in SharePoint integrated mode, where others can run it.



This paginated report features a matrix with row and column groups, sparklines, indicators, and a summary pie chart in the corner cell, accompanied by a map with two sets of geographic data represented by color and by circle size.

Jump-Start Report Creation

- **Start with a shared dataset**. Shared datasets are queries based on a shared data source and saved to a Reporting Services report server in native mode or in SharePoint integrated mode.
- **Start with the Table, Matrix, or Chart wizard**. Choose a data source connection, drag and drop fields to create a dataset query, select a layout and style, and customize your report.
- **Start with the Map wizard** to create reports that display aggregated data against a geographic or geometric background. Map data can be spatial data from a Transact-SQL query or an Environmental Systems Research Institute, Inc. (ESRI) shapefile. You can also add a Microsoft Bing map tile background.
- **Start your report with report parts**. Report parts are report items that have been published separately to a Reporting Services report server in native mode or in SharePoint integrated mode. Report parts can be

reused in other reports. Report items such as tables, matrices, charts, and images can be published as report parts.

Design Your Report

- **Create paginated reports with table, matrix, chart, and free-form report layouts.** Create table reports for column-based data, matrix reports (like cross-tab or PivotTable reports) for summarized data, chart reports for graphical data, and free-form reports for anything else. Reports can embed other reports and charts, together with lists, graphics, and controls for dynamic Web-based applications.
- **Report from a variety of data sources.** Build reports using data from any data source type that has a Microsoft .NET Framework-managed data provider, OLE DB provider, or ODBC data source. You can create reports that use relational and multidimensional data from SQL Server and Analysis Services, Oracle, Hyperion, and other databases. You can use an XML data processing extension to retrieve data from any XML data source. You can use table-valued functions to design custom data sources.
- **Modify existing reports.** By using Report Builder, you can customize and update reports that were created in SQL Server Data Tools (SSDT)Report Designer.
- **Modify your data** by filtering, grouping, and sorting data, or by adding formulas or expressions.
- **Add charts, gauges, sparklines, and indicators** to summarize data in a visual format, and present large volumes of aggregated information at a glance.
- **Add interactive features** such as document maps, show/hide buttons, and drillthrough links to subreports and drillthrough reports. Use parameters and filters to filter data for customized views.
- **Embed or reference images** and other resources, including external content.

Manage Your Report

- **Save the definition of the report** to your computer or to the report server, where you can manage it and share it with others.
- **Choose a presentation format** when you open the report, or after you open the report. You can select Web-oriented, page-oriented, and desktop application formats. Formats include HTML, MHTML, PDF, XML, CSV, TIFF, Word, and Excel.
- **Set up subscriptions.** After you publish the report to the report server or a report server in SharePoint integrated mode, you can configure your report to run at a specific time, create a report history, and set up e-mail subscriptions.
- **Generate data feeds** from your report by using the Reporting Services Atom rendering extension.

> **NOTE**
> Published reports are managed on a report server or a report server in SharePoint integrated mode by a report server administrator. Report server administrators can define security, set properties, and schedule operations such as report history and e-mail report delivery. They can create shared schedules and shared data sources and make them available for general use. Administrators also manage all of the report server folders. The ability to perform management tasks depends on user permissions.

See Also

Start Report Builder

Install Report Builder

What's New in Reporting Services and Report Builder for SQL Server 2016

Describes the new features in this version of Reporting Services and Report Builder.

Tutorial: Creating a Quick Chart Report Offline

Introduces Report Builder and the wizards available to help you create reports. The tutorial provides a set of data for you to work with so you do not need to connect to a data source to get started.

Planning a Report (Report Builder)

Provides information on what you should consider before you start to build your report.

Report Authoring Concepts (Report Builder and SSRS)

Defines key concepts used in throughout Report Builder documentation.

Report Design View (Report Builder)

Explains the different panes and regions of report design view.

Shared Dataset Design View (Report Builder)

Explains the different panes and regions of shared dataset design view.

Keyboard Shortcuts (Report Builder)

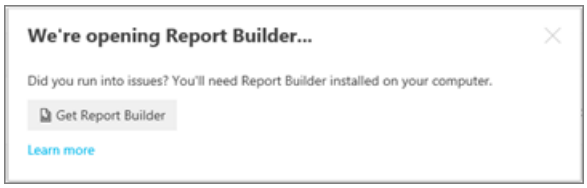
Outlines the shortcut keys available for navigating and designing reports in Report Builder.

Start Report Builder

3/24/2017 • 1 min to read • Edit Online

Microsoft SQL Server 2016 Report Builder is a stand-alone report authoring environment. With it, you can create paginated reports and publish them to Reporting Services installed in native or SharePoint integrated mode.

The first time you start Report Builder from the Reporting Services web portal or Reporting Services in SharePoint integrated mode, you're prompted to download it from the Microsoft Download Center.



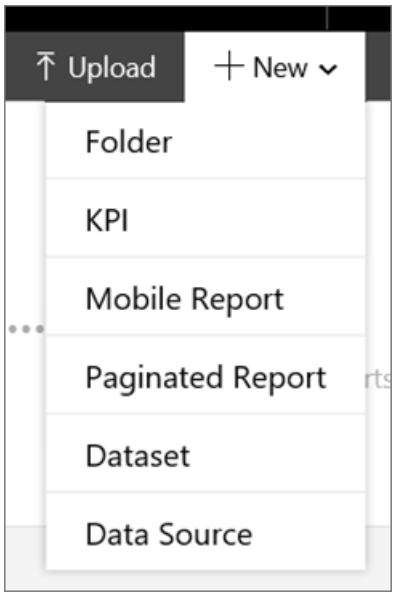
You or an administrator can also install Report Builder on your computer from the Microsoft Download Center. See "Install Report Builder with Systems Manager Server" in Install Report Builder for more details.

Report Builder isn't installed when you install SQL Server 2016; you need to download and install it separately.

When you start Report Builder from the web portal or SharePoint site, if an earlier version of Report Builder opens, contact your administrator, who can update the version on the web portal or SharePoint site.

To start Report Builder from the Reporting Services web portal

1. In your Web browser, type the URL for your report server in the address bar. By default, the URL is http://<*servername*>/reports.
2. In the top bar of the web portal, select **New** > **Paginated Report**.



The first time, you're prompted to install Report Builder.

After that first time, Report Builder opens, and you can create a paginated report or open a report from the report server.

To start Report Builder in SharePoint integrated mode

1. Navigate to the SharePoint site that contains the library you want.
2. Open the library.
3. Click **Documents**.
4. On the **New Document** menu, click **Report Builder Report**.

   The first time, this launches the SQL Server Report Builder Wizard. See Install Report Builder for more details.

   Report Builder opens, and you can create a paginated report or open a report on the report server.

   Note If the **New Document** menu does not list **Report Builder Report**, **Report Builder Model**, or **Report Data Source**, their content types need to be added to the SharePoint library. For more information, see Add Reporting Services Content Types to a SharePoint Library.

See Also

Report Builder in SQL Server 2016
Set default options for Report Builder

Set default options for Report Builder

3/24/2017 • 1 min to read • Edit Online

In Report Builder, you can set a number of useful defaults to make report authoring easier and faster. For example, if you can set or change the default report server, Report Builder saves your reports to the same report server automatically, unless you specify otherwise.

- In Report Builder, click **File** > **Options**.

UIElement List

Use this report server or SharePoint site by default
Your administrator may have configured this. The value can be a well-formed URL starting with http:// or https://. This setting determines which data source connections appear by default in the Table/Matrix and Chart wizards. In addition, your reports will be processed on this server and you can reference resources from this server.

If you select a different report server, you may need to restart Report Builder in order for this change to take affect.

Publish report parts to this folder by default
Report Builder will save report parts that you publish to this folder. If the folder does not exist yet and you have permissions to create folders on the report server, Report Builder will create this folder.

You do not need to restart Report Builder for this setting to take effect.

Show this number of recent sites and servers
Specify the number of recent sites and connections to show in the **Open Report** and **Save As Report** dialog boxes.

Show this number of recent shared datasets and data source connections
Specify the number of recent shared datasets and data source connections to show in the **Dataset Properties** dialog box and the **Choose a connection to a data source** page of the Data Regions Wizard.

Show this number of recent documents
Specify the number of recent documents to show when you click the Report Builder button.

Clear all recent item lists
Clear the current lists of recent sites and servers, shared datasets, shared data source connections, and documents.

See Also

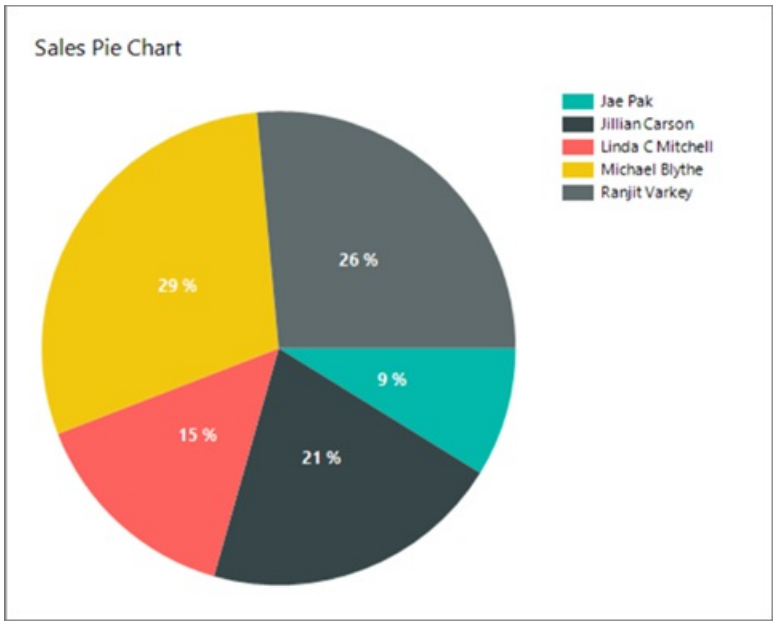
Start Report Builder

Tutorial: Create a Quick Chart Report Offline (Report Builder)

3/24/2017 • 8 min to read • Edit Online

In this tutorial, you use a wizard to create a pie chart in a Reporting Services paginated report in Report Builder. Then you add percentages and modify the pie chart a little.

You can do this tutorial two different ways. Both methods have the same outcome—a pie chart like the one in this illustration:



Prerequisites

Whether you use XML data or a Transact-SQL query, you need to have access to SQL Server 2016 Report Builder. You can start Report Builder from a Reporting Services report server in native mode or in SharePoint integrated mode, or you can download Report Builder from the Microsoft Download Center. For more information, see Install Report Builder.

Two Ways To Do This Tutorial

- Create the pie chart with XML data
- Create the pie chart with a Transact-SQL query that contains data

Using XML data for this tutorial

You can use XML data that you copy from this topic and paste into the wizard. You don't need to be connected to a Reporting Services report server in native mode or in SharePoint integrated mode, and you don't need access to an instance of SQL Server 2016.

Create the pie chart with XML data

Using a Transact-SQL query that contains data for this tutorial

You can copy a query with data included in it from this topic and paste it into the wizard. You will need the name of an instance of SQL Server 2016 and credentials sufficient for read-only access to any database. The dataset query in the tutorial uses literal data, but the query must be processed by an instance of SQL Server 2016 to return the

metadata that is required for a report dataset.

The advantage of using the Transact-SQL query is that all the other Report Builder tutorials use the same method, so when you do the other tutorials, you will already know what to do.

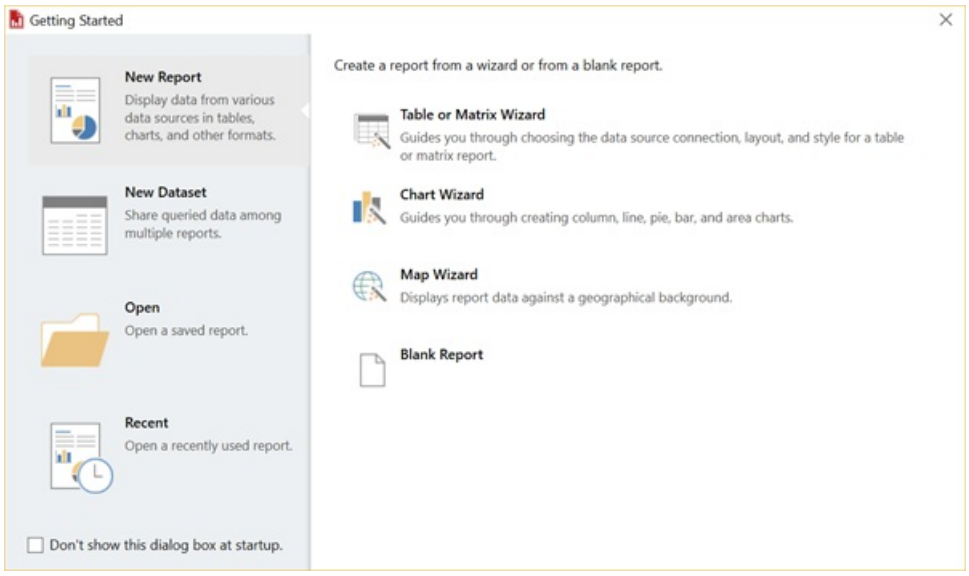
The Transact-SQL query does require a few other prerequisites. For more information, see Prerequisites for Tutorials (Report Builder).

Create the pie chart with a Transact-SQL query that contains data

Creating the pie chart with XML data

1. Start Report Builder from the Reporting Services web portal, or the report server in SharePoint integrated mode, or from your computer.

   The **Getting Started** dialog box appears.

   If the **Getting Started** dialog box does not appear, click **File** >**New**. The **New Report or Dataset** dialog box has most of the same contents as the **Getting Started** dialog box.

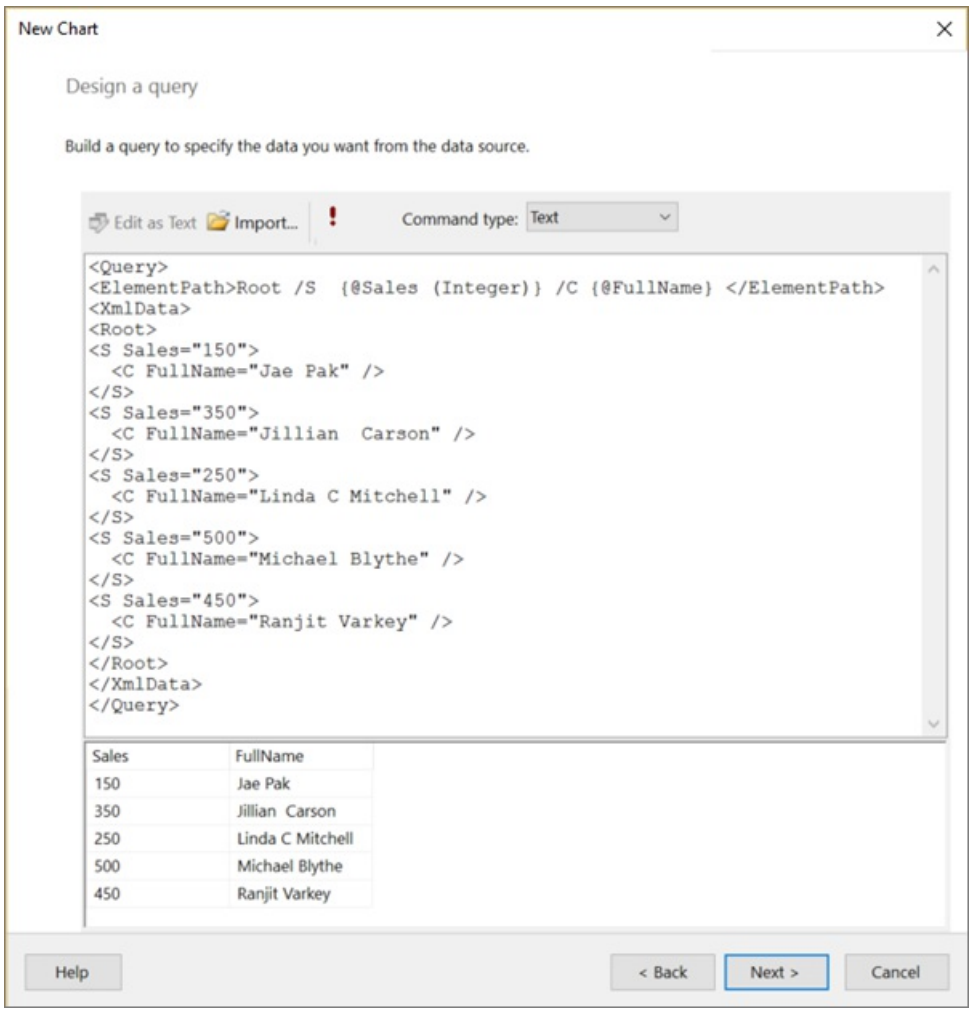
2. In the left pane, verify that **New Report** is selected.
3. In the right pane, click **Chart Wizard**, and then click **Create**.
4. In the **Choose a dataset** page, click **Create a dataset**, and then click **Next**.
5. In the **Choose a connection to a data source** page, click **New**.

   The **Data Source Properties** dialog box opens.

6. You can name a data source anything you want. In the **Name** box, type **MyPieChart**.
7. In the **Select connection type** box, click **XML**.
8. Click the **Credentials** tab, select **Use current Windows user. Kerberos delegation might be required**, and then click **OK**.
9. In the **Choose a connection to a data source** page, click **MyPieChart**, and then click **Next**.
10. Copy the following text and paste it in the large box in the top of the **Design a query** page.

```
<Query>
<ElementPath>Root /S  {@Sales (Integer)} /C {@FullName} </ElementPath>
<XmlData>
<Root>
<S Sales="150">
  <C FullName="Jae Pak" />
</S>
<S Sales="350">
  <C FullName="Jillian  Carson" />
</S>
<S Sales="250">
  <C FullName="Linda C Mitchell" />
</S>
<S Sales="500">
  <C FullName="Michael Blythe" />
</S>
<S Sales="450">
  <C FullName="Ranjit Varkey" />
</S>
</Root>
</XmlData>
</Query>
```

11. (Optional) Click the **Run** button (**!**) to see the data your chart will be based on.

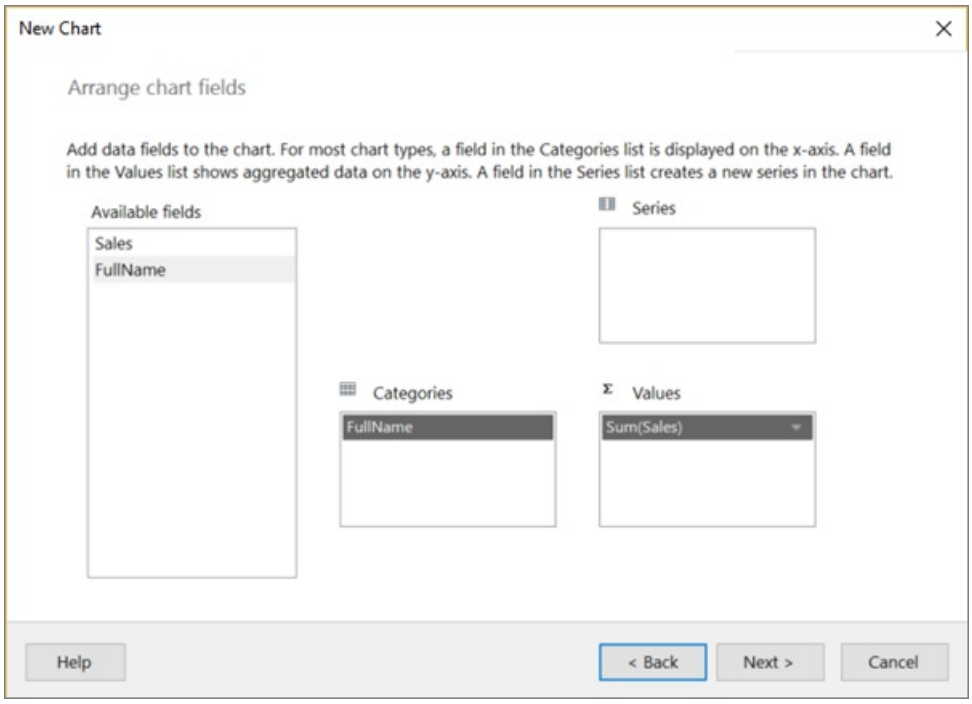


12. Click **Next**.

13. In the **Choose a chart type** page, click **Pie**, and then click **Next**.

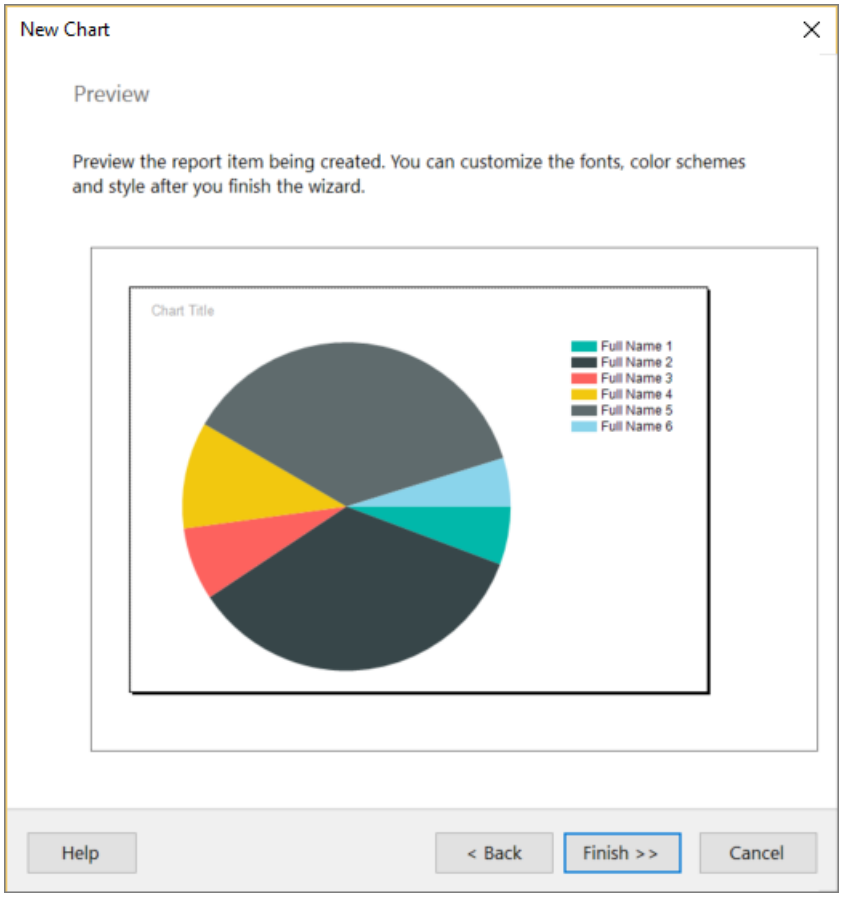
14. In the **Arrange chart fields** page, double-click the **Sales** field in the **Available fields** box.

    Note that it automatically moves to the **Values** box, because it is a numerical value.



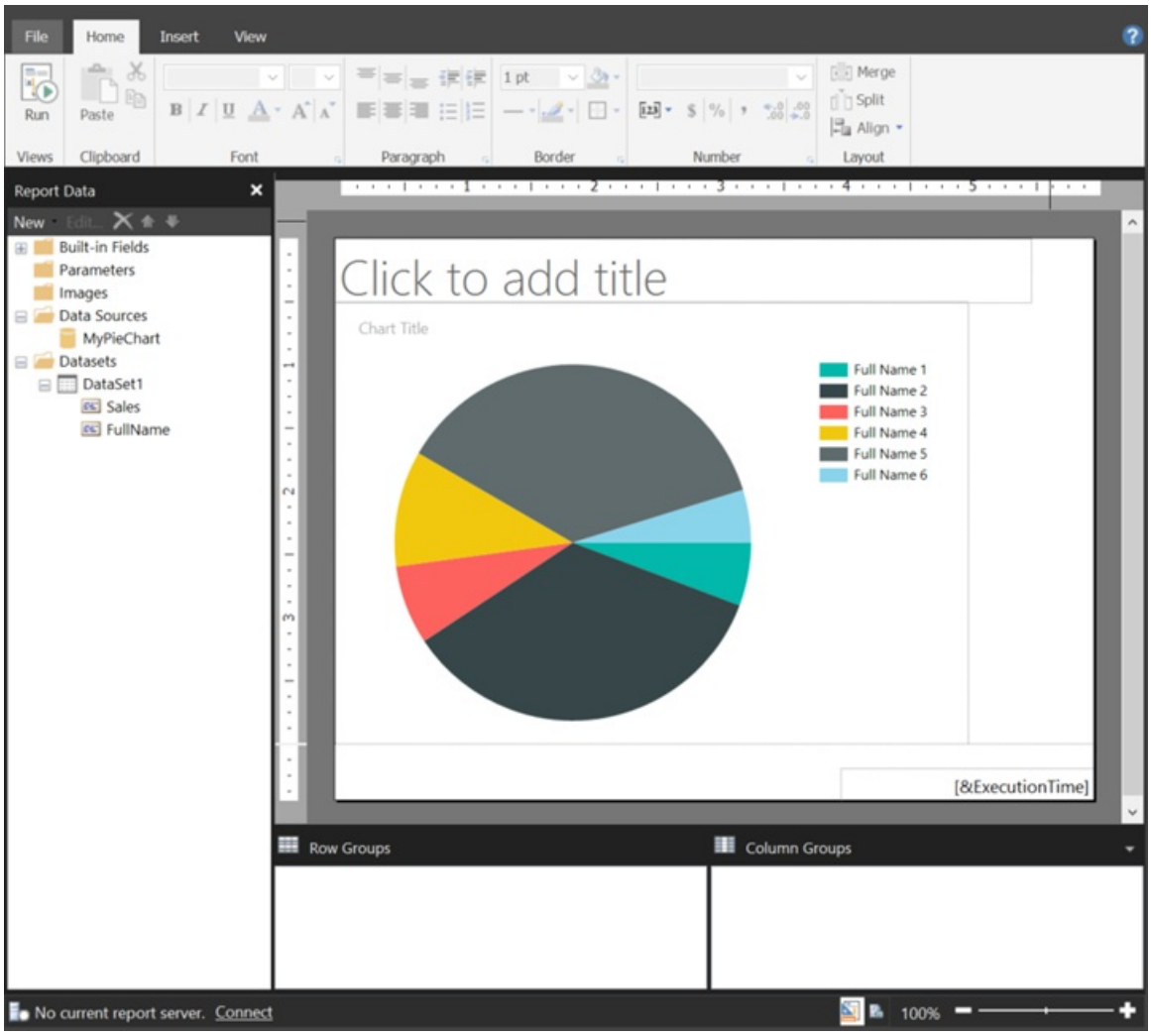
15. Drag the **FullName** field from the **Available fields** box to the **Categories** box (or double-click it; it will go to the **Categories** box), and then click **Next**.

    The Preview page shows your new pie chart with representational data. The legend reads Full Name 1, Full Name 2, etc., rather than the salespeople's names, and the size of the slices of pie are not accurate. This is just to give you an idea of what your report will look like.

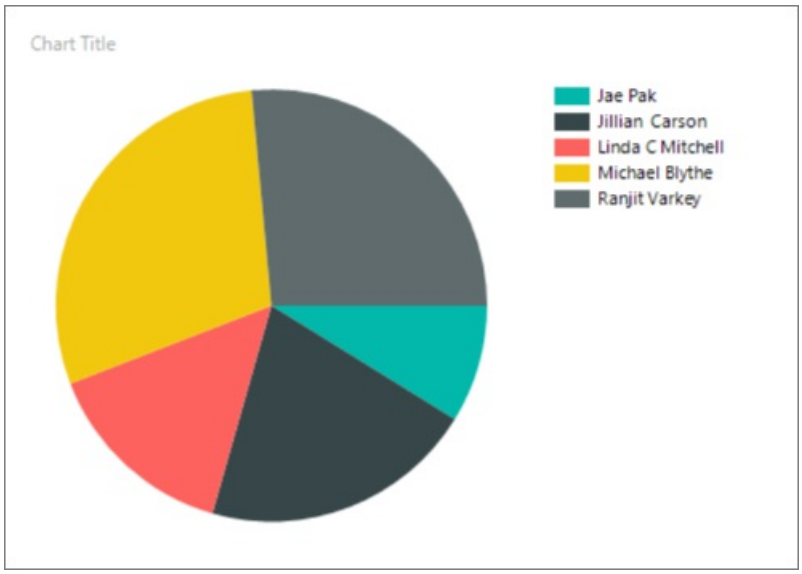


16. Click **Finish**.

    Now you see your new pie chart report in Design View, still with representational data.



17. To see your actual pie chart, click **Run** on the **Home** tab of the Ribbon.



18. To continue modifying your pie chart, go to After You Run the Wizard in this article.

Creating the pie chart with a Transact-SQL query

1. Start Report Builder from the Reporting Services web portal, from the report server in SharePoint integrated mode, or from your computer.

   The **Getting Started** dialog box appears.

> **NOTE**
> If the **Getting Started** dialog box does not appear, click **File** >**New**. The **New Report or Dataset** dialog box has most of the same contents as the **Getting Started** dialog box.

2. In the left pane, verify that **New Report** is selected.
3. In the right pane, click **Chart Wizard**, and then click **Create**.
4. In the **Choose a dataset** page, click **Create a dataset**, and then click **Next**.
5. In the **Choose a connection to a data source** page, select an existing data source or browse to the report server and select a data source, and then click **Next**. You may need to enter a user name and password.

   > **NOTE**
   > The data source you choose is unimportant, as long as you have adequate permissions. You will not be getting data from the data source. For more information, see Prerequisites for Tutorials (Report Builder).

6. On the **Design a Query** page, click **Edit as Text**.
7. Paste the following query into the query pane:

```
SELECT 150 AS Sales, 'Jae Pak' AS FullName
UNION SELECT 350 AS Sales, 'Jillian Carson' AS FullName
UNION SELECT 250 AS Sales, 'Linda C Mitchell' AS FullName
UNION SELECT 500 AS Sales, 'Michael Blythe' AS FullName
UNION SELECT 450 AS Sales, 'Ranjit Varkey' AS FullName
```

8. (Optional) Click the Run button (**!**) to see the data your chart will be based on.
9. Click **Next**.
10. In the **Choose a chart type** page, click **Pie**, and then click **Next**.
11. In the **Arrange chart fields** page, double-click the **Sales** field in the **Available fields** box.

    Note that it automatically moves to the **Values** box, because it's a numerical value.
12. Drag the **FullName** field from the **Available fields** box to the **Categories** box (or double-click it; it will go to the **Categories** box), and then click **Next**.
13. Click **Finish**.

    You're now looking at your new pie chart report on the design surface. What you see is representational. The legend reads Full Name 1, Full Name 2, etc., rather than the salespeople's names, and the size of the slices of pie are not accurate. This is just to give you an idea of what your report will look like.
14. To see your actual pie chart, click **Run** on the **Home** tab of the Ribbon.

After You Run the Wizard

Now that you have your pie chart report, you can play with it. On the **Run** tab of the Ribbon, click **Design**, so you can continue modifying it.

Make the chart bigger

You may want the pie chart to be bigger.

- Click the chart, but not on any element in the chart, to select it and drag the lower-right corner to resize it.

Notice the design surface gets larger as you drag.

Add a report title

1. Select the words **Chart title** at the top of the chart, and then type a title, such as **Sales Pie Chart**.
2. With the title selected, in the Properties pane, change **Color** to **Black** and **FontSize** to **12pt**.

Add percentages

1. Right-click the pie chart and select **Show Data Labels**. The data labels appear within each slice on the pie chart.
2. Right-click the labels and select **Series Label Properties**. The **Series Label Properties** dialog box appears.
3. In the **Label data** box, type **#PERCENT{P0}**.

   The **{P0}** gives you the percentage without decimal places. If you type just **#PERCENT**, your numbers will have two decimal places. **#PERCENT** is a keyword that performs a calculation or function for you; there are many others.
4. Click **Yes** to confirm you want to set **UseValueAsLabel** to **False**.
5. On the **Font** tab, select **Bold** and change **Color** to **White**.
6. Click **OK**.

   For more information about customizing chart labels and legends, see Display Percentage Values on a Pie Chart (Report Builder and SSRS) and Change the Text of a Legend Item (Report Builder and SSRS).

What's Next?

Now that you have created your first report in Report Builder, you are ready to try the other tutorials and to start creating reports from your own data. To run Report Builder, you need permission to access your data sources, such as databases, with a *connection string*, which actually connects you to the data source. Your system administrator will have this information and can set you up.

To work through the other tutorials, you need the name of an instance of SQL Server 2016 and credentials sufficient for read-only access to any database. Your system administrator can also set that up for you.

Finally, to save your reports to a report server or a SharePoint site that is integrated with a report server, you need the URL and permissions. You can run any report you create directly from your computer, but reports have more functionality when run from the report server or SharePoint site. You need permissions to run your reports or others from the report server or SharePoint site where they are published. Talk to your system administrator to obtain access.

It may help to read about some of the concepts and terms before you get started. See Report Authoring Concepts (Report Builder and SSRS). Also, spend some time planning, before you create your first report. It will be time well spent. See Planning a Report (Report Builder).

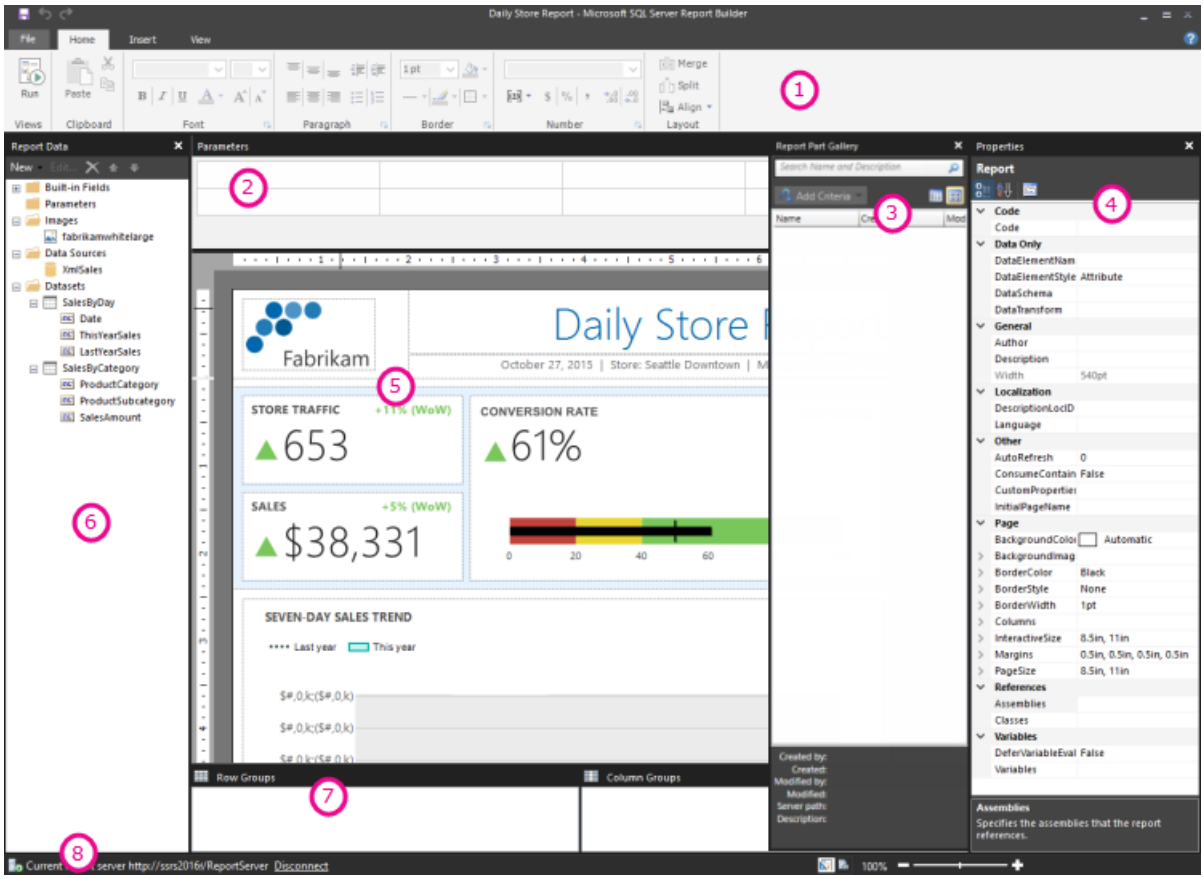
See Also

Report Builder Tutorials
Report Builder in SQL Server 2016

Report Design View (Report Builder)

3/24/2017 • 9 min to read • Edit Online

The Report Builder window is designed to help you easily organize your report resources and quickly build the paginated reports you need. The design surface is at the center of the window, with the ribbon and the panes around it. The design surface is where you add and organize your report items. This article explains the panes you use to add, select, and organize your report resources, and change report item properties.



1. Ribbon
2. Parameters pane
3. The Report Part Gallery
4. Properties pane
5. Report design surface
6. The Report Data pane
7. The Grouping Pane
8. Current report status bar

Parameters pane

With report parameters, you can control report data, connect related reports together, and vary report presentation. The Parameters pane provides a flexible layout for the report parameters.

Read more about Report Parameters (Report Builder and Report Designer).

The Report Design Surface

The Report Builder report design surface is the main work area for designing your reports. To place report items such as data regions, subreports, text boxes, images, rectangles, and lines in your report, you add them from the ribbon or the Report Part Gallery to the design surface. There, you can add groups, expressions, parameters, filters, actions, visibility, and formatting to your report items.

You can also change the following:

- The report body properties, such as border and fill color, by right-clicking the white area of the design surface, outside any report items, and clicking **Body Properties**.
- The header and footer properties, such as border and fill color, by right-clicking the white area of the design surface in the header or footer area, outside any report items, and clicking **Header Properties** or **Footer Properties**.
- The properties of the report itself, such as page setup, by right-clicking the gray area around the design surface and clicking **Report Properties**.
- The properties of report items by right-clicking them and clicking **Properties**.

  For information about using the keyboard to manipulate items on the design surface, see Keyboard Shortcuts (Report Builder)

Design Surface Size and Print Area

The design surface size might be different from the page size print area you specify to print the report. Changing the size of the design surface will not change the print area of your report. No matter what size you set for the print area of your report, the full design area size does not change. For more information, see Rendering Behaviors (Report Builder and SSRS).

> **TIP**
> To display the ruler, on the **View** tab, select the **Ruler** check box.

The Report Data Pane

From the Report Data pane, you define the report data and report resources that you need for a report before you design your report layout. For example, you can add data sources, datasets, calculated fields, report parameters, and images to the Report Data pane.

After you add items to the Report Data pane, drag fields to report items on the design surface to control where data appears in the report.

> **TIP**
> If you drag a field from the Report Data pane directly to the report design surface instead of placing it in a data region such as a table or chart, when you run the report, you will see only the first value from the data in that field.

You can also drag built-in fields from the Report Data pane to the report design surface. When rendered, these fields provide information about the report, such as the report name, the total number of pages in the report, and the current page number.

Some things are automatically added to the Report Data pane when you add something to the report design surface. For example, if you add a report part from the Report Part Gallery, and the report part is a data region, the dataset is automatically added to the Report Data pane. For more information, see Report Parts and Datasets in

Report Builder. Also, if you embed an image in your report, it will be added to the Images folder in the Report Data pane.

> **NOTE**
> You can use the **New** button to add a new item to the Report Data pane. You can add multiple datasets from the same data source or from other data sources to the report. You can add shared datasets from the report server. To add a new dataset from the same data source, right-click a data source, and then click **Add Dataset**.

For more about items in the Report Data pane, see the following topics:

- Built-in Globals and Users References (Report Builder and SSRS)
- Report Parameters (Report Builder and Report Designer)
- Images (Report Builder and SSRS)
- Data Connections, Data Sources, and Connection Strings (Report Builder and SSRS)
- Report Embedded Datasets and Shared Datasets (Report Builder and SSRS)
- Dataset Fields Collection (Report Builder and SSRS)

The Report Part Gallery

The easiest way to create a report is to find an existing report part, like a table or chart, on the report server or a report server integrated into a SharePoint site.

Click **Report Parts** on the Insert tab to open the Report Part Gallery. There you can search for report parts to add to your report. You can filter the report parts by all or part of the name of the report part, who created it, who last modified it, when it was last modified, where it's stored, or what type of report part it is. For example, you could search for all charts created last week by one of your coworkers.

> **NOTE**
> To view the Report Part Gallery, you need to be connected to a server.

You can view the search results either as thumbnails or as a list, and sort the search results by name, created and modified dates, and creator. For more information, see Report Parts (Report Builder and SSRS)..

The Properties Pane (Report Builder)

Every item in a report, including data regions, images, text boxes, and the report body itself, has properties associated with it. For example, the BorderColor property for a text box shows the color value of the text box's border, and the PageSize property for the report shows the page size of the report.

These properties are displayed in the Properties pane. The properties in the pane change depending on the report item that you select.

To see the Properties pane, on the View tab, in the Show/Hide group, click Properties.

Changing Property Values

In Report Builder, you can change the properties for report items several ways:

- By clicking buttons and lists on the ribbon.
- By changing settings within dialog boxes.

- By changing property values within the Properties pane.

  The most commonly used properties are available in the dialog boxes and on the ribbon.

  Depending on the property, you can set a property value from a drop-down list, type the value, or click `<Expression>` to create an expression.

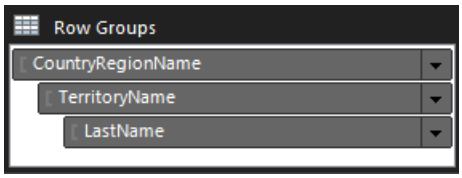
Changing the Properties Pane View

By default, properties displayed in the Properties pane are organized into broad categories, such as Action, Border, Fill, Font, and General. Each category has a set of properties associated with it. For example, the following properties are listed in the Font category: Color, FontFamily, FontSize, FontStyle, FontWeight, LineHeight, and TextDecoration. If you prefer, you can alphabetize all the properties listed in the pane. This removes the categories and lists all the properties in alphabetical order, regardless of category.

The Properties pane has three buttons at the top of pane: Category, Alphabetize, and Property Pages. Click the Category and Alphabetize buttons to switch between the Properties pane views. Click the **Property Pages** button to open the properties dialog box for a selected report item.

The Grouping Pane (Report Builder)

Groups are used to organize your report data into a visual hierarchy and to calculate totals. You can view the row and column groups within a data region on the design surface and also in the Grouping pane. The Grouping pane has two panes: Row Groups and Column Groups. When you select a data region, the Grouping pane displays all the groups within that data region as a hierarchical list: Child groups appear indented under their parent groups.



You can create groups by dragging fields from the Report Data pane and dropping them on the design surface or in the Grouping pane. In the Grouping pane, you can add parent, adjacent, and child groups, change group properties, and delete groups.

The Grouping pane is displayed by default but you can close it by clearing the Grouping pane check box on the View tab. The Grouping pane is not available for the Chart or Gauge data regions.

For more information, see Grouping Pane (Report Builder) and Understanding Groups (Report Builder and SSRS).

Previewing Your Report in Run Mode

In report design view, you are not working with the actual data but a representation of the data indicated by the field name or expression. When you want to see the actual data displayed in the context of the report that you designed, you can run the report to preview the data from the underlying database displayed in the report layout. Switching between designing and running your report allows you to adjust its design and see the results immediately. To preview your report, click **Run** in the **Views** group on the ribbon.

When you click **Run**, Report Builder connects to the report data sources, caches the data on your computer, combines the data and the layout and then renders the report in the HTML Viewer. You can run your report as often as you like while you continue to design it. When you are satisfied with your report, you can save the report to the report server where other individuals with the appropriate permissions can view your report.

Read more about Previewing a Report in Report Builder.

Running a Report with Parameters

When you run your report, it is processed automatically. If the report contains parameters, all the parameters must

have default values before the report can run automatically. If a parameter does not have a default value, when you run the report you need to choose a value for the parameter, and then click **View Report** on the Run tab. For more information, see Report Parameters (Report Builder and Report Designer).

Print Preview

When you preview a report in run mode, it resembles a report produced in HTML. The preview is not HTML, but the layout and pagination of the report is similar to HTML output. You can change the view to represent a printed report by switching to print preview mode. Click the **Print Preview** button on the **Run** tab. The report will display as though it were on a physical page. This view resembles the output produced by the Image and PDF rendering extensions. Print Preview is not an image or PDF file, but the layout and pagination of the report are similar to the output of those formats.

See Also

Finding, Viewing, and Managing Reports (Report Builder and SSRS)
Report Builder in SQL Server 2016

Previewing Reports in Report Builder

3/24/2017 • 5 min to read • Edit Online

While you create a Reporting Services paginated report, it is helpful to preview the report often to verify that the report displays what you want. To preview your report, click **Run**. The report renders in preview mode.

Report Builder improves the preview experience by using edit sessions when connected to a report server. The edit session creates a data cache and makes the datasets in the cache available for repeated report previews. An edit session is not a feature that you interact with directly, but understanding when the cached dataset is refreshed will help you improve performance when you preview a report and understand why the report renders faster or slower.

Other benefits of edit sessions are the abilities to edit reports that use embedded data sources or reference items such as images or subreports that are stored on the report server.

> **NOTE**
> There are some differences between previewing in Report Builder and viewing in a browser. For example, a calendar control, which is added to a report when you specify a Date/Time type parameter, is different in Report Builder and in a browser.

Improving Preview Performance

How you create and update reports affects how fast the report renders in preview. The first time that you preview a report that relies on a server reference, an edit session is created for you and the data used when the report is run is added to a data cache that is stored on the report server. When you make changes to the report that does not affect the data, the cached copy of the data is used by the report. This means that you will not see data change each time you preview the report. If you want new data, click the **Refresh** button on the ribbon.

The following actions cause the cache to be refreshed and slow down report rendering the next time you preview the report:

- Add, change, or delete a dataset. The cached dataset contains all the datasets that a report uses and modification to any dataset invalidates the cached dataset. This includes changing the name, query, or fields in the dataset.

  > **NOTE**
  > If the dataset has a large number of fields that you do not expect to use, you should consider updating the dataset to omit those fields. Although this creates a new edit session and the first preview of the report is slower, there smaller cached dataset is overall beneficial to the performance of the report server.

- Add, change, or delete a data source. This includes changing the name or properties of the data source, the data extension of the data source, or the properties of the connection to the data source.
- Change the shared data source that the report uses to a different data source.
- Change the language of the report.
- Change the assemblies or custom code that the report uses.
- Add, change, or delete the query parameters in the report or parameter values.

Changes to the report layout and data formatting do not affect the cached dataset. You can do the following actions without refreshing the cached dataset:

- Add or remove data regions such as tables, matrices or charts.
- Add or delete columns from the report. All the fields in the dataset are available to use in the report. Adding or removing fields in the report has no effect on the dataset.
- Change the order of fields in tables and matrices.
- Add, change, or delete row and column groups.
- Add, change, or delete formatting of data values in fields.
- Add, change, or delete images, lines, or text boxes.
- Change page breaks.

The edit session is created the first time that you preview a report. By default, an edit session lasts 7200 seconds (2 hours). The session is reset to two hours every time you run the report. When the edit session expires, the data cache is deleted. If the edit session expires, one is automatically created again the next time that you preview the report. The expiration time for edit sessions is configurable. If you find that two hours is too long or too short, contact the administrator of the report server.

By default, the data cache can hold up to five datasets. If you use many different combinations of parameter values, the report might need more data. This requires the cache be refreshed and the report renders more slowly the next time that you preview it. The number of entries in the cache is configurable by the administrator of the report server.

Concurrency of Report Updates

Frequently, you preview a report as a step in updating and then saving a report to a report server. When you are updating a report, it is possible that someone else is updating and then saving the report at the same time. The report that is saved last is the version of report that is available for future viewing and updating. This means that the version of the report that you previewed might not be the version you reopen. You have the option to save the report with a new name by using the **Save As** option on the Report Builder menu.

External Report Items

Your report might include items such as shared data sources, external images, and subreports that are stored separately from the report. Because the items are stored separately is possible that they can be moved to a different location on the report server or deleted. If this happens, your report could fail to preview. You can either update the report to indicate the updated location of the item or if the item was deleted, replace it with an existing item, or remove the reference to the item it from the report.

If a subreport used by your report is changed after your edit session was created, the report will not render in preview. To successfully preview the report, you should save the report or click **Refresh** to get fresh data.

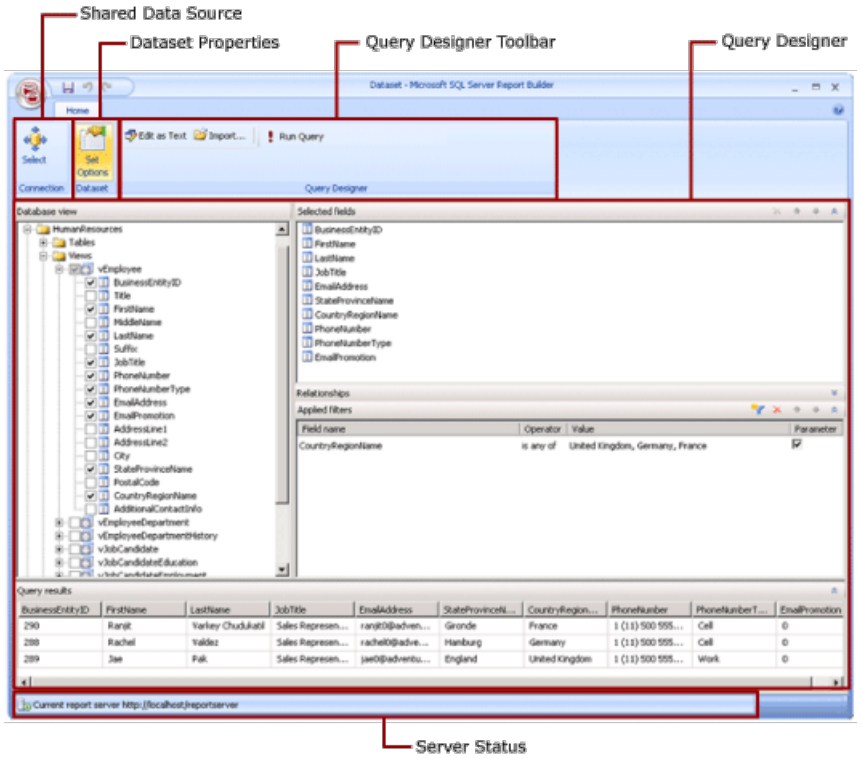
See Also

Report Datasets (SSRS)
Formatting Report Items (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)
Charts (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)
Saving Reports (Report Builder)

Shared Dataset Design View (Report Builder)

3/24/2017 • 4 min to read • Edit Online

In a report, a dataset represents report data that is returned from running a query on an external data source. Shared datasets are published on a report server and can be used by multiple reports. You can create datasets to share with others. In the Shared Dataset Design window, you select a shared data source, specify properties for the shared dataset, and create a query in the query designer.



For more information about working with data in a report, see Report Datasets (SSRS).

The Ribbon

The Ribbon helps you quickly find the commands that you need to complete a task. Commands are organized into the following logical groups: Connection, Dataset, and Query Designer.

Connection

Use the **Select** button in the Connection group to select a shared data source in your report, or browse to a shared data source on the report server.

> **NOTE**
> A shared dataset must be based on a shared data source. If the data source you need isn't available, you need to create one on the report server. For more information, see Create, Modify, and Delete Shared Data Sources (SSRS) .

For more information, see Data Connections, Data Sources, and Connection Strings in Report Builder.

Dataset

Use the **Set Options** button to set shared dataset properties. These include the following:

- Fields. You can add a field or edit a field in the field collection.

- Data options. You can set options that affect match criteria and sort order, such as case sensitivity and collation.
- Filters. You can define filters that limit the data in a report after it is retrieved from the data connection.
- Parameters. You can add a parameter or edit parameter options. For example, you can specify a default value for each parameter so that you can create a cache refresh plan for this shared dataset on the report server.

  The values that you set become part of the shared dataset definition on the report server. When a report author includes this shared dataset in a report, the options that you specify apply to that dataset instance.

  After a shared dataset is added to a report, a report author can override the following options: collation, case sensitivity, accent sensitivity, kanatype sensitivity, width sensitivity, subtotals. They can also create additional dataset filters to limit the data in the report.

  For more information, see Report Embedded Datasets and Shared Datasets (Report Builder and SSRS).

  For more information about cache refresh plans, see Cache Shared Datasets (SSRS).

Query Designer

Use the query designer toolbar to help build a query that specifies which data to retrieve from the data connection. The toolbar that you see depends on the query designer that is associated with the data source type from the data connection.

For more information, see the topic that corresponds to the data source type in Add Data from External Data Sources (SSRS).

The Query Designer Surface

A query designer helps you to build a query in the syntax that is required by the external data source.

Some data source types provide a graphical query designer that you can use to explore the metadata on an external data source. You can interactively drag names from the metadata pane to the query design surface, or interactively select the names to use.

Some data source types support a text-based query designer that you can use to paste in queries that you have created in other tools, such as SQL Server Management Studio.

Each data source type has specific requirements for the query that will work against the external data source. For more information, see the topic that corresponds to the data source type in Add Data from External Data Sources (SSRS) and Data Sources Supported by Reporting Services (SSRS).

Viewing Query Results

In shared dataset design view, you are building a query that will retrieve data from the data connection when the report is processed.

Run the query to see example data from the data connection to verify that the query returns the type of data that you expect. The columns in the result set come from the metadata for data schemas from the data connection. The column names become the dataset field collection. The values of the data that you see in the query result set is design time data. After you save the shared dataset as a shared dataset definition on the report server, only the query text is saved. The data in the query result set is not saved.

When a report author adds this shared dataset to a report, a pointer to the dataset definition on the report server is added. In the report, the dataset field collection appears in the Report Data pane. The query text is not available.

The credentials that you use to run a query are separate from the credentials that are used to preview a report or to

run a report from the report server. For more information, see Specify Credential and Connection Information for Report Data Sources.

Running a Report with Parameters

When your query includes query variables, dataset parameters are created automatically for you. In turn, when you finish building the dataset query, report parameters that are set to dataset parameters are created automatically.

If a report contains parameters, all the parameters must have default values before the report can run automatically. If a parameter does not have a default value, when you run the report you must choose a value for the parameter, and then click **View Report** on the **Run** tab.

For more information, see Report Parameters (Report Builder and Report Designer).

Saving the Shared Dataset

To save the query that you built, on the **Report Builder** button, click **Save** or **Save As**. Navigate to the appropriate folder on the report server and save the shared dataset definition. The shared dataset is not available to others until you save it to the report server.

See Also

Report Datasets (SSRS)
Filter, Group, and Sort Data (Report Builder and SSRS)
Report Parameters (Report Builder and Report Designer)

Keyboard Shortcuts (Report Builder)

3/24/2017 • 3 min to read • Edit Online

When you create and edit paginated Reporting Services paginated reports in Report Builder, you have access to all the shortcut keys for navigation that are provided by the Windows environment. You can select from several standard keyboard mapping schemes.

You can navigate through the Report Builder window, or select and change options in dialog boxes, without using the mouse.

Report Builder provides many right-click menus for quick access to frequently used dialog boxes and commands. Right-click menus are dynamic and may contain different options, depending on the user configuration and the current objects. As you explore Report Builder, make sure you right-click objects to explore the menu options and learn about these shortcuts.

> **NOTE**
> Keyboard shortcuts can contain the SHIFT, ALT, and/or CTRL keys in combination with letters, or use function keys. For example, F1 opens the Report Builder Help file.

Keyboard Shortcuts in Report Builder

DESCRIPTION	KEY COMBINATION
Open the Report Builder menu.	ALT+F
Create a report.	CTRL+N
Open a report.	CTRL+O
Save a report.	CTRL+S, ALT+S
Move between areas of Report Builder, such as the Ribbon, design surface, Grouping pane, Properties pane, and the Data pane.	F6, SHIFT+F6
Move to the Ribbon.	ALT
Move to the Home tab on the Ribbon.	ALT+H
Move to the Insert tab on the Ribbon.	ALT+I
Move to the View tab on the Ribbon.	ALT+V
Activate the division lines between areas such as the line between the Grouping pane and the design surface.	TAB

DESCRIPTION	KEY COMBINATION
Move the division lines between areas.	Any arrow key. Use the UP and DOWN arrows to move the line between the Grouping pane and design surface. Use the RIGHT and LEFT arrows to move the line between the Report Data pane and the design surface.
Move within an area of Report Builder, such as moving within a Ribbon tab.	TAB, SHIFT+TAB
Move between Ribbon tabs.	RIGHT ARROW, LEFT ARROW
Navigate the drop-down menus.	Any arrow key
Select items on the design surface.	TAB, SHIFT+TAB
Move a selected item on the design surface.	Any arrow key
Move a selected item on the design surface, in small increments.	CTRL with an arrow key
Resize a selected item on the design surface.	SHIFT with an arrow key
Resize a selected item on the design surface, in small increments.	CTRL + SHIFT with an arrow key
Align the edges of multiple selected items.	ALT+H+A
Run the report.	F5, ALT+H+R
Switch to design mode.	F8
Copy	CTRL+C, ALT+H+C
Cut	CTRL+X, ALT+H+X
Paste	CTRL+V, ALT+H+V
Select all.	CTRL+A
Undo	CTRL+Z, ALT+Z
Redo	CTRL+Y, ALT+Y
Zoom in.	CTRL+Plus sign (+)
Zoom out.	CTRL+Minus sign (-)
Select and unselect a checkbox on the View tab.	Spacebar
Display the font family list, the font size list, or the border point size list, on the Home tab.	ALT+DOWN ARROW

DESCRIPTION	KEY COMBINATION
Display the font color list.	DOWN ARROW
Display Help.	F1
Merge table cells.	ALT+H+M
Split a table cell.	ALT+H+U
Insert a table.	ALT+I+A
Insert a matrix.	ALT+I+M
Insert a list.	ALT+I+L
Insert a chart.	ALT+I+C
Insert a gauge.	ALT+I+G
Insert a map	ALT+I+P
Insert a data bar.	ALT+I+D
Insert a sparkline.	ALT+I+K
Insert an indicator.	ALT+I+U
Insert a text box.	ALT+I+T
Insert an image.	ALT+I+I
Insert a line.	ALT+I+N
Insert a rectangle.	ALT+I+E
Insert a subreport.	ALT+I+S
Insert report header.	ALT+I+H
Insert report footer.	ALT+I+F
Pan map view area when a map viewport is selected.	CTRL with an arrow key
Zoom in on map view area when a map viewport is selected.	CTRL +Plus sign (+)
Zoom out on map view area when a map viewport is selected.	CTRL +Minus sign (-)
Accelerator mode for pan and zoom in map view area when a map viewport is selected.	CTRL +Left click

See Also

Finding, Viewing, and Managing Reports (Report Builder and SSRS)

3/24/2017 • 11 min to read • Edit Online

In Report Builder, you can browse folders on a report server or SharePoint site to find reports, shared data sources, models, and other related report items and browse your computer to find local reports. To make it easier to find reports, Report Builder maintains a list of recently used servers and sites and provides direct access to the Desktop, My Documents and My Computer folders in the file system of your computer.

In Report Designer, you can also browse your computer to find local reports. After you deploy reports to a report server or SharePoint site, you can browse the report server by using Report Manager or search the SharePoint site to find reports. Reports and related items remain available locally after they are deployed.

> **NOTE**
> You can use Report Builder in local mode or connected to a report server. Certain limitations apply when you do not have an active connection to a report server.

To locate a report on a report server or SharePoint site from Report Builder, you must provide URL to the report server or SharePoint site. When you first install Report Builder you can specify the URL to use. This is the server or site that Report Builder connects to by default when you save or open reports.

Reports can be previewed in Report Builder and Report Designer when you create or update reports and viewed and managed on a report server by using Report Manager or on a SharePoint site that is integrated with Reporting Services by using the built-in SharePoint tools and features after you publish the reports. For more information, see Previewing Reports in Report Builder and Previewing Reports.

When you preview reports in Report Builder and Report Designer, or view reports in Report Manager or a SharePoint site, the data is refreshed and the reports display the current data from the data source that the report uses. If you want to view a report without refreshing its data, you can use report history and cached data with published reports. You cannot use these features when previewing reports in Report Builder and Report Designer.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Finding and Viewing Reports in Report Builder

To find a report that you want to work with or to select a shared data source, image, or subreport to use in a report, browse your computer, folders on a report server, or SharePoint site integrated with Reporting Services.

To find reports on a report server, you must specify a URL for the report server and have the appropriate permissions on the folders that enable you to read and save report items. Ask the system administrator for the report server for the appropriate URL and permissions.

After you find and open the report in Report Builder, you can preview it and make changes. When you preview it, you see the current data. For more information, see Previewing Reports in Report Builder.

Report Builder can help you with the following tasks:

- **Finding reports** When you browse for a report, you can use the familiar Microsoft Office-style **Open File** dialog box that is customized for Report Builder. You can browse the folders on a report server or on a file system, including My Reports, Sites and Servers, Desktop, My Documents, My Computer. Sites and Servers provides a recently used server list.
- **Finding shared data sources** When you browse for a shared data source, you can pick from a recently used list, or browse to another folder on the same report server as the report.
- **Viewing reports** You preview a report in Report Builder when creating or updating reports. When Report Builder is connected to a report server the report server loads and processes the report; otherwise, the reports are processed locally. The report viewer in Report Builder displays the rendered report.

Viewing and Managing Reports on a Report Server

You use Report Manager to view and manage reports on the report server. Browse the folders on the server to locate reports, run reports to view them in a browser, and perform management tasks.

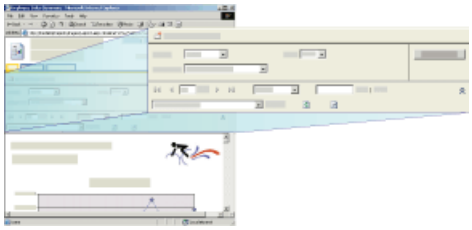
Report Manager you can help you with following management tasks:

- View and update the properties of reports, shared data sources, and other report items.
- Upload reports and create new shared data sources for reports.
- Create schedules to run reports at specified times and intervals.
- Create, change, or delete subscriptions to reports.
- Create report history and specify the number of report snapshots to keep in the report history.
- Create new folders on the server to organize your reports the way you want.

  Some of these tasks might be done for you by the administrator of the report server. To learn more about tasks performed on a report server, see Reporting Services Report Server (Native Mode).

  Report Manager typically contains folders, reports, data sources, and report models as well as the My Reports folder. My Reports is a personal workspace that you can use to store and work with reports that you own. Other report server folders are public and typically require users to have advanced permissions to add to or modify folder contents. You can create folders within My Reports to further organize your reports. For more information, see Using My Reports (Report Builder and SSRS).

  Report Manager displays reports in the Reporting Services HTML Viewer. The HTML Viewer provides a framework for viewing reports in HTML and includes a report toolbar, a parameter section, a credentials section, and a document map. The report toolbar provides page navigation, zoom, refresh, search, export, print, and data feed functionality. The report toolbar also appears in a browser window at the top of a report when you access reports through a URL. Print functionality is optional and must be turned on by your administrator. When it is available, a Printer icon appears on the report toolbar. The following illustrations show the report toolbar in a Report Manager window and the report toolbar features close up.



Report Manager window

Report toolbar

After you run a report, you can export it to another format, such as Microsoft Excel or PDF. You can also export the report using a data rendering extension such as the Comma-Separated Value (CSV) rendering extension and then use the CSV data file as input to another application. For more information about exporting reports, see Export Reports (Report Builder and SSRS) and Export a Report as Another File Type (Report Builder and SSRS).

The easiest way to select and run a report is to open Report Manager and then search for or browse to the report that you want to view. For step-by-step instruction on how to open reports, see Open and Close a Report (Report Manager).

After you run a report, you can refresh it to see new data.

Refreshing Reports

Report data frequently changes and you might want to refresh the report to view the newest data. You can refresh a report in three different ways.

OPTION	RESULT
Refresh button on the browser window	Displays the report stored in the session cache. A session cache is created when a user opens a report. Reporting Services uses browser sessions to maintain a consistent viewing experience while a report is open.
	When you click the **Refresh** button on the report toolbar, the report server re-runs the query and updates report data if the report runs on-demand. If the report is cached or is a snapshot, **Refresh** displays the report that is stored in the report server database.
CTRL+F5 keyboard combination	Produces the same result as clicking the **Refresh** button on the report toolbar.

Viewing and Managing Report Server Items from a SharePoint Site

When the system administrator configures a report server to run in SharePoint integrated mode, you can view and manage reports and other report server items from a SharePoint site.

The SharePoint site includes pages to set data source properties, report history, report processing options, schedules, subscriptions, report parameters, and create shared schedules. You can manage report server items on a SharePoint site the same way you create and manage them from other tools in SQL Server.

To access the application pages, select item-specific actions from a drop-down menu on a report or other report server item that you previously added to a SharePoint library. Depending on the item and your permissions, you might also be able to create reports in Report Builder, generate models, and set model item security.

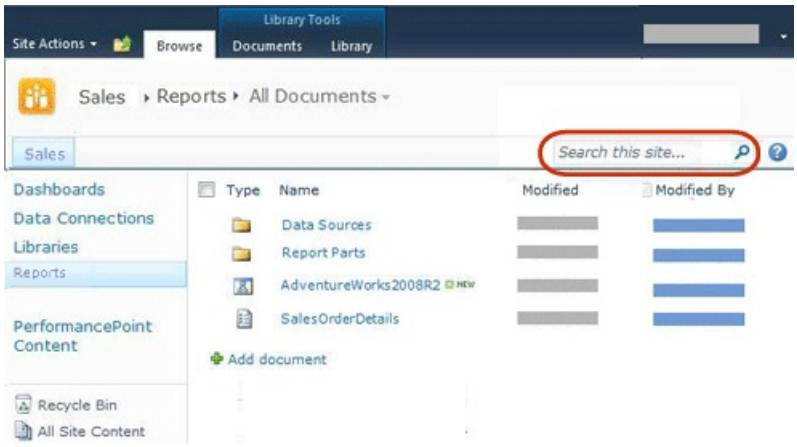
For more information about Reporting Services and SharePoint technology, see Configuration and Administration of a Report Server (Reporting Services SharePoint Mode) in SQL Server Books Online on msdn.microsoft.com.

Finding Report Server Items on a SharePoint Site

Before you can set properties, you must first be able to locate the item. Report server items are always stored in libraries or in a folder within a library.

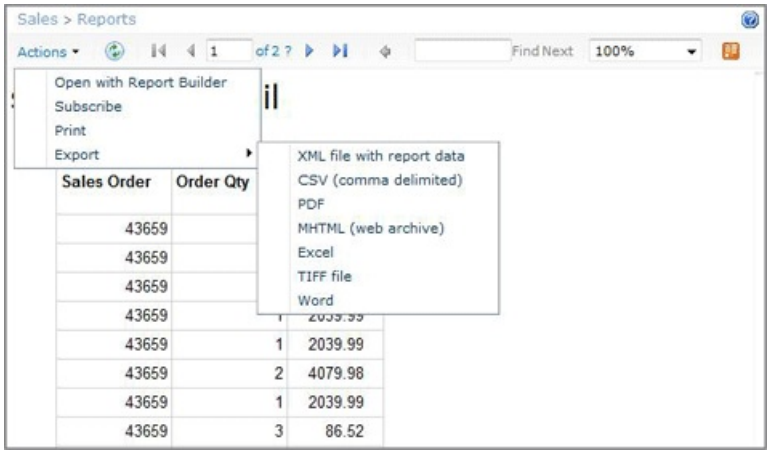
When you access the SharePoint site, you see the Browse page and the Library Tools tab. The Browse page lists the libraries and the content of the selected library. You can view the report, report models, and other items in the library, explore folders, and search the site to locate items.

To distinguish report server items from other items on a SharePoint site, you can use the icon to visually identify an item, or pause the mouse cursor over the type and read the file extension. The following image shows folders, a report model, and a report definition in the **Reports** library:



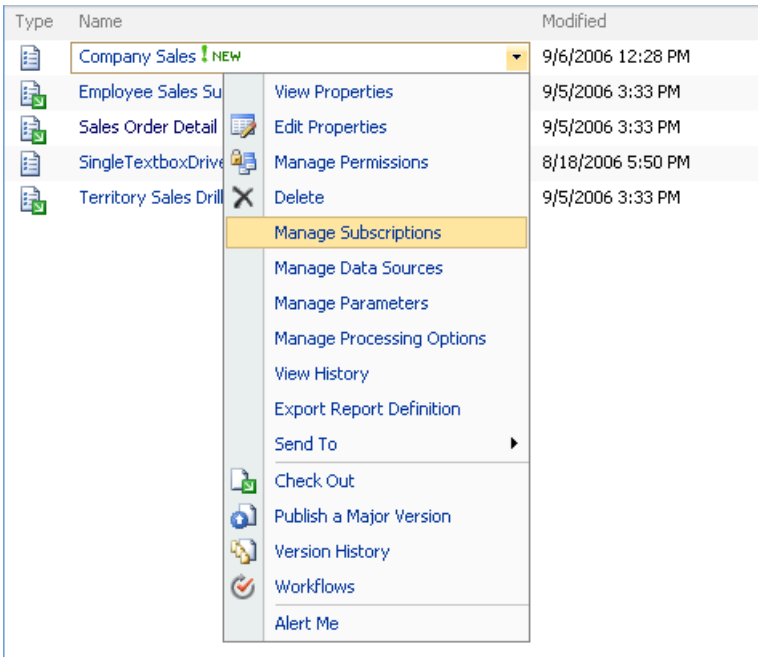
Viewing Reports

Report definitions (.rdl files) that you upload to a SharePoint library are viewed through a Report Viewer Web Part that is installed by the Reporting Services Add-in. An .rdl file association is defined automatically when you install the add-in. When you select a report, it opens automatically in the Web Part. After the report is open, you can use the report toolbar that is included in the Web Part to navigate pages, search, zoom, and print the report. The toolbar includes the Export Data Feed option to export the report as an Atom data feed and an **Actions** menu with options to print, subscribe, and export the report to different formats such as PDF, Word, and Excel. From the **Actions** menu you can also open the report in Report Builder. The following image shows a report and the options of the Export options in the **Action** menu.



Managing Items Through Actions

Management tasks are supported through actions on a drop-down menu for each item. Depending on your permissions, each item has common actions that are standard for items that are stored in a SharePoint library. **View Properties** and **Edit Properties** are examples of common actions. Custom actions provide item-specific management functionality. The following image shows the actions for a report definition. Examples of custom actions for a report definition include **Manage Subscriptions** and **Manage Processing Options**:



Viewing Reports in a Desktop Application

You can bypass browser viewing entirely and use a desktop application (such as Microsoft Excel) as your report viewer instead. To do this, define a subscription that specifies a desktop application format and a shared folder destination. The report server generates your report as an application file, appends a file name extension, and saves the report as a file on your hard disk. You can then use Microsoft Excel (or another application) instead of a browser to view your report.

About User Sessions

Reporting Services uses browser sessions to maintain consistency while viewing reports. Sessions are based on browser connections, not authenticated users. A new session is created every time that a user opens a report in a new browser window. Once a browser session is established, you continue to work with the version of the report that was opened when the session began, even if the report is modified on the report server. For example, if you open a report at 11:00 P.M., and a report author republishes the same report at 11:01 P.M., your session will contain the version that you opened for the duration of the session.

If you refresh a report within the same session using the browser's **Refresh** button, the original session version of the report is displayed. If you refresh an on-demand report using the **Refresh** button on the report toolbar, the report is re-run and new data, if any, is displayed.

Session information is stored in the report server temporary database. The report server does not use ASP.NET session management. If you restart the server or perform a database recovery operation, session state is not restored. For more information about session management, see Identifying Execution State.

In This Section

The following topics provide additional information about viewing and managing reports.

Find, view, and manage reports

Finding and Viewing Reports with a Browser (Report Builder and SSRS)
Describes how to use a URL to find and view a report.

Searching for Reports and Other Items (Report Builder and SSRS)
Describes how to use the search functionality in Report Manager to locate items on the report server.

Using My Reports (Report Builder and SSRS)

Describes how to use the My Reports folder as the personal workspace to store and work with reports that you own.

Previewing Reports in Report Builder
Describes how to preview reports while you create or update them.

See Also

Saving Reports (Report Builder)
Report Builder in SQL Server 2016
Install and Uninstall Report Builder

Finding and Viewing Reports in the web portal (Report Builder and SSRS)

3/24/2017 • 4 min to read • Edit Online

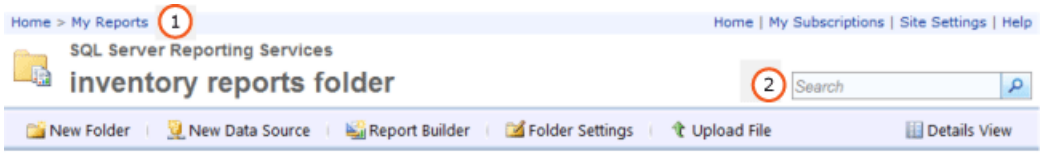
Report Manager is a Web-based tool that includes features for viewing and managing reports. It is part of a report server installation. To open Report Manager, type the Report Manager URL in a browser window. For information on browser requirements, see Browser Support for Reporting Services and Power View. For more information about how a Report Manager URL might be configured on your report server, contact your system administrator. For more information, see Configure Report Manager (Native Mode).

The permissions that the system administrator set on the report server determine what you can see when you use Report Manager. Permissions are granted via a role assignment. To find and view reports, your role assignment must include the View Reports task. To find a report on a report server, search for it by name or description, or browse report server folders. You can only search or browse for reports that have been published or uploaded to the report server. For more information about how to search for a report, see Searching for Reports and Other Items (Report Builder and SSRS).

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Navigating the Folder Hierarchy in Report Manager

To browse for the reports that you want to run, you can use the Home page, which appears automatically when you start Report Manager and when you open any folder in the folder hierarchy. The Home page shows only the items that you have permission to view. The folder path is displayed as a row of links at the top of the Home page. Folder names are listed in sequence, starting with the root folder (Home). As you open each additional folder, the folder name is added to the folder path at the top of the page. **(1)** in the image below. When you open a report, the name of the report is also added to the folder path.



Report Manager Ribbon

Use the following techniques to navigate through a folder hierarchy:

- To view the contents of a folder, click the folder name on the Home page. A folder page opens, displaying the contents of the folder.
- To navigate down through the folder hierarchy, open a subfolder of the current folder. Folders contain reports, resources, shared data source items, and other folders. Clicking a folder icon opens the folder, showing the contents of the hierarchy one level down.
- To navigate up through the folder hierarchy, in the row of links at the top of the page, click the name of the folder whose contents you want to see. **(1)** in the above image.

Opening a Report

After you find a report, click the report name to open it. The report is rendered in HTML and appears in the Contents page in Report Manager. Reports are always cached by the browser session, so if you open a report, you can usually return to it by clicking the **Back** button. This is true even if you were required to supply a user name and password to run the report. You cannot fully close a rendered report until you close the browser.

Not all reports that are visible in the folder hierarchy are immediately accessible. Some reports may prompt you for your user name and password to determine whether you can access the data source for the report. For more information about opening reports in Report Manager, see Open and Close a Report (Report Manager).

You can also browse to and open a report from the report server directly from Report Builder. For more information, see Searching for Reports and Other Items (Report Builder and SSRS).

To Search for a Items

- To search for items in Report Manager, type a search string in the **Search** text box at the top of the page. **(2)** in the above image. Searches begin at the top node in the folder hierarchy and then proceed through every branch. If you do not have permission to access a specific branch, that branch is skipped. This applies to My Reports folders that belong to other users, and to other folders that are not generally available. Only reports and items that you have permission to view are included in the search results.
- To search for an item by name or description, specify all or part of the text that you want to match. The search string is not case-sensitive. You cannot use search operators such as plus (+) or minus (–) symbols to require or exclude search criteria.
- To search for specific text within a report, use the toolbar at the top of the report.

See Also

Searching for Reports and Other Items (Report Builder and SSRS)
Finding, Viewing, and Managing Reports (Report Builder and SSRS)

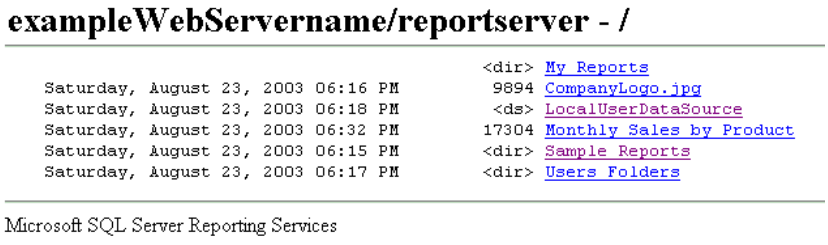
Finding and Viewing Reports with a Browser (Report Builder and SSRS)

3/24/2017 • 3 min to read • Edit Online

You can use any supported Web browser to view a report through a direct connection to a report server. Every report has a URL address on a report server. You can enter the Web address of a report to open it in a browser window independently of a Web application. The report opens in HTML format and includes the report toolbar so that you can navigate pages or search on data values within the report. You can set parameters on the URL to hide the toolbar or select the output format of the report.

Opening a report through its Web address is suitable for viewing a report, but not managing a report. You cannot access an item's property pages or subscription definition pages. You must use Report Manager or a SharePoint site for those tasks.

If you do not know the Web address of a report, you can open the Web address of the report server and then browse the report server folder hierarchy to select the report you want to view. The following diagram illustrates a folder hierarchy as it appears in a browser window.



Folders in a browser

> **NOTE**
> If you are accessing a report from a handheld device, you must use a browser to open a report. Report Manager is not scaled for handheld devices.

For more information about types of browsers that you can use, see "Browser Types Supported by Reporting Services" in the Reporting Services documentation in SQL Server Books Online.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Navigating Report Server Folders in a Web Browser

You can use a Web browser to navigate report server folders and run reports. Reports and items are displayed as links in the folder hierarchy. You can click links to open a report, resource, or folder, or view the contents of a shared data source. Navigating the folder hierarchy is useful if you do not know the URL of a report. You can specify the report server Web address to open a browser connection at the root node of the folder hierarchy, and then click folder links to navigate through the hierarchy.

When you access a report server virtual directory, you see only the folders, reports, and uploaded items to which you have access. The user interface shows only the folder hierarchy and basic information, such as creation or modification date, file size, and item type for individual items:

- A link with no other indicator is a report or a model.
- The tag <ds> indicates a shared data source.
- The tag <dir> indicates a folder item.
- A file name extension indicates a resource. The file name extension identifies the MIME type of the resource. For example, .jpg indicates an image in JPEG format.

Typing the URL Address of a Report

Reporting Services supports URL access to specific items on a report server. The URL must include a fully qualified path to the report and commands to render the report. If the report includes parameters, you must also specify any values that are required to open the report. If you are typing a URL for a report that includes spaces in the path, parameter values, or a rendering extension, you must incorporate URL encoded characters into the URL to get the results you expect. The following example illustrates a report URL that includes encoding for spaces in the

path name, parameters, and a rendering extension:

```
http://<Webservername>/reportserver?/<reportfolder>/employee+sales+summary&ReportYear=2004&ReportMonth=06&EmpID=24&rs:Command=Render&rs:Format=HTML4.0
```

The maximum limit for a URL in Internet Explorer is 2,083 characters. For more information, see Maximum URL length in Internet Explorer.

For more information about how to access a report through a URL, including information on how a URL is constructed, see "URL Access" in the Reporting Services documentation in SQL Server Books Online.

Searching for Reports and Other Items (Report Builder and SSRS)

3/24/2017 • 1 min to read • Edit Online

You can use Report Manager to search a report server for specific items by name or description. You can search for published reports, report models, folders, shared data sources, and resources. You cannot search for schedules, owners, role assignments, specific snapshots in report history, or subscriptions. The search is performed on the report server database where the items are stored.

> **NOTE**
> Performing a file system search will not return search results for items managed by a report server.

- To search for items in Report Manager, type a search string in the **Search for** text box at the top of the page. Searches begin at the top node in the folder hierarchy and then proceed through every branch. If you do not have permission to access a specific branch, that branch is skipped. This applies to My Reports folders that belong to other users, and to other folders that are not generally available. Only reports and items that you have permission to view are included in the search results.
- To search for an item by name or description, specify all or part of the text that you want to match. The search string is not case-sensitive. You cannot use search operators such as plus (+) or minus (–) symbols to require or exclude search criteria.
- To search for specific text within a report, use the toolbar at the top of the report.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

See Also

Finding and Viewing Reports in Report Manager (Report Builder and SSRS)
Using My Reports (Report Builder and SSRS)
Finding, Viewing, and Managing Reports (Report Builder and SSRS)
Open and Close a Report (Report Manager)

Using My Reports (Report Builder and SSRS)

3/24/2017 • 2 min to read • Edit Online

On a report server configured in native mode, the My Reports folder is a personal workspace that you can use to store and work with reports that you own. Other report server folders are public and typically require users to have advanced permissions to add to or modify folder contents. In contrast, the My Reports folder is a user-managed workspace. You can add or remove reports and folders and save linked reports with personalized settings.

Conceptually, the My Reports folder is similar to the My Documents folder in the Windows file system. Although each user has a folder called My Reports, the folder that each accesses is different from all other users. Except for report server administrators, other users cannot access the contents of the My Reports folder that belongs to you.

The My Reports feature is optional and can be disabled by a report server administrator. If it is enabled, you will see a My Reports folder in the Home folder, which you can access using the Report Manager or a Web browser. For more information, see Finding and Viewing Reports in Report Manager (Report Builder and SSRS).

On a report server configured in SharePoint integrated mode, there is no equivalent to the My Reports folder. For more information, see Finding, Viewing, and Managing Reports (Report Builder and SSRS).

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Ways to Use My Reports

My Reports is empty until you add reports, folders, and other items. Here are some ways to add content to My Reports.

- Create a personal linked report and store it in My Reports. Not all reports are available for linking. For more information, see Create a Linked Report.
- Upload a report definition (.rdl) file, report model (.smdl) file, or other files from the file system. You can upload any file, but the report server only processes report files that have an .rdl or .smdl file extension. For more information, see Report Definitions" in the Reporting Services documentation in SQL Server Books Online and Upload a File or Report (Report Manager).
- Create and publish your own reports to My Reports. For more information, see Report Design View (Report Builder).

  Usually, permissions on My Reports allow you to manage the folder yourself. However, the report server administrator ultimately determines which tasks users can perform. If insufficient permissions prevent you from working with My Reports, see your report server administrator.

Searching My Reports

When you search a report server database, the contents of your My Reports folder are included in the search, while the contents of other user's My Reports folders are excluded. Search results list only the reports to which you have access.

See Also

Finding, Viewing, and Managing Reports (Report Builder and SSRS)

Reporting Services Login Dialog Box (Report Builder)

3/24/2017 • 1 min to read • Edit Online

Use the **Reporting Services Login** dialog box to provide credentials to publish reports to the report server. For help with these credentials, contact your report server administrator.

Options

Server
Displays the name of the report server. For example, `http://localhost/reportserver`. For report servers that use a different port than default port 80, include the port number. For example, `http://localhost:8080/reportserver`.

User name
Type the user name to log in to the Web service.

Password
Type the password to log in to the Web service.

See Also

Report Builder Help for Dialog Boxes, Panes, and Wizards

Report Sections Dialog Box (Report Builder)

3/24/2017 • 1 min to read • Edit Online

When you open a report, Report Builder verifies that the report contains only one Report Definition Language (RDL) **<Section>** element. Opening a report with multiple **<Section>** elements is not supported.

Although a report definition with multiple sections can be generated programmatically, editing the report in a report authoring environment is not supported.

To identify the current report definition format for a report, for a report server, for a project, or for your report authoring environment, see Find the Report Definition Schema Version (SSRS).

Click **OK** to continue.

See Also

Report Builder Help for Dialog Boxes, Panes, and Wizards

Convert CRI Dialog Box (Report Builder)

3/24/2017 • 2 min to read • Edit Online

This report contains custom report items (CRIs) with unsupported features. CRIs are extensions to the Report Definition Language (RDL) that support custom objects that display data in a report. CRIs include design-time and run-time components that are supplied by third-party software vendors.

> **NOTE**
> Choosing to support custom report items on a report server is a decision made by the system administrator. To view CRIs in a report, the CRI components must be installed on the report authoring client to preview a report and on the report server to view a published or uploaded report.

Some CRIs can be converted to report items in the new report definition format. When you open the report, you are prompted whether to upgrade. Use the following information to decide whether to convert the CRIs in this report:

- **Yes** Choose **Yes** to convert all the CRIs in the report, where possible. Unsupported features in the CRIs cannot be upgraded and are removed from the report definition file. For the list of unsupported features, see Upgrade Reports. When you view the report, you might see differences in the way the CRI displays in the report.
- **No** Choose **No** when you do not want to convert the CRIs in the report. These CRIs cannot be displayed by the report processor in their current version. If your system administrator is planning to install a new version of the CRI from the third-party software vendor that is compatible with the new report definition format, you should choose **No**. Until new versions are available, the CRIs display in the report as an empty text box with a red X.

  In either case, the report is upgraded to the new report definition format and a backup copy of the original report is saved as <*Report Name*> - Backup.rdl. If you save the report in your report authoring tool, you are saving the upgraded report in the new report definition format. If you publish the report, the report is first saved on your computer, and then published to the report server. You are publishing the upgraded version of the report to the report server.

  If you do not save the report, the original report remains unchanged. However, you cannot edit this report in a SQL Server 2008 later version of SQL Server Data Tools or a report authoring environment that uses this report definition format. You can continue to run the original version of the report by uploading it to a SQL Server 2008 or later Reporting Services report server by using Report Manager. For more information, see Upload a File or Report (Report Manager).

  For reports that you upload instead of publish to a report server, the report processor determines whether the report can be upgraded on first use. Reports that cannot be upgraded are processed in backward-compatibility mode, and continue to display as they did in the earlier version of Reporting Services. For more information, see Upgrade Reports.

  To identify the current report definition format for a report, for a report server, or for your report authoring environment, see Find the Report Definition Schema Version (SSRS).

See Also

Report Builder Help for Dialog Boxes, Panes, and Wizards

Hide an Item (Report Builder and SSRS)

3/24/2017 • 2 min to read • Edit Online

Set the visibility of a report item when you want to conditionally hide an item based on a report parameter or some other expression that you specify.

You can also design a report to allow the user to toggle the visibility of report items based on clicking text boxes in the report, for example, for a drilldown report. For more information, see Add an Expand or Collapse Action to an Item (Report Builder and SSRS).

The following procedures describe how to show or hide a report item in a rendered report based on a constant or an expression.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

To hide a report item

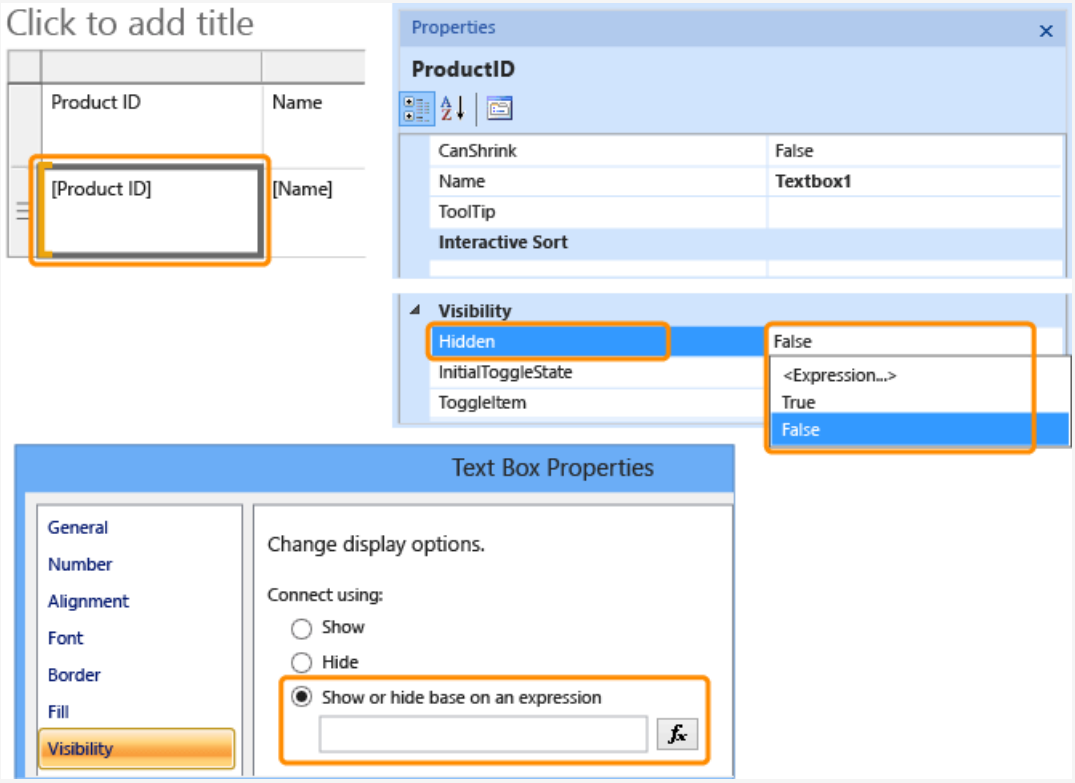
1. In report design view, right-click the report item and open its **Properties** page.

   > **NOTE**
   > To select an entire table or matrix data region, click in the data region to select it, right-click a row, column, or corner handle, and then click **Tablix Properties**.

2. Click **Visibility.**
3. In **When the report is initially run**, specify whether to hide the item when you first view the report:
   - To display the item, click **Show**.
   - To hide the item, click **Hide**.
   - To specify an expression that is evaluated at run-time, click **Show or hide based on an expression**. Type the expression or click the expression (**fx**) button to create the expression in the **Expression** dialog box.

NOTE

When you specify an expression for visibility, you are setting the Hidden property of the report item, as shown in the following image. The evaluated expression shows the report item when the value is False, and hides the report item when the value is True.



4. Click **OK** twice.

To hide static rows in a table, matrix, or list

1. In report design view, click the table, matrix, or list to display the row and column handles.
2. Right-click the row handle, and then click **Row Visibility**. The **Row Visibility** dialog box opens.
3. To set the visibility, follow steps 3 and 4 in the first procedure.

To hide static columns in a table, matrix, or list

1. In Design view, select the table, matrix, or list to display the row and column handles.
2. Right-click the column handle, and then click **Column Visibility**.
3. In the **Column Visibility** dialog box, follow steps 3 and 4 in the first procedure.

See Also

Drilldown Action (Report Builder and SSRS)
Add an Expand or Collapse Action to an Item (Report Builder and SSRS)
Expression Examples (Report Builder and SSRS)

Export Reports (Report Builder and SSRS)

3/24/2017 • 12 min to read • Edit Online

You can export a Reporting Services report to another file format, such as PowerPoint, Image, PDF, Microsoft Word, or Microsoft Excel or export the report by generating an Atom service document, listing the Atom-compliant data feeds available from the report. You can export your report from Report Builder, Report Designer (SQL Server Data Tools (SSDT)), or the report server.

Export a report to do the following:

- **Work with the report data in another application.** For example, you can export your report to Excel and then continue to work with the data in Excel.
- **Print the report in a different format.** For example, you can export the report to the PDF file format and then print it.
- **Save a copy of the report as another file type.** For example, you can export a report to Word and save it, creating a copy of the report.
- **Use report data as data feeds in applications.** For example, you can generate Atom-compliant data feeds that the SQL Server 2016 Power Pivot client can consume, and then work with the data in Power Pivot. For more information, see Generate Data Feeds from a Report (Report Builder and SSRS)
- Rendering the report on the report server is useful when you set up subscriptions or deliver your reports via e-mail, or if you want to save a report that is available on the report server. For more information, see Subscriptions and Delivery (Reporting Services).

  Reporting Services provides many rendering extensions, supporting exports of reports to common file formats. The rendering extensions support file formats with soft breaks (for example, Word or Excel), hard-page breaks (for example, PDF or TIFF), or data only (for example, CSV or Atom compliant XML).

  Report pagination might be affected when you export a report to a different format. When you preview a report, you are viewing the report as it is rendered by the HTML rendering extension, which follows soft-page break rules. When you export a report to a different file format, such as Adobe Acrobat (PDF), pagination is based on the physical page size, which follows hard-page break rules. Pages can also be separated by logical page breaks that you add to a report, but the actual length of a page varies based on the renderer type that you use. To change the pagination of your report, you must understand the pagination behavior of the rendering extension you choose. You might need to adjust the design of your report layout for this rendering extension. For more information see, Page Layout and Rendering (Report Builder and SSRS).

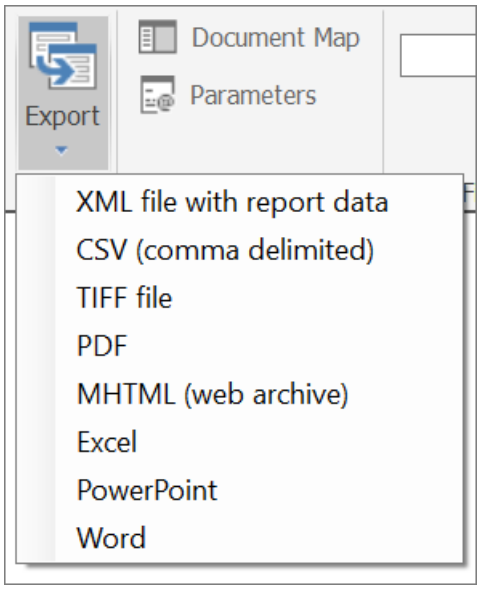
> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

In this topic

- To export a report from a SharePoint library
- Rendering Extension Types
- Formats you can export while viewing reports
- Generating Data Feeds From a Report
- Troubleshooting Exported Reports
- Other Ways of Exporting Reports

To export a report from Report Builder

1. Run or Preview the report.
2. On the ribbon, click **Export**.



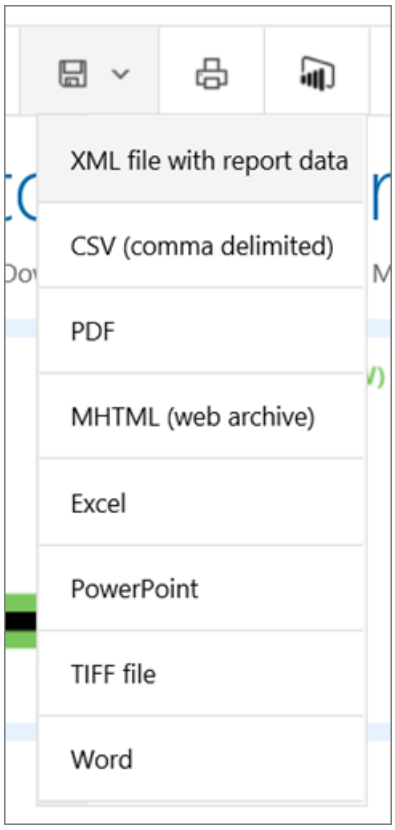
3. Select the format that you want to use.

   The **Save As** dialog opens. By default, the file name is that of the report that you exported. Optionally, you can change the file name.

In this topic

To export a report from the Reporting Services web portal

1. From the Reporting Services web portal **Home** page, navigate to the report that you want to export.
2. Click the report to render and preview the report.
3. On the Report Viewer toolbar, click the **Export** drop-down arrow.



4. Select the format that you want to use.
5. Click **Export**. A dialog appears asking you if you want to open or save the file.
6. To view the report in the selected export format, click **Open**.

   - or -

   To immediately save the report in the selected export format, click **Save**.

   Using the application that is associated with the format that you chose, the report is either displayed or saved. If you click **Save**, you will be prompted for a location where you can save your report.

In this topic

To export a report from a SharePoint library

1. Preview the report.
2. On the toolbar, click **Actions**, point to **Export**, and then select the format that you want to use.

   The **File Download** dialog box opens.
3. To view the report in the selected export format, click **Open**.

   - or -

   To immediately save the report in the selected export format, click **Save**.

   Using the application that is associated with the format that you chose, the report is either displayed or saved. If you click **Save**, you will be prompted for a location where you can save your report.

   Optionally, change the file name of the exported report.

   Note If the program cannot open the report in the format that you chose because you do not have a program associated with this file type, you will be prompted to save the exported report or to find a program online to open the report.

In this topic

Rendering Extension Types

There are three types of Reporting Services rendering extensions:

- **Data renderer extensions** Data rendering extensions strip all formatting and layout information from the report and display only the data. The resulting file can be used to import the raw report data into another file type, such as Excel, another database, an XML data message, or a custom application. Data renderers do not support page breaks.

  The following data rendering extensions are supported: CSV, XML, and Atom.

- **Soft page-break renderer extensions** Soft page-break rendering extensions maintain the report layout and formatting. The resulting file is optimized for screen-based viewing and delivery, such as on a Web page or in the **ReportViewer** controls.

  The following soft page-break rendering extensions are supported: Microsoft Excel, Microsoft Word, and Web archive (MHTML).

- **Hard page-break rendering extensions** Hard page-break renderer extensions maintain the report layout and formatting. The resulting file is optimized for a consistent printing experience, or to view the report online in a book format.

  The following hard page-break rendering extensions are supported: TIFF and PDF.

In this topic

Formats you can export while viewing reports

Reporting Services provides rendering extensions that render reports in different formats. You should optimize the report design for your chosen file format. The followin table lists the formats you can export from the user interface. There are additional formats you can use with Reporting Services subscriptions or if you are exporting from URL access. See the section Other Ways of Exporting Reportsin this topic.

FORMAT	RENDERING EXTENSION TYPE	DESCRIPTION
Acrobat (PDF) file	Hard page-break	The PDF rendering extension renders a report to files that can be opened in Adobe Acrobat and other third-party PDF viewers that support PDF 1.3. Although PDF 1.3 is compatible with Adobe Acrobat 4.0 and later, Reporting Services supports Adobe Acrobat 6 or later. The rendering extension does not require Adobe software to render the report. However, PDF viewers such as Adobe Acrobat are required for viewing or printing a report in PDF format. For more information, see Exporting to a PDF File (Report Builder and SSRS).

FORMAT	RENDERING EXTENSION TYPE	DESCRIPTION
Atom	Data	The Atom rendering extension generates Atom-compliant data feeds from reports. The data feeds are readable and exchangeable with applications such as the SQL Server 2016 Power Pivot client that can consume Atom-compliant data feeds. The output is an Atom service document that lists the data feeds available from a report. At least one data feed is created for each data region in a report. Depending on the type of data region and the data that the data region displays, multiple data feeds might be generated. For more information, see Generating Data Feeds from Reports (Report Builder and SSRS).
CSV	Data	The Comma-Separated Value (CSV) rendering extension renders reports as a flattened representation of data from a report in a standardized, plain-text format that is easily readable and exchangeable with many applications. For more information, see Exporting to a CSV File (Report Builder and SSRS).
EXCELOPENXML	Soft page-break	Displayed as "Excel" in the export menus when reviewing reports. The Excel rendering extension renders a report as an Excel document (.xlsx) that is compatible with Microsoft Excel 2013. For more information, see Exporting to Microsoft Excel (Report Builder and SSRS).
PowerPoint	Hard page-break	The PowerPoint rendering extension renders a report as an PowerPoint document (.pptx) that is compatible with PowerPoint 2013.

FORMAT	RENDERING EXTENSION TYPE	DESCRIPTION
TIFF file	Hard page-break	The Image rendering extension renders a report to a bitmap or metafile. By default, the Image rendering extension produces a TIFF file of the report, which can be viewed in multiple pages. When the client receives the image, it can be displayed in an image viewer and printed. The Image rendering extension can generate files in any of the formats supported by GDI+: BMP, EMF, EMFPlus, GIF, JPEG, PNG, and TIFF. For more information, see Exporting to an Image File (Report Builder and SSRS).
Web archive	Soft page-break	The HTML rendering extension renders a report in HTML format. The rendering extension can also produce fully formed HTML pages or fragments of HTML to embed in other HTML pages. All HTML is generated with UTF-8 encoding. The HTML rendering extension is the default rendering extension for reports that are previewed in Report Builder and viewed in a browser, including when run in the Reporting Services web portal. For more information, see Rendering to HTML (Report Builder and SSRS).
WORDOPENXML	Soft page-break	Displayed as "Word" in the export menu when viewing reports. The Word rendering extension renders a report as a Word document (.docx) that is compatible with Microsoft Word 2013. For more information, see Exporting to Microsoft Word (Report Builder and SSRS).
XML	Data	The XML rendering extension returns a report in XML format. The schema for the report XML is specific to the report, and contains data only. Layout information is not rendered and pagination is not maintained by the XML rendering extension. The XML generated by this extension can be imported into a database, used as an XML data message, or sent to a custom application. For more information, see Exporting to XML (Report Builder and SSRS).

In this topic

Generating Data Feeds From a Report

To generate data feeds from a report, run the report in the Reporting Services web portal, and then click the **Generate Data Feed** icon on the web portal toolbar. You are prompted to choose whether to save or open the file. If you chose **Open**, the Atom service document opens in the application that is associated with the .atomsvc file extension. If you chose **Save**, the document is saved as an .atomsvc file. By default, the name of the file is the name of the report. You can change the name to one that is more meaningful.

You save the Atom service document to your computer. Later you can upload it to a report server or another server to make it available for others to use. For more information, see Generating Data Feeds from Reports (Report Builder and SSRS) and Generate Data Feeds from a Report (Report Builder and SSRS).

In this topic

Troubleshooting Exported Reports

Sometimes your reports look different or do not work the way you want after you export them to a different format. This occurs because certain rules and limitations might apply to the renderer. You can address many limitations by considering them when you create the report. You might need to use a slightly different layout in your report, carefully align items within the report, confine report footers to a single line of text, and so forth.

If your report contains Unicode text with Arabic numbers, or contains dates in Arabic, the dates and numbers do not render correctly when you export the report to any of the following formats or print the report.

- PDF
- Word
- Excel
- Image/TIFF

  If you export the report to HTML, the dates and numbers render correctly.

  The topics about specific renderers describe how report items and data regions are rendered as well as the limitations and solutions for each renderer.

- Exporting to a CSV File (Report Builder and SSRS)
- Exporting to Microsoft Excel (Report Builder and SSRS)
- Exporting to Microsoft Word (Report Builder and SSRS)
- Rendering to HTML (Report Builder and SSRS)
- Exporting to a PDF File (Report Builder and SSRS)
- Exporting to an Image File (Report Builder and SSRS)
- Exporting to XML (Report Builder and SSRS)
- Generating Data Feeds from Reports (Report Builder and SSRS)

  Reporting Services provides additional features to help you create reports that work well in other formats. Page breaks on tablix data regions (table, matrix, and list), groups, and rectangles give you better control of report pagination. Report pages, delimited by page breaks, can have different page names and reset page numbering. By using expressions, the page names and page numbers can be dynamically updated when the report is run. For more information, see Pagination in Reporting Services (Report Builder and SSRS).

  In addition, you can use the RenderFormat built-in global to conditionally apply different report layouts for different renderers. For more information, see Built-in Globals and Users References (Report Builder and

SSRS).

In this topic

Other Ways of Exporting Reports

Exporting a report is an on-demand task that you perform when the report is open in the Reporting Services web portal or Report Builder. If you want to automate an export operation (for example, to export a report to a shared folder as a specific file type on a recurring schedule), create a subscription that delivers the report to a shared folder. For more information, see File Share Delivery in Reporting Services.

Reports previewed in the reporting tools or opened in a browser application such as the Reporting Services web portal are always first rendered in HTML. You cannot specify a different rendering extension as the default for viewing. You can, however, create a subscription that produces a report in the rendering format you want for subsequent delivery to an e-mail inbox or shared folder. For more information, see Create and Manage Subscriptions for Native Mode Report Servers and Create, Modify, and Delete Data-Driven Subscriptions.

You can also access a report through a URL that specifies a rendering extension as a URL parameter and render the report directly to the specified format without rendering it in HTML first. The following example renders a report in Excel format:

```
http://<Report Server Name>/reportserver?/Sales/YearlySalesSummary&rs:Format=Excel&rs:Command=Render
```

and the following renders a PowerPoint report from a named instance:

```
http://<Report Server Name/ReportServer_THESQLINSTANCE/Pages/ReportViewer.aspx?
%2freportfolder%2freport+name+with+spaces&rs:Format=pptx
```

For more information, see Export a Report Using URL Access.

In this topic

See Also

Controlling Page Breaks, Headings, Columns, and Rows (Report Builder and SSRS)
Finding, Viewing, and Managing Reports (Report Builder and SSRS)
Print Reports (Report Builder and SSRS)
Saving Reports (Report Builder)

Need help? MSDN Forum, Stackoverflow, Connect

Interactive Functionality - Different Report Rendering Extensions

3/29/2017 • 5 min to read • Edit Online

Reporting Services provides features for interacting with a paginated report at run time. Not all of the report rendering formats support the full range of interactive features. Use the following table to understand how each interactive feature works in specific formats.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Interactive Features in Different Output Formats

Paginated reports that are rendered to XML, CSV, or Image formats do not support interactive features.

Reports that you view in the Reporting Services web portal, SharePoint Web parts, or in a browser are rendered in HTML. HTML and Windows Forms are the only report output formats that fully support all interactive features. Alternate HTML formats (such as MHTML) support many interactive features. If exceptions for MHTML exist, they are noted in the following table.

Document maps

EXPORT OPTION	SUPPORT INFORMATION
Preview/Report Viewer, HTML	Interactive document map provides a navigation pane of hierarchical links that can be used to navigate to different sections of a report.
PDF	PDF renders a document map as the Bookmarks pane. All items in the document map are listed one after the other down the pane. It includes a hierarchy of links. If a page range is specified, only those bookmarks that are rendered exist in the hierarchy.
Excel	Excel renders a document map as the first worksheet in the workbook. It includes a hierarchy of links. When the link in the document map is clicked, the appropriate target cell in the respective worksheet is opened.
Word	Word renders the document map as Table of Contents labels.
Other (for example, TIFF, XML, and CSV)	Not available in MHTML, XML, CSV, or Image.

Drillthrough links to other reports

EXPORT OPTION	SUPPORT INFORMATION
Preview/Report Viewer, HTML	Users click on data values in the report to view related data in another report.

EXPORT OPTION	SUPPORT INFORMATION
PDF	Drillthrough links are not available in PDF. Consider using hyperlinks for PDF reports that link to other pages.
Excel	Drillthrough links are rendered in Excel. The link becomes a hyperlink pointing to the report referenced by the drillthrough link. Clicking the link opens a report in a browser window.
Word	Drillthrough links are rendered in Word. The link becomes a hyperlink pointing to the report referenced by the drillthrough link. Clicking the link opens a report in a browser window.
Other	Not available in XML, CSV, or Image.

Toggle items within a report

EXPORT OPTION	SUPPORT INFORMATION
Preview/Report Viewer, HTML	Users click expand and collapse icons to view sections of a report.
PDF	The report server exports the current show or hide state of the report to PDF. Interactive toggling is not supported
Excel	Drilldown links and items that can be toggled are rendered as collapsible outlines in Excel. You can expand and collapse sections of the report in Excel. For more information about Excel-imposed limitations, see Exporting to Microsoft Excel (Report Builder and SSRS).
Word	The report server exports the current show or hide state of the report to PDF. Interactive toggling is not supported
Other	Not available in MHTML, XML, or CSV. When exporting to an Image format, the report server exports the current show or hide stated of the report to PDF. Interactive toggling is not supported.

Interactive sorting

EXPORT OPTION	SUPPORT INFORMATION
Preview/Report Viewer, HTML	For tabular repots, users click sort arrows on column to change how the data is sorted.
PDF	Not available in PDF.
Excel	Not available in Excel.
Word	Not available in Word.

EXPORT OPTION	SUPPORT INFORMATION
Other	Not available in MHTML, XML, CSV, or Image.

Hyperlinks to external Web content or images

EXPORT OPTION	SUPPORT INFORMATION
Preview/Report Viewer, HTML	Users click links to open external Web pages in a new browser window.
PDF	Hyperlinks are rendered by the PDF rendering extension. When a user clicks on a hyperlink, the linked pages are opened in the browser.
Excel	Hyperlinks are rendered in Excel.
Word	Hyperlinks are rendered in Word.
Other	Hyperlinks are not available in MHTML, XML, CSV, or Image. For MHTML, and Image, external images are rendered as a static picture.

Bookmark or anchor

EXPORT OPTION	SUPPORT INFORMATION
Preview/Report Viewer, HTML	Users click links to navigate to another section of the same report.
PDF	Not available in PDF.
Excel	Bookmarks are rendered in Excel. The bookmark becomes a hyperlink pointing to the name of the report item.
Word	Bookmarks are rendered in Word. The bookmark becomes a hyperlink pointing to the bookmarked report item. Only 40 characters of a bookmark or anchor name is converted when the report is exported which can lead to duplicate bookmark or anchor names. Spaces are converted to underscores (_).
Other	Not available in XML, CSV, or Image.

Prompted parameters obtained at run time

EXPORT OPTION	SUPPORT INFORMATION
Preview/Report Viewer, HTML	A parameter input area appears at the top of the report. This area is part of the HTML Viewer used to display reports in a browser.

EXPORT OPTION	SUPPORT INFORMATION
PDF	The report server exports the report to PDF using the parameter values currently in effect for the report.
Excel	The report server exports the report to Excel using the parameter values currently in effect for the report.
Word	The report server exports the report to Word using the parameter values currently in effect for the report.
Other	The report server exports the report to other formats using the parameter values currently in effect for the report.

Filters applied at run time

EXPORT OPTION	SUPPORT INFORMATION
Preview/Report Viewer, HTML	The filtered data is displayed to the user at run time.
PDF	The report server exports the report to PDF using filtered data in the current report.
Excel	The report server exports the report to Excel using filtered data in the current report.
Word	The report server exports the report to Word using filtered data in the current report.
Other	The report server exports the report to other formats using filtered data in the current report.

See Also

Export Reports (Report Builder and SSRS)
Interactive Sort, Document Maps, and Links (Report Builder and SSRS)

Tables, Matrices, and Lists (Report Builder and SSRS)
Charts (Report Builder and SSRS)

Exporting to a CSV File (Report Builder and SSRS)

3/24/2017 • 5 min to read • Edit Online

The Comma-Separated Value (CSV) rendering extension renders paginated reports as a flattened representation of data from a report in a standardized, plain-text format that is easily readable and exchangeable with many applications.

The CSV rendering extension uses a string character delimiter to separate fields and rows, with the string character delimiter configurable to be a character other than a comma. The resulting file can be opened in a spreadsheet program like Microsoft Excel or used as an import format for other programs. The exported report becomes a .csv file, and returns a MIME type of **text/csv**.

If you want to work with data related to charts, data bars, sparklines, gauges, and indicators in Microsoft Excel, export the report to a CSV file, and then open the file in Microsoft Excel.

See Export Reports (Report Builder and SSRS) for details on how to export to CSV format.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

CSV Rendering

When rendered using the default settings, a CSV report has the following characteristics:

- The default field delimiter string is a comma (,).

  > **NOTE**
  > You can change the field delimiter to any character that you want, including TAB, by changing the device information settings. For more information, see CSV Device Information Settings.

- The record delimiter string is the carriage return and line feed (<cr><lf>).
- The text qualifier string is a quotation mark (").

  The CSV renderer does not add qualifiers around all text strings. Text qualifiers are added only when the value contains the delimiter character or when the value has a line break.

- If the text contains an embedded delimiter string or qualifier string, the text qualifier is placed around the text, and the embedded qualifier strings are doubled.
- Formatting and layout are ignored.

  The following items are ignored during rendering:

- Page header
- Page footer
- Custom report items
- Line

- Image
- Rectangle
- Automatic subtotals

The remaining report items are sorted, from top to bottom, then left to right. Each item is then rendered to a column. If the report has nested data items like lists or tables, the parent items are repeated in each record.

The following table indicates the appearance of report items when rendered:

ITEM	RENDERING BEHAVIOR
Text box	Renders the contents of the text box. In default mode, items are formatted based on the item's formatting properties. In compliant mode, formatting can be changed by device information settings. For more information about CSV rendering modes, see below.
Table	Renders by expanding the table and creating a row and column for each row and column at the lowest level of detail. Subtotal rows and columns do not have column or row headings. Drillthrough reports are not supported.
Matrix	Renders by expanding the matrix and creating a row and column for each row and column at the lowest level of detail. Subtotal rows and columns do not have column or row headings.
List	Renders a record for each detail row or instance in the list.
Subreport	The parent item is repeated for each instance of the contents.
Chart	Renders by creating a row for each chart value and member labels. Labels from series and categories in hierarchies are flattened and included in the row for a chart value.
Data bar	Renders like a chart. Typically, a data bar does not include hierarchies or labels.
Sparkline	Renders like a chart. Typically, a sparkline does not do not include hierarchies or labels.
Gauge	Renders as a single record with the minimum and maximum values of the linear scale, start and end values of the range, and the value of the pointer.
Indicator	Renders as a single record with the active state name, available states, and the data value.
Map	Renders a row with the labels and values for each map member of a map layer. If the map has multiple layers the values in the rows varies depending on whether the map layers use the same or different map data regions. If multiple map layers use the same data region, the rows contain data from all layers.

Hierarchical and Grouped Data

Hierarchical and grouped data must be flattened in order to be represented in the CSV format.

The rendering extension flattens the report into a tree structure that represents the nested groups within the data region. To flatten the report:

- A row hierarchy is flattened before a column hierarchy.
- Columns are ordered as follows: text boxes in body order left-to-right, top-to-bottom followed by data regions ordered left-to-right, top-to-bottom.
- Within a data region, the columns are ordered as follows: corner members, row hierarchy members, column hierarchy members, and then cells.
- Peer data regions are data regions or dynamic groups that share a common data region or dynamic ancestor. Peer data is identified by branching of the flattened tree.

  For more information, see Tables, Matrices, and Lists (Report Builder and SSRS).

Renderer Modes

The CSV rendering extension can operate in two modes: one is optimized for Excel and the other is optimized for third-party applications that require strict CSV compliance to the CSV specification in RFC 4180. Depending on which mode you use, peer data regions are handled differently.

Default Mode

The default mode is optimized for Excel. When rendered in default mode, the report is rendered as a CSV file with multiple sections of CSV-rendered data. Each peer data region is delimited by an empty line. Peer data regions within the report body are rendered as separate blocks of data within the CSV file. The result is a CSV file in which:

- Individual text boxes within the report body are rendered once as the first block of data within the CSV file.
- Each top-level peer data region in the report body is rendered in its own data block.
- Nested data regions are rendered diagonally into the same data block.

Formatting

Numeric values are rendered in their formatted state. Excel can recognize formatted numeric values, such as currency, percentage and date, and format the cells appropriately when importing the CSV file.

Compliant Mode

Compliant mode is optimized for third-party applications.

Data Regions

Only the first row of the file contains the column headers and each row has the same number of columns.

Formatting

Values are unformatted.

Interactivity

Interactivity is not supported by either CSV formats generated by this renderer. The following interactive elements are not rendered:

- Hyperlinks
- Show or Hide
- Document Map
- Drillthrough or clickthrough links

- End user sort
- Fixes headers
- Bookmarks

Device Information Settings

You can change some default settings for this renderer, including which mode to render in, which characters to use as delimiters and which characters to use as the text qualifier default string, by changing the device information settings. For more information, see CSV Device Information Settings.

See Also

Pagination in Reporting Services (Report Builder and SSRS)
Rendering Behaviors (Report Builder and SSRS)
Interactive Functionality for Different Report Rendering Extensions (Report Builder and SSRS)
Rendering Report Items (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)

Exporting to Microsoft Excel (Report Builder and SSRS)

3/24/2017 • 18 min to read • Edit Online

The Reporting Services Excel rendering extension renders a Reporting Services paginated report to the Microsoft Excel format (.xlsx). With the Excel rendering extension, the width of columns in Excel more accurately reflects the width of columns in reports.

The format is Office Open XML. The content type of files generated by this renderer is **application/vnd.openxmlformats-officedocument.spreadsheetml.sheet** and the file extension is .xlsx.

You can change some default settings for this renderer by changing the device information settings. For more information, see Excel Device Information Settings.

See Export Reports (Report Builder and SSRS) for details on how to export to Excel.

> **IMPORTANT**
> When you define a parameter of type **String**, the user is presented with a text box that can take any value. If a report parameter is not tied to a query parameter and the parameter values are included in the report, it is possible for a report user to type expression syntax, script, or a URL into the parameter value, and render the report to Excel. If another user then views the report and clicks the rendered parameter contents, the user may inadvertently execute the malicious script or link.
>
> To mitigate the risk of inadvertently running malicious scripts, open rendered reports only from trusted sources. For more information about securing reports, see Secure Reports and Resources.

Excel Limitations

Microsoft Excel places limitations on exported reports due to the capabilities of Excel and its file formats. The most significant are the following:

- The maximum column width is limited to 255 characters or 1726.5 points. The renderer does not verify that the column width is less than the limit.
- The maximum number of characters in a cell is limited to 32,767. If this is exceeded, the renderer displays an error message.
- The maximum row height is 409 points. If the contents of the row cause the row height to increase beyond 409 points, the Excel cell shows a partial amount of text up to 409 points. The rest of the cell contents is still within the cell (up to Excel's max number of characters of 32,767).
- Because the maximum row height is 409 points, if the defined height of the cell in the report is something larger than 409 points, Excel splits the cell contents into multiple rows.
- The maximum number of worksheets is not defined in Excel, but external factors, such as memory and disk space, might causes limitations to be applied.
- In outlines, Excel permits up to seven nested levels only.
- If the report item that controls whether another item is toggled is not in the previous or next row or column of the item being toggled, the outline is disabled also.

  For more details about Excel limitations, see Excel specifications and limits.

Sizes of Excel 2003 (.xls) Files

> **IMPORTANT**
> The Microsoft Excel 2003 rendering extension is deprecated. For more information, see Deprecated Features in SQL Server Reporting Services in SQL Server 2016.

When reports are first exported and saved to Excel 2003, they do not benefit from the file optimization that Excel automatically applies to its *.xls workbook files. The larger file size can cause problems for e-mail subscriptions and attachments. To reduce the size of the* \.xls files for exported reports, open the *.xls files and then resave the workbooks. Resaving the workbooks typically reduces their file sizes by 40 to 50 percent.

> **NOTE**
> In Excel 2003, approximately 1000 characters are displayed in an Excel cell on the worksheet but up to the maximum number of characters can be edited in the formula bar. This limitation does not apply to current (.xlsx) Excel files.

Text Boxes and Text

The following limitations apply to text boxes and text:

- Text box values that are expressions are not converted to Excel formulas. The value of each text box is evaluated during report processing. The evaluated expression is exported as the contents of each Excel cell.
- Text boxes are rendered within one Excel cell. Font size, font face, decoration, and font style are the only formatting that is supported on individual text within an Excel cell.
- The text effect "Overline" is not supported in Excel.
- Excel adds a default padding of approximately 3.75 points to the left and right sides of cells. If a text box's padding settings are less than 3.75 points and is just barely wide enough to accommodate the text, the text may wrap in Excel.

  > **NOTE**
  > To work around this issue, increase the width of the text box in the report.

Images

The following limitations apply to images:

- Background images for report items are ignored because Excel does not support background images for individual cells.
- The Excel rendering extension only supports the background image of the report body. If a report body background image is displayed in the report, the image is rendered as a worksheet background image.

Rectangles

The following limitation apply to rectangles.

- Rectangles in report footers are not exported to Excel. However, rectangles in the report body, tablix cells, and so forth are rendered as a range of Excel cells.

Report Headers and Footers

The following limitations apply to report headers and footers:

- Excel headers and footers support a maximum of 256 characters including markup. The rendering extension truncates the string at 256 characters.
- Reporting Services does not support margins on report headers and footers. When exported to Excel, these margin values are set to zero and any header or footer that contains multiple rows of data might not print multiple rows, depending on the printer settings.
- Text boxes in a header or footer maintain their formatting but not their alignment when exported to Excel. This occurs because leading and trailing spaces are trimmed when the report is rendered to Excel.

Merging Cells

The following limitation applies to merging cells:

- If cells are merged, word-wrap does not work correctly. If any merged cells exist on a row where a text box is rendered with the AutoSize property, autosize will not work.

  The Excel renderer is primarily a layout renderer. Its goal is to replicate the layout of the rendered report as closely as possibly in an Excel worksheet and consequently cells might be merged in the worksheet to preserve the report layout.

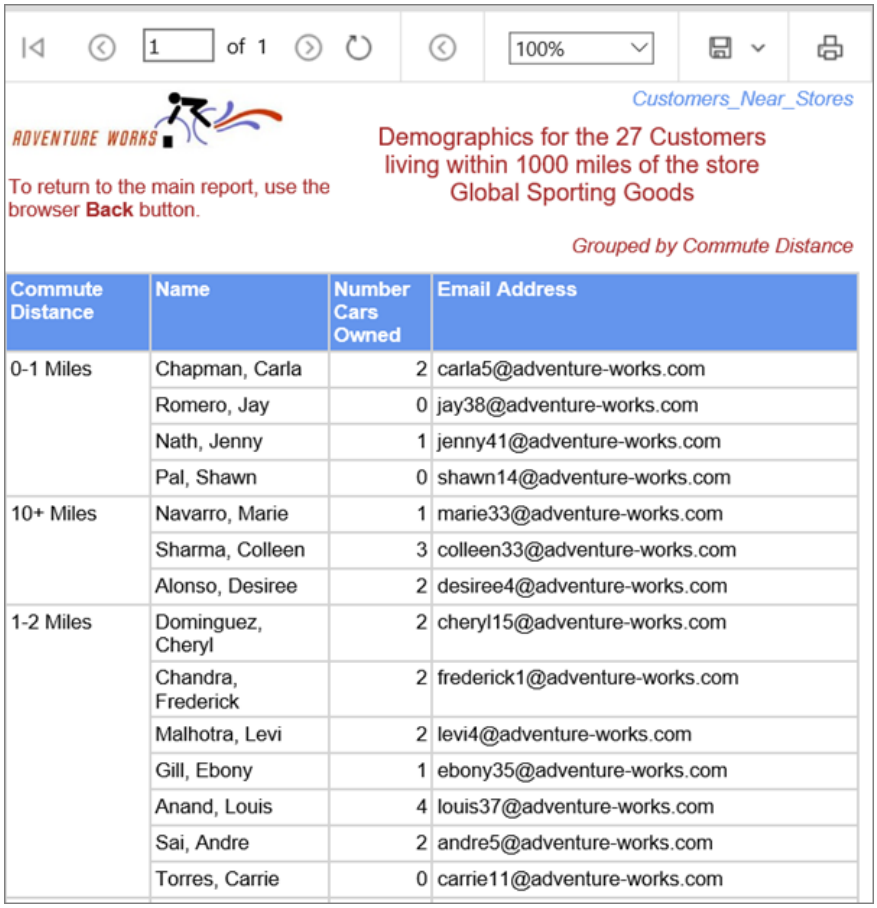
Merged cells can cause problems because the sort functionality in Excel requires cells to be merged in a very specific way for sort to work properly. For example, Excel requires that the ranges of merged cells have the same size in order to be sorted.

If it is important that reports exported to Excel worksheets can be sorted, then the following can help you reduce the number of merged cells in your Excel worksheets, which is the common cause for difficulties with Excel sort functionality.

- Not aligning items left and right is the most common cause of merged cells. Make sure the left and right edges of all report items line up with one another. Making items align and the same width will solve the problem in the majority of cases.
- Although you align all items precisely, you might find in some rare cases that some columns continue to be merged. This could be caused by internal unit conversion and rounding when the Excel worksheet is rendered. In the report definition language (RDL), you can specify position and size in different measurement units such as inches, pixels, centimeters, and points. Internally the Excel uses points. To minimize conversion and the potential inaccuracy of rounding when converting inches and centimeters to points, consider specifying all measurements in whole points for the most direct results. One inch is 72 points.

Report Row Groups and Column Groups

Reports that include row groups or column groups contain empty cells when exported to Excel. Imagine a report that groups rows on commute distance. Each commute distance can contain more than one customer. The following picture shows the report.



Commute Distance	Name	Number Cars Owned	Email Address
0-1 Miles	Chapman, Carla	2	carla5@adventure-works.com
	Romero, Jay	0	jay38@adventure-works.com
	Nath, Jenny	1	jenny41@adventure-works.com
	Pal, Shawn	0	shawn14@adventure-works.com
10+ Miles	Navarro, Marie	1	marie33@adventure-works.com
	Sharma, Colleen	3	colleen33@adventure-works.com
	Alonso, Desiree	2	desiree4@adventure-works.com
1-2 Miles	Dominguez, Cheryl	2	cheryl15@adventure-works.com
	Chandra, Frederick	2	frederick1@adventure-works.com
	Malhotra, Levi	2	levi4@adventure-works.com
	Gill, Ebony	1	ebony35@adventure-works.com
	Anand, Louis	4	louis37@adventure-works.com
	Sai, Andre	2	andre5@adventure-works.com
	Torres, Carrie	0	carrie11@adventure-works.com

When the report is exported to Excel, the commute distance appears only in one cell of the Commute Distance column. Depending on the alignment of the text in the report (top, middle, or bottom) the value is in the first, middle, or last cell. The other cells are empty. The Name column that contains customer names has no empty cells. The following picture shows the report after it is exported to Excel. The red cell borders were added for emphasis. The gray boxes are the empty cells. (Neither the red lines nor the gray boxes are part of the exported report.)

	A	B	C	E	F	G	H	I
7	Commute Distance	Name	Distance in Miles			Number Cars		Email Address
8	0-1 Miles	Chapman, Carla	650			2		carla5@adventure-works.com
9		Romero, Jay	791			0		jay38@adventure-works.com
10		Nath, Jenny	773			1		jenny41@adventure-works.com
11		Pal, Shawn	953			0		shawn14@adventure-works.com
12	10+ Miles	Navarro, Marie	91			1		marie33@adventure-works.com
13		Sharma, Colleen	243			3		colleen33@adventure-works.com
14		Alonso, Desiree	950			2		desiree4@adventure-works.com
15	1-2 Miles	Dominguez, Cheryl	884			2		cheryl15@adventure-works.com
16		Chandra, Frederick	956			2		frederick1@adventure-works.com
17		Malhotra, Levi	949			2		levi4@adventure-works.com
18		Gill, Ebony	587			1		ebony35@adventure-works.com
19		Anand, Louis	728			4		louis37@adventure-works.com
20		Sai, Andre	799			2		andre5@adventure-works.com
21		Torres, Carrie	1,000			0		carrie11@adventure-works.com

This means that reports with row groups or column groups require modification after exporting to Excel and before you can display the exported data in pivot table. You must add the group value to cells in which they are missing to make the worksheet a flat table with values in all cells. The following picture shows the updated worksheet.

	A	B	G	H	I
7	Commute Distance	Name	Number Cars		Email Address
8	0-1 Miles	Chapman, Carla	2		carla5@adventure-works.com
9	0-1 Miles	Romero, Jay	0		jay38@adventure-works.com
10	0-1 Miles	Nath, Jenny	1		jenny41@adventure-works.com
11	0-1 Miles	Pal, Shawn	0		shawn14@adventure-works.com
12	10+ Miles	Navarro, Marie	1		marie33@adventure-works.com
13	10+ Miles	Sharma, Colleen	3		colleen33@adventure-works.com
14	10+ Miles	Alonso, Desiree	2		desiree4@adventure-works.com
15	1-2 Miles	Dominguez, Cheryl	2		cheryl15@adventure-works.com
16	1-2 Miles	Chandra, Frederick	2		frederick1@adventure-works.com
17	1-2 Miles	Malhotra, Levi	2		levi4@adventure-works.com
18	1-2 Miles	Gill, Ebony	1		ebony35@adventure-works.com
19	1-2 Miles	Anand, Louis	4		louis37@adventure-works.com
20	1-2 Miles	Sai, Andre	2		andre5@adventure-works.com
21	1-2 Miles	Torres, Carrie	0		carrie11@adventure-works.com

So if you create a report for the specific purpose of exporting it to Excel for further analysis of the report data, consider not grouping on rows or columns in your report.

Excel Renderer

Current (.xlsx) Excel File Renderer

In SQL Server Reporting Services, the default Excel renderer is the version compatible with current (.xlsx) Microsoft Excel files. This is the **Excel** option on the **Exporting** menus in the Reporting Services web portal and SharePoint list.

When you use the default Excel renderer, instead of the earlier Excel 2003 (.xls) renderer, you can install the Microsoft Office Compatibility Pack for Word, Excel, and PowerPoint to allow earlier versions of Excel to open the files that are exported.

Excel 2003 (.xls) Renderer

> **IMPORTANT**
>
> The Microsoft Excel 2003 rendering extension is deprecated. For more information, see Deprecated Features in SQL Server Reporting Services in SQL Server 2016.

The earlier version of the Excel renderer, compatible with Excel 2003, is now named Excel 2003 and is listed on menus using that name. The content type of files generated by this renderer is **application/vnd.ms-excel** and the file name extension of files is .xls.

By default, the **Excel 2003** menu option is not visible. An administrator can make it visible under certain circumstances by updating the RSReportServer configuration file. To export reports from SQL Server Data Tools (SSDT) using the Excel 2003 renderer, you update the RSReportDesigner configuration file.

The **Excel 2003** menu option extension is never visible in the following scenarios:

- Report Builder in disconnected mode and you preview a report in Report Builder. Because the RSReportServer configuration file resides on the report server, the tools or products from where you export reports must be connected to

a report server to read the configuration file.

- Report Viewer Web Part in local mode and the SharePoint farm is not integrated with a Reporting Services report server. For more information, see Local Mode vs. Connected Mode Reports in the Report Viewer (Reporting Services in SharePoint Mode)

  If the **Excel 2003** menu option renderer is configured to be visible, both the Excel and Excel 2003 options are available in the following scenarios:

- Reporting Services web portal native mode.
- SharePoint site when Reporting Services is installed in SharePoint integrated mode.
- SQL Server Data Tools (SSDT) and you preview reports.
- Report Builder connected to a report server.
- The Report Viewer Web Part in remote mode.

  The following XML shows the elements for the two Excel rendering extensions in the RSReportServer and RSReportDesigner configuration files:

```
<Extension Name="EXCELOPENXML"
Type="Microsoft.ReportingServices.Rendering.ExcelOpenXmlRenderer.ExcelOpenXmlRenderer,Microsoft.ReportingServices.ExcelRendering"/>
```

```
<Extension Name="EXCEL"
Type="Microsoft.ReportingServices.Rendering.ExcelRenderer.ExcelRenderer,Microsoft.ReportingServices.ExcelRendering"
Visible="false"/>
```

  The EXCELOPENXML extension defines the Excel renderer for current (.xlsx) Excel files. The EXCEL extension defines the Excel 2003 version. `Visible = “false”` indicates the Excel 2003 renderer is hidden. For more information, see RsReportServer.config Configuration File and RSReportDesigner Configuration File.

Differences Between the current (.xlsx) Excel and Excel 2003 Renderers

Reports, rendered by using the current (.xlsx) Excel or the Excel 2003 renderers are typically identical and only under rare circumstances will you notice differences between the two formats. The following table compares the Excel and the Excel 2003 renderers.

PROPERTY	EXCEL 2003	CURRENT EXCEL
Maximum columns per worksheet	256	16,384
Maximum rows per worksheet	65,536	1,048,576
Number of colors allowed in a worksheet	56 (palette) If more than 56 colors are used in the report, the rendering extension matches the required color to one of the 56 colors already available in the custom palette.	Approximately 16 million (24-bit color)
ZIP compressed files	None	ZIP compression
Default font family	Arial	Calibri
Default font size	10pt	11pt
Default row height	12.75 pt	15 pt

Because the report explicitly sets the row height, the default row height affects only rows that are sized automatically upon export to Excel.

Report Items in Excel

Rectangles, subreports, the report body, and data regions are rendered as a range of Excel cells. Text boxes, images, and charts, data bars, sparklines, maps, gauges, and indicators must be rendered within one Excel cell, which might be merged depending

on the layout of the rest of the report.

Images, charts, sparklines, data bars, maps, gauges, indicators, and lines are positioned within one Excel cell but they sit on top of the cell grid. Lines are rendered as cell borders.

Charts, sparklines, data bars, maps, gauges, and indicators are exported as pictures. The data they depict, such as the value and member labels for a chart, is not exported with them and is not available in the Excel workbook unless it is included in a column or row in a data region within a report.

If you want to work with chart, sparkline, data bar, maps, gauge, and indicator data, export the report to a .csv file or generate Atom-compliant data feeds from the report. For more information, see Exporting to a CSV File (Report Builder and SSRS) and Generating Data Feeds from Reports (Report Builder and SSRS).

Page Sizing

The Excel rendering extension uses the page height and width settings to determine what paper setting to define in the Excel worksheet. Excel tries to match the PageHeight and PageWidth property settings to one of the most common paper sizes.

If no matches are found, Excel uses the default page size for the printer. Orientation is set to Portrait if the page width is less than the page height; otherwise, orientation is set to Landscape.

Worksheet Tab Names

When you export a report to Excel, the report pages that were created by page breaks are exported to different worksheets. If you provided an initial page name for the report, each worksheet of the Excel workbook will have this name by default. The name appears on the worksheet tab. However, since each worksheet in a workbook must have a unique name, an integer starting at 1 and incremented by 1 is appended to the initial page name for each additional worksheet. For example, if the initial page name is **Sales Report by Fiscal Year**, the second worksheet would be named **Sales Report by Fiscal Year1**, the third one **Sales Report by Fiscal Year2**, and so forth.

If all report pages created by page breaks provide new page names, each worksheet will have the associated page name. However, these page names might not be unique. If page names are not unique, the worksheets are named the same way as initial page names. For example, if the page name of two groups is **Sales for NW**, one worksheet tab will have the name **Sales for NW**, and the other **Sales for NW1**.

If the report provides neither an initial page name, nor page names related to page breaks, the worksheet tabs will have the default names **Sheet1**, **Sheet2**, and so forth.

Reporting Services provides properties to set on reports, data regions, groups, and rectangles to help you create reports that can be exported to Excel in a way that you want. For more information, see Pagination in Reporting Services (Report Builder and SSRS).

Document Properties

The Excel renderer writes the following metadata to the Excel file.

REPORT ELEMENT PROPERTIES	DESCRIPTION
Created	Date and time of report execution as an ISO date/time value.
Author	Report.Author
Description	Report.Description
LastSaved	Date and time of report execution as an ISO date/time value.

Page Headers and Footers

Depending on the Device Information SimplePageHeaders setting, the page header can be rendered in two ways: the page header can be rendered at the top of each worksheet cell grid, or in the actual Excel worksheet header section. By default, the header is rendered to the cell grid on the Excel worksheet.

The page footer is always rendered to the actual Excel worksheet footer section, regardless of the value of the SimplePageHeaders setting.

Excel header and footer sections support a maximum of 256 characters, including markup. If this limit is exceeded, the Excel renderer removes markup characters starting at the end of the header and/or footer string to reduce the number of total characters. If all markup characters are removed and the length still exceeds the maximum, the string is truncated starting from the right.

SimplePageHeader Settings

By default, the Device Information SimplePageHeaders setting is set to **False**; therefore, the page headers are rendered as rows in the report on the Excel worksheet surface. The worksheet rows that contain the headers become locked rows. You can freeze or unfreeze the pane in Excel. If the **Print Titles** option is selected, these headers are automatically set to print on every worksheet page.

The page header repeats at the top of every worksheet in the workbook except the document map cover sheet if the **Print Titles** option is selected on the Page Layout tab in Excel. If the **Print on first page** or the **Print on last page** option is not selected in the Report Header Properties or Report Footer Properties dialog boxes, the header will not be added to the first or last page respectively.

Page footers are rendered in the Excel footer section.

Because of Excel limitations, text boxes are the only type of report item that can be rendered in the Excel header/footer section.

Interactivity

Some interactive elements are supported in Excel. The following is a description of specific behaviors.

Show and Hide

Microsoft Excel has limitations with how it manages hidden and displayed report items when they are exported. Groups, rows, and columns that contain report items that can be toggled are rendered as Excel outlines. Excel creates outlines that expand and collapse rows and columns across the entire row or column which can cause the collapse of report items that are not intended to be collapsed. In addition, Excel's outlining symbols can become cluttered with overlapping outlines. To address these issues, the following outlining rules are applied when using the Excel rendering extension:

- The report item in the top-left corner that can be toggled can continue to be toggled in Excel. Report items that can be toggled and share vertical or horizontal space with the report item that can be toggled in the top-left corner cannot be toggled in Excel.
- To determine whether a data region will be collapsible by rows or columns, the position of the report item that controls the toggling and the position of the report item that is toggled are determined. If the item controlling the toggling appears before the item to be toggled, the item is collapsible by rows. Otherwise, the item is collapsible by columns. If the item controlling the toggling appears beside and above the area to be toggled equally, the item is rendered with row collapsible by rows.
- To determine where the subtotals are placed in the rendered report, the rendering extension examines the first instance of a dynamic member. If a peer static member appears immediately above it, the dynamic member is assumed to be the subtotals. Outlines are set to indicate that this is summary data. If there are no static siblings of a dynamic member, the first instance of the instance is the subtotal.
- Due to an Excel limitation, outlines can be nested up to 7 levels only.

Document Map

If any document map labels exist in the report, a document map is rendered. The document map is rendered as an Excel cover worksheet inserted at the first tab position in the workbook. The worksheet is named **Document map**.

The text displayed in the document map is determined by the report item's or group's DocumentMapLabel property. Document map labels are listed in the order that they appear in the report, starting at the first row, in the first column. Each document map label cell is indented the number of levels deep it appears in the report. Each level of indentation is represented by placing the label in a subsequent column. Excel supports up to 256 levels of outline nesting.

The document map outline is rendered as a collapsible Excel outline. The outline structure matches the nested structure of the document map. The expand and collapse state of the outline starts at the second level.

The root node of the map is the report name, the <*reportname*>.rdl, and it is not interactive. The document map links font is Arial, 10pt.

Drillthrough Links

Drillthrough links that appear in text boxes are rendered as Excel hyperlinks in the cell in which the text is rendered. Drillthrough links for images and charts are rendered as Excel hyperlinks on the image when rendered. When clicked, the drillthrough link opens the client's default browser and navigates to the HTML view of the target.

Hyperlinks

Hyperlinks that appear in text boxes are rendered as Excel hyperlinks in the cell in which the text is rendered. Hyperlinks for images and charts are rendered as Excel hyperlinks on the image when rendered. When clicked, the hyperlink opens the client's default browser and navigates to the target URL.

Interactive Sorting

Excel does not support interactive sort.

Bookmarks

Bookmark links in text boxes are rendered as Excel hyperlinks in the cell in which the text is rendered. Bookmark links for images and charts are rendered as Excel hyperlinks on the image when rendered. When clicked, the bookmark goes to the Excel cell in which the bookmarked report item is rendered.

Changing Reports at Run-Time

If a report must render to multiple formats and it is not possible to create a report layout that renders the way you want in all required formats, then you might consider using the value in the RenderFormat built-in global to conditionally change the report appearance at run time. This way you can hide or show report items depending the renderer used to get the best results in each format. For more information, see Built-in Globals and Users References (Report Builder and SSRS).

See Also

Pagination in Reporting Services (Report Builder and SSRS)
Rendering Behaviors (Report Builder and SSRS)
Interactive Functionality for Different Report Rendering Extensions (Report Builder and SSRS)
Rendering Report Items (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)

Exporting to Microsoft Word (Report Builder and SSRS)

3/24/2017 • 13 min to read • Edit Online

The Word rendering extension renders paginated reports to the Microsoft Word format (.docx). The format is Office Open XML.

The content type of files generated by this renderer is **application/vnd.openxmlformats-officedocument.wordprocessingml.document** and the file extension is .docx.

See Export Reports (Report Builder and SSRS) for details on how to export to Word.

After you export the report to a Word document, you can change the contents of your report and design document-style reports such as mailing labels, purchase orders, or form letters.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Report Items in Word

Reports exported to Word appear as a nested table that represents the report body. A tablix data region is rendered as a nested table that reflects the structure of the data region in the report. Text boxes and rectangles are each rendered as a cell within the table. The text box value is displayed inside the cell.

Images, charts, data bars, sparklines, maps, indicators, and gauges are each rendered as a static image within a table cell. Hyperlinks and drillthrough links on these report items are rendered. Maps and areas that can be clicked within a chart are not supported.

Newsletter-style column reports are not rendered in Word. Report body and page background images and colors are not rendered.

Pagination

After the report is opened in Word, Word repaginates the entire report again based on the page size. Repagination may cause page breaks to be inserted in locations where you did not intend to add them and, in some instances, may cause the exported report to have two successive page breaks in a row or add blank pages. You can try to change Word's pagination by adjusting the page margins.

This renderer supports only logical page breaks.

Page Sizing

When the report is rendered, the Word page height and width are set by the following RDL properties: paper size height and width, left and right page margins, and the top and bottom page margins.

Page Width

Word supports page widths that are up to 22 inches wide. If the report is wider than 22 inches, the renderer will still render the report; however, Word will not display the report contents while in print layout view or reading layout view. To view the data, switch to normal view or Web layout view. In these views, Word reduces the amount of whitespace, thereby displaying more of your report contents.

When rendered, the report grows as wide as required, up to 22 inches, to display the contents. The minimum width of the report is based on the RDL Width property in the Properties pane.

Document Properties

The Word renderer writes the following metadata to the DOCX file.

REPORT ELEMENT PROPERTIES	DESCRIPTION
Report Title (report title)	Title
Report.Author	Author
Report.Description	Comments

Page Headers and Footers

Page headers and footers are rendered as header and footer regions in Word. If a report page number or an expression that indicates the total number of report pages appears in the page header or footer, they are translated to a Word field so that the accurate page number is displayed

in the rendered report. If the header or footer height is set in the report, Word cannot support this setting. The PrintOnFirstPage property can under some circumstances specify whether text in a page header page footer prints on the first page of a report. If the rendered report has multiple pages and each page contains only a single section, then you can set PrintOnFirstPage to False and the text is suppressed on the first and page; otherwise, the text prints regardless of the value of the PrintOnFirstPage property.

The Word renderer attempts to parse all expressions in page headers and footers when reports are exported to Word. Many forms of expressions parse successfully and the expected values appear in page footers and headers on all report pages.

However, when a page footer or page header contains a complex expression that evaluates to different values on different pages of a report, the same value might display on all report pages. The page numbers in the following two expressions do not increment in the exported report. The page number translates to the same value on all report pages.

- `="Page: " + Globals!PageNumber.ToString + " of " + Globals!TotalPages.ToString`
- `=Avg(Fields!YTDPurchase.Value, "Sales") & " Page Number " & Globals!PageNumber`

  This occurs because Word renderer parses the report for fields related to pagination such as **PageNumber** and **TotalPages** and handles only simple reference, not calls to a function. In this case, the expression calls the **ToString** function. The following two expressions are equivalent and both render correctly when you preview the report in Report Builder or Report Designer or render the published report in a Reporting Services web portal or a SharePoint library. However, the Word renderer parses only the second expression successfully and renders the correct page numbers.

- **Complex expression:** Expression is `="Average Sales " & Avg(Fields!YTDPurchase.Value, "Sales") & " Page Number " & Globals!PageNumber`
- **Expression with Text Runs:** Text, **Average Sales**, and expression, `=Avg(Fields!YTDPurchase.Value, "Sales)`, and text, **Page Number**, and expression `=Globals!PageNumber`

  To avoid this problem, use multiple text runs instead of one complex expression when you use expressions in footers and headers. The following two expressions are equivalent. The first one is a complex expression the second one uses text runs. The Word renderer parses only the second expression successfully.

Interactivity

Some interactive elements are supported in Word. The following is a description of specific behaviors.

Show and Hide

The Word renderer renders report items based on their state when rendered. If a report item's state is hidden, the report item is not rendered in the Word document. If a report item's state is shown, the report item is rendered in the Word document. Toggle functionality is not supported in Word.

Document Map

If any document map labels exist in the report, they are rendered as Word Table of Contents (TOC) labels on the respective report items and groups. The document map label is used as the label text for the TOC labels. The target link is positioned near the item on which the label is set. While a TOC is not created for you in the Word document, you can build your own TOC using the document map labels that are rendered in the report.

Hyperlink and Drillthrough Links

Hyperlinks and drillthrough links on text box and image report items are rendered as hyperlinks in the Word document. When you click the hyperlink, the default Web browser opens and navigates to the URL. When you click the drillthrough hyperlink, the originating report server is accessed.

Interactive Sorting

The report contents are rendered based on how they are currently sorted within the report data region. Word does not support interactive sorting. After the report is rendered, you can apply table sorting within Word.

Bookmarks

Bookmarks in the report are rendered as Word bookmarks. Bookmark links are rendered as hyperlinks that connect to the bookmark labels within the document. Bookmark labels must be less than 40 characters long. The only special character that can be used in a bookmark label is an underscore (_). Unsupported special characters are stripped from the bookmark label name and, if the name is longer than 40 characters, the name is truncated. If there are duplicate bookmark names in the report, the bookmarks are not rendered in Word.

Word Style Rendering

The following is a brief description of how styles are rendered in Word.

Color Palette

Colors rendered in the report are rendered in the Word document.

Border

Borders for report items, other than the page border, are rendered as Word table cell borders.

Squiggly Lines in Exported Reports

When exported and viewed in Word, report data or constants might be underlined by red or green squiggly lines. The red squiggly lines identify spelling errors. The green squiggly lines identify grammar errors. This occurs when the report includes words that do not comply with the proofing (spelling and grammar) of the editing language that is specified in Word. For example, English report column titles will likely be underlined by red squiggly lines when the report is rendered in a Spanish version of Word. Perceived spelling errors are more common in reports than perceived grammar errors because reports typically include only short text, not complete sentences or paragraphs.

The presence of squiggly lines in reports implies the report has errors, which it likely does not. You can remove the squiggly lines by changing the proofing language for the report. To change the proofing language, select the content of the report and then specify the appropriate language for the content. You can select all or part of the content. In Word, the language option **Set Proofing Language** is in the **Language** area on the **Review** tab. After you update the content, you need to resave the document.

Depending on the language version of your Office program, the proofing tools (for example, dictionary) of the language that you chose is included with the program or provided in a Microsoft Office language pack that you purchase.

The following topics provide additional information about setting Office and Word options.

- Change the editing language in **Microsoft Office Language Preferences** or **Word Options** dialog box in Word. For more information, see Enable the use of other languages in your Office programs.
- Add Office language packs and then change the editing language. For more information, see Enable the use of other languages in your Office programs and Office Language Options.

> **NOTE**
> When you change the editing language in **Microsoft Office Language Preferences** or the **Word Options** dialog box in Word, the change applies to all Office programs.

Word Limitations

The following limitations are applied by Microsoft Word:

- Word tables support a maximum of 63 columns. If your report has more than 63 columns and you try to render it, Word splits the table. The additional columns are placed adjacent to the 63 columns displayed in the report body. Therefore, the report columns may not line up as expected.
- Word supports a maximum page width of 22 inches wide and 22 inches high. If your content is wider than 22 inches, some data may not be displayed in Print Layout view.
- Word ignores page header and footer height settings.
- After the report is exported, Word paginates the report again. This may cause additional page breaks to be added to the rendered report.
- Word does not repeat header rows on page two and greater, although you set the RepeatOnNewPage property of the static header row in a tablix (table, matrix, or list) to **True**. You can define explicit page breaks in your report to force header rows to appear on new pages. However, because Word applies its own pagination to the rendered report exported to Word, results might vary and the header row might not repeat predictably. The static header row is the row that contains the column headings.
- Text boxes grow when they contain non-breaking spaces.
- When text is exported to Word, text with font decoration in certain fonts may generate unexpected or missing glyphs in the rendered report.

Benefits of Using the Word Renderer

In addition to making the features that are new in Microsoft Word .docx files available to exported reports, *.docx files of exported reports tend to be smaller. Reports exported by using the Word renderer are typically significantly smaller than the same reports exported by using the Word 2003 renderer.

Backward Compatibility of Exported Reports

You can select a Word compatibility mode and set compatibility options. The Word renderer creates documents with compatibility mode turned on. Resaving the documents with compatibility mode turned off might affect the layout of the document.

If you turn off compatibility mode and then resave a report, the report layout might change in unexpected ways.

The Word 2003 Renderer

> **IMPORTANT**
> The Microsoft Word 2003 (.doc) rendering extension is deprecated. For more information, see Deprecated Features in SQL Server Reporting Services in SQL Server 2016.

The Word renderer is compatible with Microsoft Word 2003 with the Microsoft Office Compatibility Pack for Word, Excel, and PowerPoint installed. For more information, see Microsoft Office Compatibility Pack for Word, Excel, and PowerPoint.

The previous version of the Word rendering extension, compatible with Microsoft Word 2003, is renamed to Word 2003. Only the Word rendering extension is available by default. You must update the Reporting Services configuration files to make the Word 2003 rendering extension available. The content type of files generated by the Word 2003 renderer is **application/vnd.ms-word** and the file name extension of files is .doc.

In SQL Server 2016 Reporting Services, the default Word renderer is the version that renders to the Microsoft Word format (.docx). This is the **Word** option that the **Export** menus in a Reporting Services web portal and SharePoint list. The earlier version, compatible only with Microsoft Word 2003, is now named Word 2003 and is listed on menus using that name. The **Word 2003** menu option is not visible by default, but an administrator can make it visible by updating the RSReportServer configuration file. To export reports from SQL Server Data Tools (SSDT) using the Word 2003 renderer, you update the RSReportDesigner configuration file. However, making the Word 2003 renderer visible does not make it available in all scenarios. Because the RSReportServer configuration file resides on the report server, the tools or products from where you export reports must be connected to a report server to read the configuration file. If you use tools or products in disconnected or local mode, making the Word 2003 renderer visible has no effect. The **Word 2003** menu option remains unavailable. If you make the Word 2003 renderer visible in the RSReportDesigner configuration file, the **Word 2003** menu option is always available in SQL Server Data Tools (SSDT) report preview.

The **Word 2003** menu option is never visible in the following scenarios:

- Report Builder in disconnected mode and you preview a report in Report Builder.
- Report Viewer Web Part in local mode and the SharePoint farm is not integrated with a Reporting Services report server. For more information, see Local Mode vs. Connected Mode Reports in the Report Viewer (Reporting Services in SharePoint Mode)

  If the **Word 2003** renderer is configured to be visible, both the **Word** and **Word 2003** menu options are available in the following scenarios:
- Reporting Services web portal when Reporting Services is installed in native mode.
- SharePoint site when Reporting Services is installed in SharePoint integrated mode.
- SQL Server Data Tools (SSDT) and you preview reports.
- Report Builder connected to a report server.
- The Report Viewer Web Part in remote mode.

  The following XML shows the elements for the two Word rendering extensions in the RSReportServer and RSReportDesigner configuration files:

```
<Extension Name="WORDOPENXML"
Type="Microsoft.ReportingServices.Rendering.WordRenderer.WordOpenXmlRenderer.WordOpenXmlDocumentRenderer,Microsoft.ReportingServices.WordRendering"/>
```

```
<Extension Name="WORD"
Type="Microsoft.ReportingServices.Rendering.WordRenderer.WordDocumentRenderer,Microsoft.ReportingServices.WordRendering"
Visible="false"/>
```

  The WORDOPENXML extension defines the Word renderer for Microsoft Word .docx files. The WORD extension defines the Microsoft Word 2003 version. `Visible = “false”` indicates the Word 2003 renderer is hidden. For more information, see RsReportServer.config Configuration File and RSReportDesigner Configuration File.

Differences Between the Word and Word 2003 Renderers

Reports, rendered by using the Word or Word 2003 renderers tend to be visually indistinguishable. However, you might notice minor differences between the two the Word or Word 2003formats.

Device Information Settings

You can change some default settings for this renderer, such as omit hyperlinks and drillthrough links or expand all items that can be toggled regardless of the original state of the item when rendered, by changing the device information settings. For more information, see Word Device Information Settings.

See Also

Pagination in Reporting Services (Report Builder and SSRS)
Rendering Behaviors (Report Builder and SSRS)
Interactive Functionality for Different Report Rendering Extensions (Report Builder and SSRS)

Rendering Report Items (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)

Rendering to HTML (Report Builder and SSRS)

3/24/2017 • 4 min to read • Edit Online

The HTML rendering extension renders a paginated report in HTML format. The rendering extension can also produce fully formed HTML pages or fragments of HTML to embed in other HTML pages. All HTML is generated with UTF-8 encoding.

The HTML rendering extension is the default rendering extension for reports that are viewed in a browser, including when run in the SQL Server 2016 Reporting Services (SSRS) web portal.

The HTML rendering extension is the default rendering extension for reports that are viewed in a browser, including when run in the SQL Server 2016 Reporting Services (SSRS) web portal. The HTML rendering extension can render HTML as a fragment or as a full HTML document. If the HTML is a fragment, the **HEAD**, **HTML**, and **BODY** tags of the HTML document are removed. Only the contents of the **BODY** tag are rendered. This is useful for embedding the HTML in the HTML produced by another application.

In some scenarios, report parameters can be used to launch script injection attacks when rendering reports to HTML. For more information about securing reports, see Secure Reports and Resources.

For more information about browsers, see Browser Support for Reporting Services and Power View.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Rendering in MHTML

The HTML rendering extension can also render reports in MHTML (MIME Encapsulation of Aggregate HTML Documents). MHTML extends HTML to embed encoded objects, such as images, in the HTML document. Using the MHTML rendering extension, you can embed resources such as images, documents, or other binary files as MIME structures within the report HTML, into a single file. MHTML reports are also useful for embedding within e-mail messages because all resources are included with the report. Although it is actually the HTML rendering extension that renders MHTML, this functionality may also be referred to as the MHTML rendering extension.

Browser Support

This rendering extension supports the following browser versions:

- Internet Explorer 5.5 and later
- Firefox 1.5 and later
- Safari 3.0 and later

  Due to cross browser considerations, the rendered report may vary slightly from browser to browser. For example, the text box contains a property called WritingMode. This property is not supported in Firefox.

HTML-Specific Rendering Rules

The following HTML-specific rules are applied when rendering:

- The renderer builds an HTML table structure to contain all of the items in each **ReportItems** collection, if

there is more than one.

- Every item within the table structure occupies a single cell.
- Empty cells are collapsed together as much as possible to reduce the size of the HTML.
- A row of empty cells is added to the top edge and another column to the left edge to improve the speed at which browsers can render the table.
- Table rows or columns that contain no items, just gaps between items, are given fixed widths and heights.
- All other rows and columns are allowed to grow depending on the size of each report item.
- All coordinates and report item sizes are converted to millimeters. All other sizes, including style properties, retain their original units. Size and position differences smaller than .2mm are treated as 0mm.

Interactivity

Some interactive elements are supported in HTML. The following is a description of specific behaviors.

Show and Hide

A report item whose visibility can be toggled is rendered with a +/- toggle image and is clickable. When the item is clicked, a call back to the server takes place in order to re-render the output with the changed show or hide state.

Document Map

Document map labels are rendered and can be navigated to by using the document map in the viewer control. For omitted data region headers, labels are rendered on the first child cell. If there is no child cell present, the label is rendered on the child that precedes it.

Bookmarks

Bookmark links are rendered and appear as hyperlinks. Bookmark targets are rendered and can be navigated to by clicking the bookmark links. When a bookmark link is clicked, the report goes to the first occurrence of the target bookmark label and, when possible, the browser is scrolled so that the bookmark link is at the top of the window. HTML anchor (<a>) tags are used to mark bookmark targets.

Interactive Sorting

If a text box has user sort defined, the HTML rendering extension renders the sort icons in the text box to the right of its contents. If a report contains any text box where user sort is defined, JavaScript is rendered that causes a postback to the server when the sort image is clicked.

Hyperlinks and Drillthrough

Hyperlinks and drillthrough links are rendered as hyperlinks on report items using the HTML anchor (<a>) tags around the item on which they are defined.

Search

The Search feature allows users to search for a string of text within the report.

Additional search and find functionality is provided by the ReportViewer Web Forms control.

Device Information Settings

You can change some default settings for this renderer, including which mode to render in, by changing the device information settings. For more information, see HTML Device Information Settings.

See Also

Pagination in Reporting Services (Report Builder and SSRS)

Rendering Behaviors (Report Builder and SSRS)
Interactive Functionality for Different Report Rendering Extensions (Report Builder and SSRS)
Rendering Report Items (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)

Exporting to an Image File (Report Builder and SSRS)

3/24/2017 • 1 min to read • Edit Online

The Image rendering extension renders a paginated report to a bitmap or metafile. By default, the Image rendering extension produces a TIFF file of the report, which can be viewed in multiple pages. When the client receives the image, it can be displayed in an image viewer and printed. This topic provides Image renderer-specific information and describes exceptions to the rendering rules.

The Image rendering extension can generate files in any of the formats supported by GDI+: BMP, EMF, EMFPlus, GIF, JPEG, PNG, and TIFF. For TIFF format, the file name of the primary stream is *ReportName*.tif. For all other formats, which render as a single page per file, the file name is *ReportName_Page.ext* where. *ext* is the file extension for the chosen format. To produce a file in another Image-supported format, specify any of the above listed strings in the **OutputFormatDeviceInfo** setting.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Supported Image Formats

The following table shows the file extension and MimeType for each Image renderer format.

TYPE	EXTENSION	MIMETYPE
BMP	bmp	image/bmp
GIF	gif	image/gif
JPEG	jpeg	image/jpeg
PNG	png	image/png
TIFF	tif	image/tiff
EMF	emf	image/emf
EMFPlus	emf	image/emf

Rendering Multiple Pages

TIFF is the only format that supports multiple page documents in a single file. Other formats, such as JPG or PNG, output one page at a time and require a separate call to the rendering extension for each page.

Interactivity

Interactivity is not supported in any Image formats generated by this renderer. The following interactive elements are not rendered:

- Hyperlinks
- Show or Hide
- Document Map
- Drillthrough or clickthrough links
- End user sort
- Fixed headers
- Bookmarks

Device Information Settings

You can change some default settings for this renderer by changing the device information settings. For more information, see Image Device Information Settings.

See Also

Pagination in Reporting Services (Report Builder and SSRS)
Rendering Behaviors (Report Builder and SSRS)
Interactive Functionality for Different Report Rendering Extensions (Report Builder and SSRS)
Rendering Report Items (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)

Exporting to a PDF File (Report Builder and SSRS)

3/24/2017 • 5 min to read • Edit Online

The PDF rendering extension renders Reporting Services paginated reports to files that can be opened in Adobe Acrobat and other third-party PDF viewers that support PDF 1.3. Although PDF 1.3 is compatible with Adobe Acrobat 4.0 and later versions, Reporting Services supports Adobe Acrobat 11.0 or later. The rendering extension does not require Adobe software to render the report. However, PDF viewers such as Adobe Acrobat are required to view or print a report in PDF format.

The PDF rendering extension supports ANSI characters and can translate Unicode characters from Japanese, Korean, Traditional Chinese, Simplified Chinese, Cyrillic, Hebrew, and Arabic with certain limitations. For more information about the limitations, see Export Reports (Report Builder and SSRS).

The PDF renderer is a physical page renderer and, therefore, has pagination behavior that differs from other renderers such as HTML and Excel. This topic provides PDF renderer-specific information and describes exceptions to the rules.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Font Embedding

When possible, the PDF rendering extension embeds the subset of each font that is needed to display the report in the PDF file. Fonts that are used in the report must be installed on the report server. When the report server generates a report in PDF format, it uses the information stored in the font referenced by the report to create character mappings within the PDF file. If the referenced font is not installed on the report server, the resulting PDF file might not contain the correct mappings and might not display correctly when viewed.

Fonts are embedded in the PDF file when the following conditions apply:

- Font embedding privileges are granted by the font author. Installed fonts include a property that indicates whether the font author intends to allow embedding a font in a document. If the property value is EMBED_NOEMBEDDING, the font is not embedded in the PDF file. For more information, see "TTGetEmbeddingType" on msdn.microsoft.com.
- The Font is TrueType.
- Fonts are referenced by visible items in a report. If a font is referenced by an item that has the Hidden property set to True, the font is not needed to display rendered data and will not be included in the file. Fonts are embedded only when they are needed to display the rendered report data.

  If all of these conditions are met for a font, the font is embedded in the PDF file. If one or more of these conditions is not met, the font is not embedded in the PDF file.

> **NOTE**
> Although the conditions are met, there is one circumstance under which fonts are not embedded in the PDF file. If the fonts used are the ones in the PDF specification that are commonly known as standard type 1 fonts or the base fourteen fonts, then fonts are not embedded for ANSI content.

Fonts on the Client Computer

When a font is embedded in the PDF file, the computer that is used to view the report (the client computer) does not need to have the font installed for the report to display correctly.

When a font is not embedded in the PDF file, the client computer must have the correct font installed for the report to display correctly. If the font is not installed on the client computer, the PDF file displays a question mark character (?) for unsupported characters.

Verifying Fonts in a PDF File

Differences in PDF output occur most often when a font that does not support non-Latin characters is used in a report and then non-Latin characters are added to the report. You should test the PDF rendering output on both the report server and the client computers to verify that the report renders correctly.

Do not rely on viewing the report in Preview or exporting to HTML because the report will look correct due to automatic font substitution performed by the graphical design interface or by Microsoft Internet Explorer, respectively. If there are Unicode Glyphs missing on the server, you may see characters replaced with a question mark (?). If there is a font missing on the client, you may see characters replaced with boxes (□).

The fonts that are embedded in the PDF file are included in the Fonts property that is saved with the file, as metadata.

Metadata

In addition to the report layout, the PDF rendering extension writes the following metadata to the PDF Document Information Dictionary.

PDF PROPERTY	CREATED FROM
Title	The **Name** attribute of the **Report** RDL element.
Author	The **Author** RDL element.
Subject	The **Description** RDL element.
Creator	Reporting Services product name and version.
Producer	Rendering extension name and version.
CreationDate	Report execution time in PDF **datetime** format.

Interactivity

Some interactive elements are supported in PDF. The following is a description of specific behaviors.

Show and Hide

Dynamic show and hide elements are not supported in PDF. The PDF document is rendered to match the current state of any items in the report. For example, if the item is displayed when the report is run initially, then the item is rendered. Images that can be toggled are not rendered, if they are hidden when the report is exported.

Document Map

If there are any document map labels present in the report, a document outline is added to the PDF file. Each document map label appears as an entry in the document outline in the order that it appears in the report. In Acrobat, a target bookmark is added to the document outline only if the page it is on is rendered.

If only a single page is rendered, no document outline is added. The document map is arranged hierarchically to

reflect the level of nesting in the report. The document outline is accessible in Acrobat under the Bookmarks tab. Clicking an entry within the document outline causes the document to go to the bookmarked location.

Bookmarks

Bookmarks are not supported in PDF rendering.

Drillthrough Links

Drillthrough links are not supported in PDF rendering. The drillthrough links are not rendered as clickable links and drillthrough reports cannot connect to the target of the drillthrough.

Hyperlinks

Hyperlinks in reports are rendered as clickable links in the PDF file. When clicked, Acrobat will open the default client browser and navigate to the hyperlink URL.

Compression

Image compression is based on the original file type of the image. The PDF rendering extension compresses PDF files by default.

To preserve any compression for images included in the PDF file when possible, JPEG images are stored as JPEG and all other image types are stored as BMP.

> **NOTE**
> PDF files don't support embedding PNG images.

Device Information Settings

You can change some default settings for this renderer by changing the device information settings. For more information, see PDF Device Information Settings.

See Also

Pagination in Reporting Services (Report Builder and SSRS)
Rendering Behaviors (Report Builder and SSRS)
Interactive Functionality for Different Report Rendering Extensions (Report Builder and SSRS)
Rendering Report Items (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)

Exporting to XML (Report Builder and SSRS)

4/14/2017 • 7 min to read • Edit Online

The XML rendering extension returns a paginated report in XML format. The schema for the report XML is specific to the report, and contains data only. Layout information is not rendered and pagination is not maintained by the XML rendering extension. The XML generated by this extension can be imported into a database, used as an XML data message, or sent to a custom application.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Report Items

The following table describes how report items are rendered.

ITEM	RENDERING BEHAVIOR
Report	Renders as the top-level element of the XML document.
Data regions	Renders as an element within the element for its container. Data regions include table, matrix, and list that display data as text and chart, data bars, sparklines, gauges, and indicators that visualize data.
Group and detail sections	Each instance renders as an element within the element for its container.
Text box	Renders as an attribute or element within its container.
Rectangle	Renders as an element within its container.
Matrix column groups	Renders as elements within row groups.
Map	Renders as an element within the element for its container. Map layers are child elements of the map and each map layer includes elements for their map members and map member attributes.
Chart	Renders as an element within the element for its container. Series are child elements of the chart, and categories are child element of a series. Renders all chart labels for each chart value. Labels and values are included as attributes.
Data bar	Renders as an element within the element for its container, similar to a chart. Typically, a data bar does not include hierarchies or labels, only values.

ITEM	RENDERING BEHAVIOR
Sparkline	Renders as an element within the element for its container, similar to a chart. Typically, a sparkline does not include hierarchies or labels, only values.
Gauge	Renders as an element within the element for its container. Renders as a single element with the minimum and maximum values of the scale, start and end values of the range, and the value of the pointer as attributes.
Indicator	Renders as an element within the element for its container, similar to a gauge. Renders as a single element with the active state name, available states, and the data value as attributes.

Reports that are rendered using the XML rendering extension also follow these rules:

- XML elements and attributes are rendered in the order that they appear in the report definition.
- Pagination is ignored.
- Page headers and footers are not rendered.
- Hidden items that cannot be made visible by toggling are not rendered. Initially visible items and hidden items that can be made visible through a toggle are rendered.
- **Images, lines, and custom report items** are ignored.

Data Types

The text box element or attribute is assigned an XSD data type based on the values that the text box displays.

IF ALL TEXT BOX VALUES ARE	ASSIGNED DATA TYPE IS
Int16, **Int32**, **Int64**, **UInt16**, **UInt32**, **UInt64**, **Byte**, **SByte**	**xsd:integer**
Decimal (or **Decimal** and any integer or byte data type)	**xsd:decimal**
Float (or **Decimal** and any integer or byte data type)	**xsd:float**
Double (or **Decimal** and any integer or byte data type)	**xsd:double**
DateTime or DateTime Offset	**xsd:dateTime**
Time	**xsd:string**
Boolean	**xsd:boolean**
String, **Char**	**xsd:string**
Other	**xsd:string**

XML-Specific Rendering Rules

The following sections describe how the XML rendering extensions interprets the items within the report.

Report Body

A report is rendered as the root element of the XML document. The name of the element comes from the DataElementName property set in the Properties pane.

XML namespace definitions and schema reference attributes are also included in the report element. Variables are noted in bold face type:

<**Report** xmlns="**SchemaName**" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:**schemaLocation**="**SchemaNameReportURL**&rc%3aSchema=true" Name="ReportName">

The values for the variables are as follows:

NAME	VALUE
Report	Report.DataElementName
ReportURL	URLEncoded absolute URL to the report on the server.
SchemaName	Report.SchemaName. If null, then Report.Name. If Report.Name is used, it is first encoded with XmlConvert.EncodeLocalName.
ReportName	The name of the report.

Text boxes

Text boxes are rendered as elements or attributes according to the DataElementStyle RDL property. The name of the element or attribute comes from the TextBox.DataElementName RDL property.

Charts, Data Bars, and Sparklines

Charts ,data bars, and sparklines are rendered in XML. The data is structured.

Gauges and Indicators

Gauges and indicators are rendered in XML. The data is structured.

Subreports

A subreport is rendered as an element. The name of the element is taken from the DataElementName RDL property. The TextBoxesAsElements property setting of the report overrides that of the subreport. Namespace and XSLT attributes are not added to the subreport element.

Rectangles

A rectangle is rendered as an element. The name of the element is taken from the DataElementName RDL property.

Custom Report Items

CustomReportItems (CRI) are not visible to the rendering extension. If a custom report item exists in the report, the rendering extension renders it as a conventional report item.

Images

Images are not rendered.

Lines

Lines are not rendered.

Tables, Matrices, and Lists

Tables, matrices, and lists, are rendered as an element. The name of the element comes from the Tablix DataElementName RDL property.

Rows and Columns

Columns are rendered within rows.

Tablix Corner

The corner is not rendered. Only the contents of the corner are rendered.

Tablix Cells

Tablix cells are rendered as elements. The name of the element is taken from the cell's DataElementName RDL property.

Automatic Subtotals

Tablix automatic subtotals are not rendered.

Row and Column Items that Do Not Repeat with a Group

Items that do not repeat with a group, such as labels, subtotals and totals, are rendered as elements. The name of the element comes from the TablixMember.DataElementName RDL property.

The TablixMember.DataElementOutput RDL property controls whether a non-repeating item is rendered.

If the DataElementName property of the Tablix member is not provided, a name for the non-repeating item is dynamically generated in this form:

RowX For non-repeating rows, where X is a zero-based row index within the current parent.

ColumnY For non-repeating columns, where Y ix a zero-based column index within the current parent.

A non-repeating header is rendered as a child of the row or column that does not repeat with a group.

If a non-repeating member has no corresponding Tablix cells, it is not rendered. This may occur in the case of a Tablix cell where it spans more than one column.

Rows and Columns that Repeat with a Group

Rows and columns that repeat within a group are rendered according to Tablix.DataElementOutput rules. The name for the element is taken from the DataElementName property.

Each unique value within a group is rendered as a child element of the group. The name for the element is taken from the Group.DataElementName property.

If the DataElementOutput property value equals Output, a repeating item's header is rendered as a child of the detail element.

Custom Formats and XSL Transformations

XML files produced by the XML rendering extension can be transformed into almost any format using XSL Transformations (XSLT). This functionality can be used to produce data in formats not already supported by existing rendering extensions. Consider using the XML rendering extension and XSLT before attempting to create your own rendering extension.

Duplicate Names

If there are duplicate data element names within the same scope, the renderer displays an error message.

XSLT Transformations

The XML renderer can apply a server-side XSLT transformation to the original XML data. When an XSLT is applied, the renderer outputs the transformed content instead of the original XML data. The transformation occurs on the server, not on the client.

The XSLT to apply to the output is defined either in the report definition file with the DataTransform property of the

report or with the XSLT *DeviceInfo* parameter. If either of these values are set, the transform occurs each time the XML renderer is used. When using subscriptions, the XSLT must be defined in the RDL DataTransform property.

If an XSLT file is specified, by both the DataTransform definition property and the device information setting, the XSLT specified in DataTransform occurs first, followed by the XSLT set by the device information settings.

Device Information Settings

You can change some default settings for this renderer by changing the device information settings, including the following:

- A transformation (XSLT) to apply to the XML.
- The MIME type of the XML document.
- Whether to apply format strings to data.
- Whether to indent the XML output.
- Whether to include the XML schema name.
- The encoding for the XML document.
- The file extension of the XML document.

  For more information, see XML Device Information Settings.

See Also

Pagination in Reporting Services (Report Builder and SSRS)
Rendering Behaviors (Report Builder and SSRS)
Interactive Functionality for Different Report Rendering Extensions (Report Builder and SSRS)
Rendering Report Items (Report Builder and SSRS)
Tables, Matrices, and Lists (Report Builder and SSRS)

Generating Data Feeds from Reports (Report Builder and SSRS)

4/14/2017 • 10 min to read • Edit Online

The Reporting Services Atom rendering extension generates an Atom service document that lists the data feeds available from a paginated report and the data feeds from the data regions in a report. You use this extension to generate Atom-compliant data feeds that are readable and exchangeable with applications that can consume data feeds generated from reports. For example, you can use the Atom rendering extension to generated data feeds that you can then use in the SQL Server 2016 Power Pivot client.

The Atom service document lists at least one data feed for each data region in a report. Depending on the type of data region and the data that the data region displays, Reporting Services might generate multiple data feeds from a data region. For example, a matrix or chart can provide multiple data feeds. When the Atom rendering extension creates the Atom service document, a unique identifier is created for each data feed and you use the identifier in the URL to access the content of the data feed.

The way that the Atom rendering extension generates data for a data feed is similar to how the Comma-Separated Value (CSV) rendering extension renders data to a CSV file. Like a CSV file, a data feed is a flattened representation of the report data. For example, a table with a row group that sums the sales within a group repeats the sum in every data row and there is no separate row that contains only the sum.

You can generate Atom service documents and data feeds using the Reporting Services web portal, Report Server, or a SharePoint site that is integrated with Reporting Services.

Atom applies to a pair of related standards. The Atom service document conforms to the RFC 5023 Atom publishing protocol specification and the data feeds conform to the RFC 4287 Atom syndication format protocol specification.

The following sections provide additional information about how to use the Atom rendering extension:

You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Reports as Data Feeds

You can export a production report as a data feed or you can create a report whose primary purpose is provide data, in the form of data feeds, to applications. Using reports as a data feed gives you an additional way to provide data to applications when the data is not easy to access through client data providers, or when you prefer to hide the complexity of the data source and make it simpler to use the data. Another benefit of using report data as a data feed is that you can use Reporting Services features such as security, scheduling, and report snapshots to manage the reports that provide data feeds.

To get the most from the Atom rendering extension, you should understand how the report is rendered into data feeds. If you are using existing reports, being able to predict what the data feeds the reports will generate is useful; if you are writing report specifically for use as data feeds, being able to include the data and fine tune the report layout to maximize the usefulness of the data feeds is valuable.

For more information, see Generate Data Feeds from a Report (Report Builder and SSRS).

Atom Service Document (.atomsvc file)

An Atom service document specifies a connection to one or more data feeds. At a minimum, the connection is a simple URL to the data service that produces the feed.

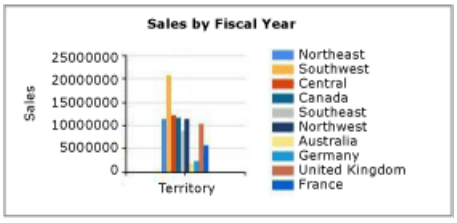
When you render report data by using the Atom rendering extension, the Atom service document lists the data feeds available for a report. The document lists at least one data feed for each data region in the report. Tables and gauges generate only one data feed each, but matrices, lists, and charts might generated multiple depending on the data they display.

The following diagram shows a report that uses two tables and a chart.

Sales By Sales Person & Fiscal Year

Full Name	Job Title	Sales Territory
Michael G Blythe	Sales Representative	Northeast
Lynn N Tsoflias	Sales Representative	Australia
Rachel B Valdez	Sales Representative	Germany
Jae B Pak	Sales Representative	United Kingdom
Ranjit R Varkey Chudukatil	Sales Representative	France

Full Name	2002	2003	2004
Michael G Blythe	19510856.8256	4743906.8935	4557045.0459
Lynn N Tsoflias			1758385.9260
Rachel B Valdez			2241204.0424
Jae B Pak		5287044.3125	5015682.3751
Ranjit R Varkey Chudukatil		16775 652.4369	3827950.2378



The Atom service document generated from this report includes three data feeds, one for each table and one for the chart.

The matrix data regions might have more than one data feed, depending on the structure of the matrix. The following diagram shows a report that uses a matrix that generates two data feeds.

Territory Year

Sales by Area and Year		North America		2003	2004	Total
		US	CA			
Clothing	Caps	$9798.76	$2801.06	$9008.44	$3591.38	$12599.82
	Tights	$51391.51	$8233.90	$59625.41		$59625.41
Components	Chains	$4972.00	$793.83	$3543.48	$2222.35	$5765.83
	Cranksets	$105118.64	$18515.99	$72889.77	$50744.86	$123634.62
	Touring Frames	$767130.33	$196141.85	$603497.08	$359775.10	$963272.17

The Atom service document generated from this report includes two data feeds, one for each of the dynamic peer columns: Territory and Year. The following diagram shows the content of each data feed.

Sales by Area and Year		North America	
		US	CA
Clothing	Caps	$9798.76	$2801.06
	Tights	$51391.51	$8233.90
Components	Chains	$4972.00	$793.83
	Cranksets	$105118.64	$18515.99
	Touring Frames	$767130.33	$196141.85

Data Feed

Sales by Area and Year		2003	2004	Total
Clothing	Caps	$9008.44	$3591.38	$12599.82
	Tights	$59625.41		$59625.41
Components	Chains	$3543.48	$2222.35	$5765.83
	Cranksets	$72889.77	$50744.86	$123634.62
	Touring Frames	$603497.08	$359775.10	$963272.17

Data Feed

Data Feeds

The data feed is an XML file that has a consistent tabular format that does not change over time and variable data that can be different each time the report is run. The data feeds generated by Reporting Services are in the same format as those generated by that ADO.NET Data Services.

A data feed contains two sections: header and data. The Atom specification defines the elements in each section. The header includes information such as the character encoding schema to use with the data feeds.

Header Section

The following XML code shows the header section of a data feed.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?><feed
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns="http://www.w3.org/2005/Atom">
```

```
<title type="text"></title>
```

```
<id>uuid:1795992c-a6f3-40ec-9243-fbfd0b1a5be3;id=166321</id>
```

```
<updated>2009-05-08T23:09:58Z</updated>
```

Data Section

The data section of the data feeds contains one <**entry**> element for each row in the flattened rowset generated by the Atom rendering extension.

The following diagram shows a report that uses groups and totals.

Product Category	Order Year	Sum Line Total	Sum Order Qty
Accessories	2001	$20,235.36	1003
	2002	$92,735.35	5207
	2003	$590,257.59	28162
	2004	$568,844.58	27560
	Total	$1,272,072.88	61932
Bikes	2001	$10,661,722.28	7139
	2002	$26,486,358.20	24908
	2003	$34,923,280.24	37042
	2004	$22,579,811.98	21179
	Total	$94,651,172.70	90268
Clothing	2001	$34,376.34	2132
	2002	$485,587.15	16927
	2003	$1,011,984.50	35377
	2004	$588,594.53	19234
	Total	$2,120,542.52	73670
Components	2001	$615,474.98	1574
	2002	$3,610,092.47	13876
	2003	$5,485,514.83	24118
	2004	$2,091,511.00	9476
	Total	$11,802,593.29	274914
Total		$109,846,381.40	274914

The following XML shows an <**entry**> element from that report in a data feed. Notice that the <**entry**> element includes the totals of the sales and orders for the group and the totals of sales and orders for all the groups. The <**entry**> element includes all values on the report.

```
<entry><id>uuid:1795992c-a6f3-40ec-9243-fbfd0b1a5be3;id=166322</id><title type="text"></title><updated>2009-05-08T23:09:58Z</updated><author /><content type="application/xml"><m:properties>

<d:ProductCategory_Value>Accessories</d:ProductCategory_Value>

<d:OrderYear_Value m:type="Edm.Int32">2001</d:OrderYear_Value>

<d:SumLineTotal_Value m:type="Edm.Decimal">20235.364608</d:SumLineTotal_Value>

<d:SumOrderQty_Value m:type="Edm.Int32">1003</d:SumOrderQty_Value>

<d:SumLineTotal_Total_2_1 m:type="Edm.Decimal">1272072.883926</d:SumLineTotal_Total_2_1>

<d:SumOrderQty_Total_2_1 m:type="Edm.Double">61932</d:SumOrderQty_Total_2_1>

<d:SumLineTotal_Total_2_2 m:type="Edm.Decimal">109846381.399888</d:SumLineTotal_Total_2_2>

<d:SumOrderQty_Total_2_2 m:type="Edm.Double">274914</d:SumOrderQty_Total_2_2></m:properties></content>

</entry>
```

Working with Data Feeds

All data feeds generated by the report include the report items that are in scope of the parent of the data region that generate the data feeds. . Imagine a report that has several tables and a chart. Text boxes in the report body provides descriptive text of each data region. Every entry in every data feed that the report generates includes the value of the text box. For example, if the text is "Chart displays monthly sales averages by sales region", all three data feeds would include this text on each row.

If the report layout includes hierarchical data relationships, such as nested data regions, those relationships are included in the flattened rowset of report data.

The data rows for nested data regions are typically wide, especially if the nested tables and matrices include groups and totals. You might find it useful to export the report to a data feed and view the data feed to verify that the data generated is what you expected.

When the Atom rendering extension creates the Atom service document, a unique identifier is created for the data

feed and you use the identifier in the URL to view the content of the data feed. The sample Atom service document, shown above, includes the URL

```
http://ServerName/ReportServer?
%2fProduct+Sales+Summary&rs%3aCommand=Render&rs%3aFormat=ATOM&rc%3aDataFeed=xAx0x1
```

. The URL identifies the report (Product Sales Summary), the Atom rendering format (ATOM), and the name of the data feed (xAx0x1).

Report item names default to the report definition language (RDL) element names of the report items and often they are not intuitive or easy to remember. For example, the default name of the first matrix placed in a report is Tablix 1. The data feeds use these names.

To make the data feed easier to work with, you can use the DataElementName property of the data region to provide friendly names. If you provide a value for DataElementName the data feed subelement <**d**> will use is it instead of the default data region name. For example, if the default name of a data regions is Tablix1 and DataElementName set SalesByTerritoryYear then the <**d**> in the data feed uses SalesByTerritoryYear. If the data regions has two data feeds like the matrix report described above, the names used in the data feeds are SalesByTerritoryYear _Territory and SalesByTerritoryYear _Year.

If you compare the data shown on the report and the data in the data feed, you might notice some differences. Reports often shows formatted numeric and time/date data whereas the data feed contains unformatted data.

A data feed is saved with the .atom file name extension. You can use a text or XML editor such as Notepad or XML Editor to view the file structure and content.

Flattening Report Data

The Atom renderer provides report data as flattened rowsets in an XML format. The rules for flattening data tables are the same to those of the CSV renderer with a few exceptions:

- Items in scope are flattened to the detail level. Unlike the CSV renderer, the text boxes at the top level appear in each entry written to the data feed.
- Report parameter values are rendered on each row of the output.

  Hierarchical and grouped data must be flattened in order to be represented in the Atom-compliant format. The rendering extension flattens the report into a tree structure that represents the nested groups within the data region. To flatten the report:
- A row hierarchy is flattened before a column hierarchy.
- Members of the row hierarchy are rendered to the data feed before members of the column hierarchy.
- Columns are ordered as follows: text boxes in body order left-to-right, top-to-bottom followed by data regions ordered left-to-right, top-to-bottom.
- Within a data region, the columns are ordered as follows: corner members, row hierarchy members, column hierarchy members, and then cells.
- Peer data regions are data regions or dynamic groups that share a common data region or dynamic ancestor. Peer data is identified by branching of the flattened tree.

  For more information, see Tables, Matrices, and Lists (Report Builder and SSRS).

Atom Rendering Rules

The Atom rendering extension ignores the following information when rendering a data feed:

- Formatting and layout

- Page header
- Page footer
- Custom report items
- Rectangles
- Lines
- Images
- Automatic subtotals

  The remaining report items are sorted, from top to bottom, and then left to right. Each item is then rendered to a column. If the report has nested data items like lists or tables, the parent items are repeated on each row.

  The following table indicates the appearance of report items when rendered:

ITEM	RENDERING BEHAVIOR
Table	Renders by expanding the table and creating a row and column for each row and column at the lowest level of detail. Subtotal rows and columns do not have column or row headings. Drillthrough reports are not supported.
Matrix	Renders by expanding the matrix and creating a row and column for each row and column at the lowest level of detail. Subtotal rows and columns do not have column or row headings.
List	Renders a record for each detail row or instance in the list.
Subreport	The parent item is repeated for each instance of the contents.
Chart	Renders a record with all chart labels for each chart value. Labels from series and categories in hierarchies are flattened and included in the row for a chart value.
Data bar	Renders like a chart. Typically, a data bar does not include hierarchies or labels.
Sparkline	Renders like a chart. Typically, a sparkline does not include hierarchies or labels.
Gauge	Renders as a single record with the minimum and maximum values of the linear scale, start and end values of the range, and the value of the pointer.
Indicator	Renders as a single record with the active state name, available states, and the data value.
Map	Generates a data feed for each map data region. If multiple map layers use the same data region, the data feed includes all of them. The data feed includes a record with the labels and values for each map member of the map layer.

Device Information Settings

You can change some default settings for this renderer, including the encoding schema to use. For more information, see ATOM Device Information Settings.

See Also

Exporting to a CSV File (Report Builder and SSRS)
Export Reports (Report Builder and SSRS)

Generate Data Feeds from a Report (Report Builder and SSRS)

3/24/2017 • 2 min to read • Edit Online

You can generate Atom-compliant data feeds from paginated reports, and then use the data feeds in applications, such as the SQL Server 2016 Power Pivot client, that can consume data feeds.

The Reporting Services Atom rendering extension generates an Atom service document that lists the data feeds available from a report. The document lists at least one data feed for each data region in the report. Depending on the type of data region and the data that the data region displays, Reporting Services might generate multiple data feeds from a data region.

Atom service document contains a unique identifier for each the data feed and you use the identifier in a URL to view the content of the data feed.

For more information, see Generating Data Feeds from Reports (Report Builder and SSRS).

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

To generate an Atom service document

1. on the Reporting Services web portal, navigate to the report for which you want to generate data feeds.
2. Click the report.

   The report is run.
3. On the toolbar, click the **Export to Data Feed** icon.

   A message appears asking you if you want to save or open the atom document that contains the data feed.
4. Click **Save** to save the document to the file system, or click **Open** to view the document content before saving. **By default, the document opens in a browser.**
5. Browse to the location to save the document.
6. Optionally, change the name of the document.

   > **NOTE**
   > By default, the document name is the report name.
7. Verify the document type is **ATOMSVC File**, and then click **Save**.
8. Optionally, open the .atomsvc file in a browser or text or XML editor.

To view an Atom-compliant data feed

1. If the Atom service document is not already open, locate it and open it in a browser such as Internet Explorer.
2. Copy the URL of the data feed that you want to view from the Atom service document to the browser.

The format of the URL is the following:

```
http://<server name>/ReportServer?%2f<ReportName>rs%3aCommand=Render&rs%3aFormat=ATOM&rc%3aDataFeed=
<Identifier>
```

3. Press ENTER.

   A message appears asking you if you want to save or open the atom document that contains the data feed.

4. Click **Save** to save the document to the file system, or click **Open** to view the data feed before saving.

5. Browse to the location to save the document.

6. Optionally, change the name of the document.

   > **NOTE**
   > By default the document name is the report name. If the Atom service document has multiple feeds, by default all use the same name, the report name. To differentiate them, rename them to use meaningful names.

7. Verify the document type is **ATOM File**, and then click **Save**.

8. Optionally, open the .atom file in a browser or text editor or XML editor.

See Also

Export Reports (Report Builder and SSRS)

Work Around the Excel 2003 Row Limitation

3/24/2017 • 1 min to read • Edit Online

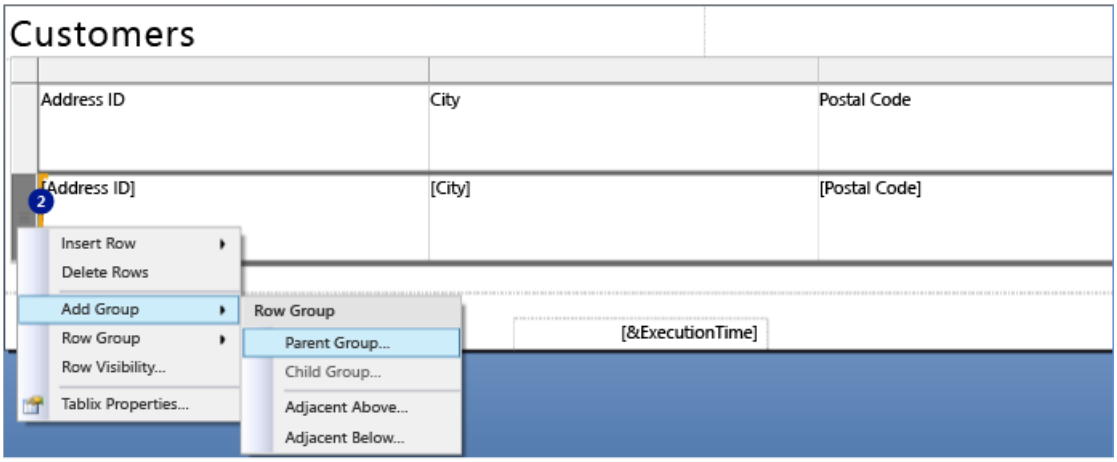
This topic explains how to work around the Excel 2003 row limitation when you export paginated reports to Excel. The workaround is for a report that contains only a table.

> **IMPORTANT**
> The Microsoft Excel 2003 (.xls) rendering extension is deprecated. For more information, see Deprecated Features in SQL Server Reporting Services in SQL Server 2016.

Excel 2003 supports a maximum of 65,536 rows per worksheet. You can work around this limitation by forcing an explicit page break after a certain number of rows. The Excel renderer creates a new worksheet for each explicit page break.

To create an explicit page break

1. Open the report in SQL Server Data Tools - Business Intelligence or the Reporting Services web portal.
2. Right click the Data row in the table, and then click **Add Group** > **Parent Group** to add an outer table group.



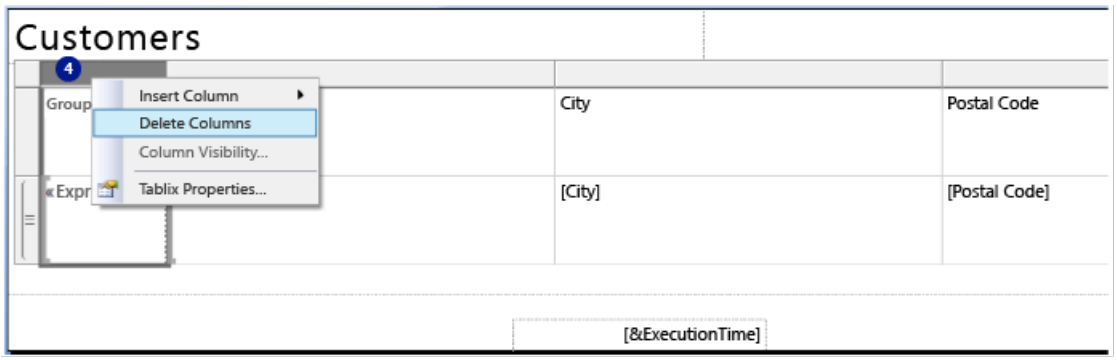
3. Enter the following formula in the **Group by** expression box, and then click **OK** to add the parent group.

   =Int((RowNumber(Nothing)-1)/65000)

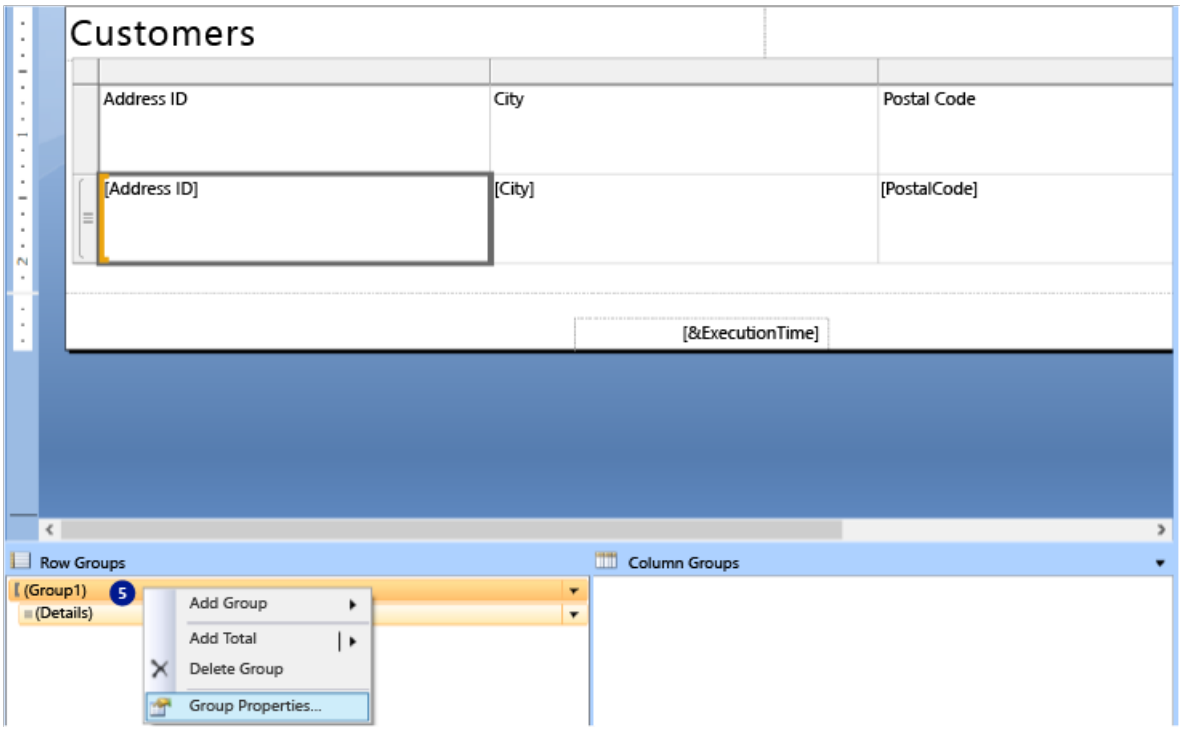
   The formula assigns a number to each set of 65000 rows in the dataset. When a page break is defined for the group, the expression results in a page break every 65000 rows.

   Adding the outer table group adds a group column to the report.

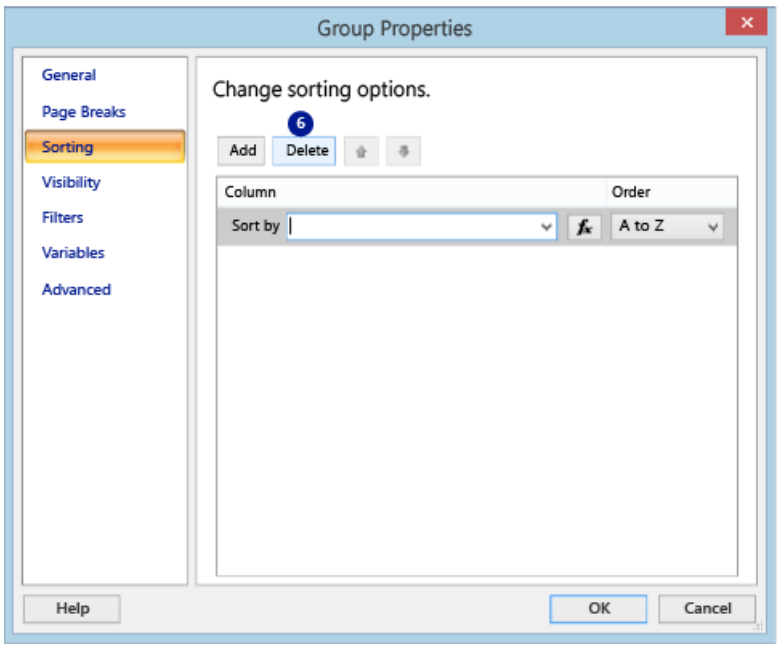
4. Delete the group column by right-clicking on the column header, clicking **Delete Columns**, selecting **Delete columns only**, and then click **OK**.



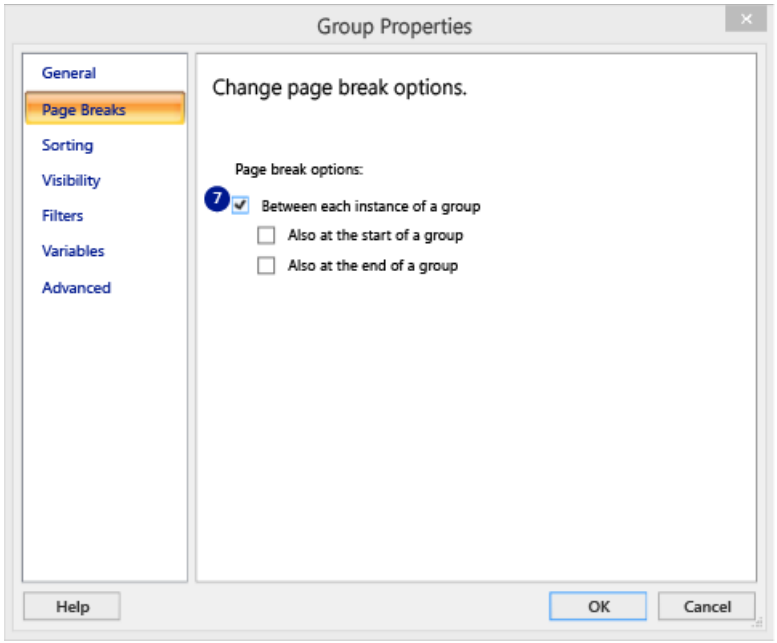
5. Right click **Group 1** in the **Row Groups** section, and then click **Group Properties**.



6. On the **Sorting** page of the **Group Properties** dialog box, select the default sorting option and click **Delete**.



7. On the **Page Breaks** page, click **Between each instance of a group** and then click **OK**.



8. Save the report. When you export it to Excel, it exports to multiple worksheets and each worksheet contains a maximum of 65000 rows.

Print Reports (Report Builder and SSRS)

3/24/2017 • 2 min to read • Edit Online

After you save a report to a report server, you can view and print the report from a browser, Report Manager, or any application that you use to view an exported report. Before saving a report, you can print it when you preview it.

All print processing is performed on demand and on the client computer. There is no server-side print functionality that enables you to route a print job directly from a report server to a printer that is attached to the Web server. Printers and print options are selected by individual report users by using a standard **Print** dialog box.

Report authors who design reports specifically for print output can use page breaks, page headers and footers, expressions, and background images to create a print-based design. Examples of report design elements intended for print output might include terms and conditions that you print on the back of every report, or graphic and text elements that mimic letterhead.

Due to the way pagination is implemented for different rendering formats, you might not be able to achieve optimum print output results for every report in every rendering format. The following list provides examples:

1. Report pages are designed to accommodate variable amounts of data. Reports that include a matrix, for example, can cause a page to grow both horizontally and vertically depending on whether a user interactively toggles rows and columns. A user who does not expand a matrix will get different print results than a user who does.
2. You cannot combine landscape and portrait mode pages in the same report, nor is there a way to create a print-based layout that replaces or exists alongside the layout of a report as rendered in a browser or other application.
3. For most exported reports, report printouts include everything that is visible on the report, as viewed by the user on a computer monitor. White space from the report design surface is preserved. To add or remove extra blank pages horizontally, change the report page width.

> **NOTE**
> HTML report printouts may contain only the content on the first page if you are using the browser's Print command. You can achieve better results if you print HTML reports using the Reporting Services client printing functionality. For more information, see Print Reports from a Browser with the Print Control (Report Builder and SSRS).

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

In This Section

Print Reports from a Browser with the Print Control (Report Builder and SSRS)
Describes how to use client-side printing to print reports from your Web browser or Report Manager.

Print Reports from Other Applications (Report Builder and SSRS)
Describes how to print reports exported to another application.

Print a Report (Report Builder and SSRS)

Provides step-by-step instructions on how to print a report, how to control the margins on a page, and on how to specify the paper size for reports that will be rendered by hard-page break renderers: PDF, Image, or Print.

See Also

Export Reports (Report Builder and SSRS)

Page Headers and Footers (Report Builder and SSRS)

Images (Report Builder and SSRS)

Pagination in Reporting Services (Report Builder and SSRS)

Print Reports from a Browser with the Print Control (Report Builder and SSRS)

3/24/2017 • 2 min to read • Edit Online

Although a browser is the most common client application used to view a report, browser print functionality is not ideal for printing reports. Print functionality in a browser is designed for printing Web pages. Typically, pages that you print from a browser include all of the visual elements that are on a Web page, plus header and footer information that identifies the page or Web site. Printing from a browser prints the contents of the current window. For a multipage report, the browser prints the first page at most, and possibly less if the report page extends beyond the dimensions of a printed page.

To improve the print quality of reports that you view in a browser and to print multiple pages, you can use the client-side print functionality provided in SQL Server Reporting Services. Client-side printing provides a standard **Print** dialog box that can be used to select a printer, specify pages and margins, and preview the report before you print. Client-side printing is intended to be used in place of the **Print** command on the browser's **File** menu. When you use client-side printing, the report is printed as it was designed, without the extra elements you see in a Web page print out.

To use client-side printing, you need to install a Microsoft ActiveX control. For more information, see Enable and Disable Client-Side Printing for Reporting Services.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

Print Options

To configure print properties for your report, in the **Print** dialog box, click the **Properties** button. **Paper size** is determined by the default height and width of the report page size as defined in the report definition. The available values are dependent on the printer type and its capabilities. Width and height display default values as determined by the print drivers that are configured on the computer. Changing these values causes the report to print using the new dimensions. Page width and height are each determined by **Orientation**, which is set to either **Portrait** or **Landscape**. The default orientation displayed is dependent on the page width and page height of the report.

> **NOTE**
> The **Print** dialog box and the default printer settings for width, height, and page orientation are determined by the report definition.

Print Preview

To preview a report, in the **Print** dialog box, click the **Preview** button. Clicking preview opens the first page of the report in a separate preview window. Additional pages are made available as the report is rendered on the report server. A previewed report is rendered in EMF format. You can navigate to the previous or next page until the last page is reached, and the **Next** button is disabled.

Adjusting Print Margins

You can modify the print margins in the rendered EMF report prior to printing the report. To do this, in the **Print** dialog box, click the **Preview** button. At the top of the preview page, click the **Margins** button. The Margins dialog box is displayed. Configure the top, bottom, right, and left margins as desired. Click **OK**. The dialog box closes and the settings are stored for rendering preview and printing.

See Also

Print Reports (Report Builder and SSRS)
Print a Report (Report Builder and SSRS)

Print Reports from Other Applications (Report Builder and SSRS)

3/24/2017 • 1 min to read • Edit Online

Report Builder provides an export option that allows you to easily view a report in other applications. The **Export** command is available on the report toolbar that appears at the top of a report when you open it in a browser or Web-based application. Exporting a report displays it in a different application (for example, exporting a report to Excel opens the report in Microsoft Excel). For printing purposes, exporting a report is recommended only if the application has specific printing features that you want to use.

To export a report to another application, you must have that application installed. For example, you must have Adobe Acrobat Reader installed on your computer before you can export to the Acrobat (PDF) format. If you choose to export a report to TIFF format, the report server places the report in a viewing application that is associated with the TIFF file type. Although the application used depends on which version of Microsoft Windows you have, typically this tool is Windows Picture and Fax Viewer. The default resolution corresponds to a screen resolution of 96 dots per inch (DPI). You can increase the resolution in Windows Picture and Fax Viewer to 300 DPI or 600 DPI to match the capabilities of your printer. For more information about adjusting the resolution, see the Windows product documentation.

If you choose Web archive (also known as MHTML), the report is exported to your default browser. Printing from the browser may result in report path information being added at the bottom of every page. In most cases, you can set browser options to omit path information on a printed page. For more information, see the product documentation for the browser you are using.

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

See Also

Print a Report (Report Builder and SSRS)
Print Reports from a Browser with the Print Control (Report Builder and SSRS)
Export Reports (Report Builder and SSRS)
Export a Report as Another File Type (Report Builder and SSRS)
Finding, Viewing, and Managing Reports (Report Builder and SSRS)

Print a Report (Report Builder and SSRS)

3/31/2017 • 3 min to read • Edit Online

After you save a report to a report server, you can view and print the report from a browser, the Reporting Services web portal, or any application that you use to view an exported report. Before saving a report, you can print it when you preview it.

When you print a report, you can specify the size of the paper to use. The size of the paper determines the number of pages in a report and which report data fits on each page. Paper size affects only reports that are rendered with hard page-break renders: PDF, Image, and Print. Setting the paper size has no effect on other renderers. For more information, see Rendering Behaviors (Report Builder and SSRS).

From the report viewer toolbar in the Reporting Services web portal or in preview in Report Builder, you can export a report to a hard page-break renderer or click the Print button to print a copy of the report. You might need to set the paper size or other page setup properties. Use the **Report Properties** dialog box to change page setup properties, including paper size.

You can specify print page margins in two different locations: in design mode and in run mode.

- **Design mode.** When you set page margins in design mode, these settings are saved in the report definition when you save the report.
- **Run mode.** When you set page margins in run mode, this information is not saved in the report definition. The next time you print the report, you will get the settings from the report definition, unless you indicate your print margins again.

  > **NOTE**
  > Print margins are not displayed in design or run modes. There is no relationship between the design surface area and the print area of your report. To see print margins, in run mode, click Print Layout on the **Run** tab on the Ribbon.

  For more information about report paging, see Pagination in Reporting Services (Report Builder and SSRS).

> **NOTE**
> You can create and modify paginated report definition (.rdl) files in Report Builder and in Report Designer in SQL Server Data Tools. Each authoring environment provides different ways to create, open, and save reports and related items.

To print a report in Report Builder

1. Open a report.
2. On the Home tab, click **Run**.
3. (optional) Click **Print Layout** to see how the report will look when it is printed.
4. (optional) Click **Page Setup** to set paper, orientation, and margins.

> **NOTE**
> The default values for these come from the report properties, which are set in Design view. The values you set here in the **Page Setup** dialog box are for this session only. When you close this report and reopen it, it will have the default values again.

5. Click **Print**.
6. In the **Print** dialog box, select a printer and specify other printing options.

To print a report from a Web browser application

1. In the Reporting Services web portal, navigate to the report that you want to print. Open the report.
2. On the toolbar at the top of the report, click **Print**.

> **NOTE**
> The first time you print an HTML report, the report server prompts you to install an ActiveX control used for printing. You must install and configure the control to print.

3. In the **Print** dialog box, select a printer, and then click **Prin**t.

To print a report from other applications

1. In the Reporting Services web portal, navigate to the report that you want to print. Open the report.
2. On the toolbar at the top of the report, select a rendering format, and then click **Export**. The report opens in a viewer application that corresponds to the rendering format.

   For example, if you select PDF, the report opens in Adobe Acrobat Reader.
3. On the **File** menu in that program, click **Print**.

To change paper size

1. Right-click outside of the report body and click **Report Properties**.
2. In **Page Setup**, select a value from the **Paper Size** list. Each option populates the **Width** and **Height** properties. You can also specify a custom size by typing numeric values in the **Width** and **Height** boxes. Click **OK**.

> **NOTE**
> Size values have a default unit based on the user's locale settings. To designate a different unit, type a physical unit designator such as cm, mm, pt, or pc after the numeric value.

To set page margins in design mode

- Right-click the blue area around the design surface, click **Report Properties**, and then click the **Page Setup** page.

To set page margins in run mode

- Click **Page Setup** on the **Run** tab.

See Also

Print Reports (Report Builder and SSRS)
Export Reports (Report Builder and SSRS)

Report Properties Dialog Box, Page Setup (Report Builder)
Report Design View (Report Builder)

Print a Report (Reporting Services in SharePoint Mode)

3/24/2017 • 3 min to read • Edit Online

For a report server that runs in SharePoint mode, there are three ways to print a report from a SharePoint Web application:

- **From a SharePoint site** Choose **Print** from the **Actions** menu that appears in the report toolbar when you open the report. This provides Reporting Services print functionality, which includes a standard **Print** dialog box used to select a printer, specify pages and margins, and preview the report. This print feature is intended to be used in place of the Print command on a browser's File menu. When you print reports this way, the report is printed as it was designed, without the extra elements you see in a Web page print out.
- **From a browser** The print features of a browser work best for HTML reports that fit on a single page. Typically, pages that you print from a browser include all of the visual elements that are on a Web page, plus header and footer information that identifies the page or Web site. When you print from a browser, only the contents of the current window are printed. If the report is long, the browser prints only a portion of the report (typically just the first page).
- **From a target application** You can export a report to use the print features of a target application, such as Microsoft Office Excel or Adobe Acrobat Reader. Some application formats, such as TIFF or PDF, are ideally suited for printing multipage reports. When you export a report to a desktop application, you can use any specialized print features that the application provides. To export a report, choose **Export** from the **Actions** menu that appears in the report toolbar when you open the report.

> **NOTE**
> To print a report, you must have permission to view it.

For best results when printing a report from a Web page, use **Print** on the **Actions** menu. The **Print** action is tied to a client print control that is downloaded from the report server. The download occurs once, the first time you select **Print**.

Report authors can design reports specifically for print output or for a specific application format. Recognize that due to the way pagination is implemented for different application formats, you may not be able to achieve optimum print output results for every report in every export format. In contrast with reports that are designed for print output, on-screen report pages are designed to accommodate variable amounts of data. For example, reports that include a matrix can cause a page to grow both horizontally and vertically depending on how you expand the rows and columns. When printing a variable sized report, a user who does not expand a matrix will get different print results than a user who does. For most exported reports, report printouts include everything that is visible on the report, as viewed by the user on a computer monitor.

How to print reports from the Actions menu

1. Open the report.
2. On the **Actions** menu, click **Print**. If you do not see the **Actions** menu, the report toolbar has been hidden and you cannot use the features it provides. If you have an **Actions** menu but **Print** is not on it, the print functionality has been disabled on the report server and you cannot use it.
3. In the **Print** dialog box, select the printer and settings you want to use and click **OK**.

To modify the default settings, click the **Properties** button. Page size is determined by the default height and width of the report page size as defined in the report definition. The extent to which you can change page dimensions depends on the capabilities of the printer you are using.

To view the report before you print it, click the **Preview** button. This opens the first page of the report in a separate preview window. Additional pages are made available as the report is rendered on the report server. A previewed report is rendered in EMF format. You can navigate to the previous or next page until the last page is reached, and the **Next** button is disabled. To modify the print margins in the preview page, click the **Margins** button. The **Margins** dialog box is displayed. Configure the top, bottom, right, and left margins and click **OK**. The dialog box closes and the settings are stored for rendering preview and printing.

See Also

Enable and Disable Client-Side Printing for Reporting Services

Saving Reports (Report Builder)

3/24/2017 • 3 min to read • Edit Online

In Report Builder you can save a paginated report to a Reporting Services report server, SharePoint library, file share where you have write permission, or your computer.

When you save a report, what you are really saving is the report definition, which describes the report layout. You are not saving the data. Every time you run the report, the report data is refreshed and is likely to be different than the previous time you ran the report.

If you want to save the report to a different format or save the report definition with the data, use one of the following Reporting Services features:

- Export a rendered report to a different file format such as comma separated files (CSV) or Excel workbooks and save the report in that format. You can also generate data feeds from reports and save the report data.
- Create report subscriptions to deliver and save reports to a file share.
- Use report history to save versions of rendered reports as historical copies.

  To learn more about viewing and managing reports directly on the report server, see Finding, Viewing, and Managing Reports (Report Builder and SSRS) and Reporting Services Report Server (Native Mode).

Saving Reports to a Report Server

Saving a report to a report server is also known as publishing a report. Although you can save reports to your computer, saving reports to a report server offers many advantages:

- Reports become available to others who have permission to access the folder in which you saved the report.
- Reports can be managed and viewed on the Reporting Services web portal.
- Report resources such as data sources, images, and subreports are stored in one place for easier access.
- Reports can be delivered to others by subscriptions.
- Reports are securely stored in the report server database.
- Report runs can be logged and provide performance and auditing information.

Exporting and Saving Reports

If you have a small number of reports to archive, consider exporting a report and saving it as a file. After you export a report to an application (such as PDF or Excel), you can save it as a file and place it in a protected shared directory on the network. Alternatively, you can upload a saved PDF or Excel file as a resource item if you want to keep all copies of a report, regardless of the format, in the report server database. For more information about exporting a report, see Export Reports (Report Builder and SSRS) and Upload a File or Report.

Using File-Share Delivery

If you have a large number of reports to archive, create a subscription that delivers the report directly to the file system. For this approach, you must create a subscription for each report, choose a shared folder to store the reports, and define a schedule that determines when the file is created. Once you define a subscription, the report

server can run the report unattended and add report files to the archive using the schedule that you provide. You can also create single-use schedules if you want to archive reports on an occasional basis. For more information about subscriptions and file share delivery, see File Share Delivery in Reporting Services.

Using Report History

You can also use the report history feature to create historical copies. You can then back up the report server database and store the backup in a safe location for future use. All report history (along with reports, shared data source items, folders, subscriptions, and shared schedules) is stored in the report server database. You can create a backup to maintain a permanent copy of report history and metadata such as subscription information that indicates the recipients of a report. For more information, see Create, Modify, and Delete Snapshots in Report History.

How-To Topics

- Save Reports to a Report Server (Report Builder)
- Save a Report to a SharePoint Library (Report Builder)

See Also

Reports, Report Parts, and Report Definitions (Report Builder and SSRS)
Install and Uninstall Report Builder
Finding, Viewing, and Managing Reports (Report Builder and SSRS)
Export Reports (Report Builder and SSRS)
Print Reports (Report Builder and SSRS)

Save Reports to a Report Server (Report Builder)

3/24/2017 • 1 min to read • Edit Online

In Report Builder, you can save a report definition to a report server (also known as publishing a report). When the report is saved to a report server, other users can view the report. Each time you run the published report, you will retrieve the most current data. To save a static copy of a rendered report, export the report to a different file format and save it or use the report history feature to save versions of rendered reports.

> **NOTE**
> The location of the saved report definition does not affect whether the report is processed on the server or processed locally when you preview the report.

To save a report to a report server

1. From the Report Builder button, click **Save**. The **Save As**<*Report Item*> dialog box opens.

   > **NOTE**
   > If you are resaving a report, it is automatically resaved to its previous location. Use the Save As option to change location.

2. Optionally, click **Recent Sites and Servers** to show a list of recently used report servers and SharePoint sites.
3. Browse to the report server location where you want to save the report.
4. In **Name**, type the name of the report.
5. In **Items of type**, select the type of report item you are saving. The type for reports is Reports(*.rdl).

To save a report as a different name

1. From the Report Builder button, click **Save As**. The **Save As**<*Report Item*> dialog box opens.
2. Browse to the report server location or to the file share where you want to save the report.
3. In **Name**, type the name of the report.
4. In **Items of type**, select the type of report item you are saving. The type for reports is Reports(*.rdl).

See Also

Finding, Viewing, and Managing Reports (Report Builder and SSRS)
Export Reports (Report Builder and SSRS)
Saving Reports (Report Builder)
Export a Report as Another File Type (Report Builder and SSRS)

Save a Report to a SharePoint Library (Report Builder)

3/24/2017 • 1 min to read • Edit Online

To save a report to a report server configured for SharePoint integration, you must browse to the SharePoint server and establish a connection to the report server. In the report definition, all references to items related to the report must use values that are specific to a SharePoint report server. Related items include subreports, drillthrough reports, and resources such as Web-based images. For more information, see Specifying Paths to External Items (Report Builder and SSRS).

You must have **Member** or **Owner** permission on the SharePoint site to set the properties on the project.

To save a report to a SharePoint site

1. From the Report Builder button, click **Save**. The **Save As**<*Report Item*> dialog box opens.

   > **NOTE**
   > If you are resaving a report, it is automatically resaved to its previous location. Use the **Save As** option to change location.

2. Optionally, click **Recent Sites and Servers** to show a list of recently used report servers and SharePoint sites.

3. Browse to the SharePoint site, and then click **Save**.

   > **NOTE**
   > If you leave a changed report for more than 10 hours without saving it, it is disconnected from the server without being saved. If that happens, in the lower-right status bar, click **Disconnect**, and then click **Connect**. The most recent server will be in the list of available servers. Select it and the report will reconnect.

See Also

Finding, Viewing, and Managing Reports (Report Builder and SSRS)

Security (Report Builder)

3/24/2017 • 6 min to read • Edit Online

Report Builder is a report authoring client application that is designed to work with a SQL Server Reporting Services report server. The report server can be configured to work in native mode as a stand-alone server or in SharePoint integrated mode to support reports on a SharePoint site.

In Report Builder, you can author reports, shared datasets, and reusable report parts. From a report server or SharePoint site, you can edit reports and add shared data sources, shared datasets, and shared report parts.

To author, publish, and use reports and report-related items, you should understand how security features relate to the following areas:

- **The report server or SharePoint site where you publish reports** These features are managed by the report server administrator or SharePoint site administrator.
- **Published reports and report-related items** Report-related items include embedded and shared data sources and their credentials, shared datasets, parameters, report parts, and report models. Security features that apply to these items are managed by the report author. The report author must be granted sufficient permissions by the report server administrator or SharePoint site administrator to publish and share the items.
- **External data sources that are used by a report** These features are managed by the owner of the external data source.
- **Report models that are based on external data sources** These features are managed by the model designer.
- **Interactive report features such as parameters** These features are managed by the report author.

  Review the information in this topic to better understand how to use security features to help manage and secure reports and report-related items.

Understanding Security for Report Servers

Publishing reports and viewing reports are privileged operations. A report server administrator grants permissions to ensure that only authorized users can publish and view reports on one of the following types of report servers:

- Report server configured in native mode

  To connect to or browse to a report server, you must have a valid URL and have sufficient permissions to access the server.

  To view or publish items on a report server, sets of permissions that apply to report-related items and operations are organized into roles. A report server administrator assigns you to one or more roles. For example, the predefined role Browser enables you to view reports, folders, models, and resources.

  If you cannot connect to or browse to a report server, contact the report server administrator. For more information, see Reporting Services Security and Protection in the Reporting Services documentation in SQL Server Books Online.

- Report server configured in SharePoint integrated mode

  To connect to a SharePoint site that is integrated with a report server, you must have a valid URL to the SharePoint site or subsite and have sufficient permissions to access it.

Permission to access report-related items and operations is granted through SharePoint security policies that map a user or group account with a permission level, relative to an item.

If you cannot connect to or browse to a SharePoint site or subsite, contact the SharePoint site administrator.

Understanding Security for Published Reports and Report-related Items

Security for reports and report-related items is managed by the report server administrator. Report-related items include embedded and shared data sources including credentials, shared datasets, parameters, report parts, and models.

On a report server or SharePoint site, reports and report-related items and operations are independently securable. Permission to access items and operations is granted through security policies that map a user or group account with a permission level, relative to an item. To reduce the complexity and overhead of maintaining a large number of policies, permissions on a container, such as a folder, are inherited by items in the container. For example, if a user has the specific View Reports permission on a folder, they have View Reports permission on the items in the folder.

Permissions can be overridden on items or folders by using item level security. When item-level security is applied, permission inheritance from the parent container no longer applies to the item. If item-level security is applied to a folder, nested folders inherit the same permissions.

If you are not able to browse to and find items that someone else has published for you, you might have a permissions issue on the item or on the folder.

To enable others to browse to and find items that you published to be shared, you must work with the report server administrator to set up a folder organization that provides access to your users. Access must be available for authoring reports and for running published reports.

For more information, see the following topics in the Reporting Services documentation in SQL Server Books Online:

- Roles and Permissions (Reporting Services)
- Manage Shared Datasets

Update Notifications for Report Parts

Report parts are published to a report server so that others can share them. By design, you specify the location to publish report parts to.

Users who include report parts in their reports can enable the update feature. When this feature is enabled, users receive notifications when report parts change on the report server.

If report parts are moved from the original location, the update notice includes both the current location and the previous location of the report part. Accept updates only from trusted locations.

For more information, see Report Parts (Report Builder and SSRS).

Understanding Security for Report Data and External Data Sources

To access data from each external data source in a report, you create an embedded data source or add a reference to a shared data source or shared dataset in your report.

For each external data source, you must supply credentials that are sufficient to access the source and the underlying data. The data source owner specifies the type of credentials that provides this access.

Credentials are not saved in the report definition. They are managed independently from the report on the report server or SharePoint site and on the report authoring client.

At report design time, credentials are used to run dataset queries and preview the report. At run time, credentials are used to run the report and cache query results. You can also cache shared dataset query results independently. Design time and run time credentials might differ. For more information, see Specify Credentials in Report Builder.

For more information about securing data, see the following topic in the Reporting Services documentation in SQL Server Books Online:

- Security Center for SQL Server Database Engine and Azure SQL Database

  For more information about data sources, see Data Connections, Data Sources, and Connection Strings in Report Builder.

Understanding Models and Security Filters

When data is retrieved from a report model that is based on external data, you can apply security filters in the model This is a good way to secure data so that each user who runs a report can see only the data that they have permissions to.

Report parameters are not used for row-level security; they do not prevent users or groups of users from seeing specific rows of data. To apply security to the data displayed within a report, you must use security filters or model item security.

Understanding Security for Report Authoring for Interactive Features

Reports frequently use parameters to enable a user to interactively customize their view of a report. Use the following tips to help design reports that follow good practices:

- Do not use parameters that are based on query parameters and that are type **Text** unless you provide valid values. An available values list helps a user choose only valid values. Without an available values list, you cannot restrict which values a user can enter.
- Do not use the the global [&UserID] to secure private data. As a report parameter, this value can be specified in a report URL by using URL access syntax. Using this value in an expression in a shared dataset prevents the dataset from being cached. For more information, see URL Access Parameter Reference in the Reporting Services documentation in SQL Server Books Online.

  After items are published to a report server, the report server administrator can help secure them by assigning role-based security or folder and item level security. For more information, see Secure Reports and Resources in the Reporting Services documentation in SQL Server Books Online.

See Also

Install and Uninstall Report Builder
Report Parameters (Report Builder and Report Designer)

Glossary (Report Builder)

3/24/2017 • 42 min to read • Edit Online

TERM	DEFINITION
.NET Framework	An integral Windows component that supports building, deploying, and running the next generation of applications and Web services. It provides a highly productive, standards-based, multilanguage environment for integrating existing investments with next generation applications and services, as well as the agility to solve the challenges of deployment and operation of Internet-scale applications. The .NET Framework consists of three main parts: the common language runtime, a hierarchical set of unified class libraries, and a componentized version of ASP called ASP.NET. See also: ASP.NET, common language runtime, .NET Framework class library.
ActiveX Data Objects	Component Object Model objects that provide access to data sources. This API provides a layer between OLE DB and programming languages such as Visual Basic, Visual Basic for Applications, Active Server Pages, and Microsoft Internet Explorer Visual Basic Scripting.
ad hoc report	An .rdl report created with Report Builder 1.0 that accesses report models.
adornment	A control or status area that is attached to the edge of a pane or window, such as a toolbar or ruler.
aggregate	A summary value comprised of multiple individual values.
aggregate function	A function that performs a calculation on multiple values and returns a single summary value.
aggregate of aggregates	A summary value calculated from aggregates, such as the maximum of a set of sums.
aggregate query	A query that summarizes information from multiple rows by including an aggregate function such as Sum or Avg.
aggregation	A table or structure that contains precalculated data for a cube.
alias	An alternative name for a table or column in expressions that is often used to shorten the name for subsequent reference in code, prevent possible ambiguous references, or provide a more descriptive name in the query output.
analytical data	In a report, a set of numeric values that can be analyzed. In a report map, analytical values control color, marker type, or size of map elements on each map layer.

TERM	DEFINITION
antialiasing	In graphics, using an algorithm to smoothing out a jagged approximation of a curved line.
application database	The database that stores user and system data for one application.
argument	A value that provides information to an action, an event, a method, a property, a function, or a procedure.
assembly	A managed application module containing class metadata and managed code as an object in SQL Server, against which CLR functions, stored procedures, triggers, user-defined aggregates, and user-defined types can be created in SQL Server.
authentication	The process of verifying the identity of a user, computer, process, or other entity by validating the credentials provided by the entity. Common forms of credentials are digital signatures, smart cards, biometric data, and a combination of user names and passwords.
authorization	1. The operation that verifies the permissions and access rights that are granted to a user. 2. The process of granting a person, computer process, or device access to certain information, services or functionality.
axis	The vertical and horizontal lines on a graph used to show the position of a point.
axis interval	The number of units between major tick marks on a chart axis. The interval is automatically calculated, but can be manually set.
back up	To make a duplicate copy of a program, a disk, or data.
backup	A collection of files, folders, and other data that have been duplicated and stored in a file or on one or more tapes.
base table	A table stored permanently in a database. Base tables are referenced by views, cursors, SQL statements, and stored procedures.
batch	1. A set of SQL statements submitted together and run as a group. 2. In Reporting Services, a collection of SOAP method calls within a single transaction.
binary large object (BLOB)	A piece of binary data that has an exceptionally large size, such as images, audio, or multimedia tracks that are stored as digital data, or any variable or table column that is large enough to hold such values.
bitwise operation	An operation that manipulates a single bit, or tests whether a bit is on or off.

TERM	DEFINITION
bookmark	A link in a report that jumps to another place within the report.
Boolean	An operation or expression that can be evaluated only as either true or false.
bubble map	A map layer that displays variable size circles. Bubbles center on polygon center points or on points.
built-in functions	A group of predefined functions provided as part of the Transact-SQL and Multidimensional Expressions languages.
calculated column	A column in a table that displays the result of an expression instead of stored data.
calculated field	A field, defined in a query, that displays the result of an expression instead of stored data.
calculated member	A member of a dimension whose value is calculated at run time by using an expression.
cardinality	Defines the number of instances allowed and/or required of a child entity per parent entity.
cascading parameters	Parameters that derive their available values from the selection of a preceding parameter. Cascading parameters are used to filter a set of parameter values.
category (x) axis	The axis for grouping data in a chart, usually the horizontal axis. Exception: in bar charts, the axes are reversed and the y axis displays grouping data.
certificate	A digital document that is commonly used for authentication and to help secure information on a network. A certificate binds a public key to an entity that holds the corresponding private key. Certificates are digitally signed by the certification authority that issues them, and they can be issued for a user, a computer, or a service.
character set	The types of characters that SQL Server recognizes in the char, varchar, and text data types.
chart data region	A report item on a report layout that displays data in a graphical format.
child	A member in the next lower level in a hierarchy that is directly related to the current member.
clause	In Transact-SQL, a subunit of an SQL statement. A clause begins with a keyword.
clickthrough report	A report that displays related report model data when you click data within a rendered Report Builder report.

TERM	DEFINITION
client application	An application that retrieves data from a server and performs local analysis and presentation of data from relational or multidimensional databases.
client cursor	A cursor that is implemented on the client. The entire result set is first transferred to the client, and the client API software implements the cursor functionality from this cached result set.
CLR	See Other Term: common language runtime
CLR function	A function created against a SQL Server assembly whose implementation is defined in an assembly created in the .NET Framework common language runtime.
CLR stored procedure	A stored procedure created against a SQL Server assembly whose implementation is defined in an assembly created in the .NET Framework common language runtime.
CLR trigger	A trigger created against a SQL Server assembly whose implementation is defined in an assembly created in the .NET Framework common language runtime.
CLR user-defined type	A user-defined data type created against a SQL Server assembly whose implementation is defined in an assembly created in the .NET Framework common language runtime.
code access security	A mechanism provided by the common language runtime whereby managed code is granted permissions by security policy and these permissions are enforced, helping to limit the operations that the code will be allowed to perform.
collation	A set of rules that determines how data is compared, ordered, and presented.
collection	1. An object that contains a set of other objects. 2. In Master Data Services, a hierarchical grouping of members from explicit hierarchies and other collections.
color rules	The algorithm that automatically assigns colors to each map element on a map layer.
color scale	A type of map legend that is used to interpret what colors represent on a map.
common language runtime	The engine at the core of managed code execution. The runtime supplies managed code with services such as cross-language integration, code access security, object lifetime management, and debugging and profiling support.
complex expression	An expression that cannot be represented as a single built-in field reference, and is displayed on the design surface as <<Expr>>.

TERM	DEFINITION
concatenation	The combining of two or more character strings or expressions into a single character string or expression, or to combine two or more binary strings or expressions into a single binary string or expression.
concurrency	A process that allows multiple users to access and change shared data at the same time. SQL Server uses locking to allow multiple users to access and change shared data at the same time without conflicting with each other.
config file	See other term: configuration file
configuration	In reference to a single microcomputer, the sum of a system's internal and external components, including memory, disk drives, keyboard, video, and generally less critical add-on hardware, such as a mouse, modem, or printer.
configuration file	A file that contains machine-readable operating specifications for a piece of hardware or software, or that contains information about another file or about a specific user.
containment scope	The scope in which an expression is evaluated. Scope is determined by the report processor as data and layout elements are combined.
credential	Data used by a principal to establish the identity of the principal, such as a password or user name.
cube	A set of data that is organized and summarized into a multidimensional structure defined by a set of dimensions and measures.
cursor	An entity that maps over a result set and establishes a position on a single row within the result set.
data connection	A collection of information that is required to access a specific data source. The collection includes a data source name and logon information.
data extension	A component in Reporting Services that is used to retrieve report data from an external data source.
data feed	A mechanism for users to receive a stream of data from data sources. In Reporting Services, the Atom rendering extension generates data feeds in Atom format from reports.
data hierarchy	The organization of data that is processed by the report processor: dataset, data region, data region group, details group.
data mining	The process of analyzing data to identify patterns or relationships.
data point	The smallest individual entity in a chart.

TERM	DEFINITION
data processing extension	A component in Reporting Services that is used to retrieve report data from an external data source.
data provider	The layer of sofware that handles communication between data extensions and customized software specific to each type of external data source. Depending on the specific data source, multiple data providers are available from Microsoft and from third-party vendors.
data region	A report item that displays repeated rows of data from an underlying dataset in a table, matrix, list, or chart.
data source	1. In ADO and OLE DB, the location of a source of data exposed by an OLE DB provider. 2. The source of data for an object such as a cube or dimension. It is also the specification of the information necessary to access source data. It sometimes refers to object of ClassType clsDataSource. 3. In Reporting Services, a specified data source type, connection string, and credentials, which can be saved separately to a report server and shared among report projects or embedded in a report definition (.rdl) file.
data source name	In a report, the name of the data source object.
data type	An attribute that specifies what type of information can be stored in a column, parameter, or variable.
data warehouse	A database specifically structured for query and analysis.
database	A collection of information, tables, and other objects organized and presented to serve a specific purpose, such as searching, sorting, and recombining data.
database catalog	The part of a database that contains the definition of all the objects in the database, as well as the definition of the database.
database engine	The program module or modules that provide access to a database management system (DBMS).
database file	One of the physical files that make up a database.
database language	The language used for accessing, querying, updating, and managing data in relational database systems.
database object	A database component in a database. Can also refer to the database itself.
database owner	The member of the database administrator role of a database.
database project	A collection of one or more data connections (a database and the information needed to access that database).
database role	A collection of users and groups with the same access to an Analysis Services database.

TERM	DEFINITION
database schema	The names of tables, fields, data types, and primary and foreign keys of a database. Also known as the database structure.
database script	A collection of statements used to create database objects.
data-bound image	An image that exists in a database, such as a field in a database that contains images.
data-driven subscription	A subscription in Reporting Services that uses a query to retrieve subscription data from an external data source at run time.
dataset	1. In OLE DB for OLAP, the set of multidimensional data that is the result of running a Multidimensional Expressions (MDX) SELECT statement. 2. In Reporting Services, a named specification that includes a data source definition, a query definition and optional parameter values, calculated fields, and filtering and collation information as part of a report definition (.rdl) file. An .rdl file can have multiple datasets.
DBCS	double-byte character set
default	1. A data value, option setting, collation, or name assigned automatically by the system if a user does not specify the value, setting, collation, or name. 2. An action taken automatically at certain events if a user has not specified the action to take.
default database	The database the user is connected to immediately after logging in to SQL Server.
default instance	The instance of SQL Server that uses the same name as the computer name on which it is installed.
delimiter	1. In Transact-SQL, characters that indicate the start and end of an object name, by using either double quotation marks ("") or brackets ([]). 2. In Integration Services and Reporting Services, characters that are used to separate records, fields, or strings.
delivery extension	A component in Reporting Services that is used to distribute a report to specific devices or target locations.
delivery protocol	The set of communication rules used to route notification messages to external delivery systems.
dependent item	On a report server, a report catalog item to which other items contain a reference. For example, when a report definition contains a reference to a shared data source, the data source is a dependent item.

TERM	DEFINITION
distance scale	A type of map legend that displays distance units for the current resolution of the map data.
distribution	In a map layer, the algorithm to use to divide analytical values into subranges. Range intervals can be calculated to provide equally spaced intervals between the start and end value (EqualInterval), or be set to values so that an equal number of items is in each range interval (EqualDistribution).
document map	A navigation pane in a report arranged in a hierarchy of links to report sections and groups.
domain	1. In Windows security, a collection of computers grouped for viewing and administrative purposes that share a common security database. 2. In relational databases, the set of valid values allowed in a column.
drill down/drill up	To navigate through levels of data ranging from the most summarized (up) to the most detailed (down).
drill through	1. In Analysis Services, to retrieve the detailed data from which the data in a cube cell was summarized. 2. In Reporting Services, to open related reports by clicking hyperlinks in the main drillthrough report.
drilldown/drillup	A technique for navigating through levels of data ranging from the most summarized (up) to the most detailed (down). Created by adding toggles (plus and minus signs) to show and hide levels of the hierarchy of a table or matrix.
drillthrough	1. In Analysis Services, a technique to retrieve the detailed data from which the data in a cube cell was summarized. 2. In Reporting Services, a way to open related reports by clicking hyperlinks in the main drillthrough report.
drillthrough report	A drillthrough report is the report that opens as the result of a drillthrough action on a report item in another report. The drillthrough report is usually related to the main, or summary, report through data. A common example of a drillthrough report might be a Monthly Sales report that contains links to individual sales orders for that month.
dynamic connection string	An expression that you build into the report, allowing the user to select which data source to use at run time. You must build the expression and data source selection list into the report when you create it in Report Designer.
edit session	A temporary container that is created on the report server and used by a client application such as Report Builder. The edit session enables the reuse of report data across multiple previews of reports in Report Builder.
embedded data source	A data source definition that is included in a report definition file. Embedded data sources are report-specific.

TERM	DEFINITION
embedded data sources	An embedded data source is an XML element that represents a data connection and is included in and used by a single report definition.
embedded dataset	A report dataset and is included in a single report definition or a single report part definition.
encryption	A method for keeping sensitive information confidential by changing data into an unreadable form.
entity	1. In Reporting Services, an entity is a logical collection of model items, including source fields, roles, folders, and expressions, presented in familiar business terms. 2. In Master Data Services, an entity is an object within a model. Each entity contains related members that are defined by attributes.
enumeration	A fixed set of integer values or string constants that may be used to specify the value of certain properties.
equirectangular projection	In a map report item, a very simple equidistant cylindrical projection in which the horizontal coordinate is the longitude and the vertical coordinate is the latitude.
error log	1. In SQL Server, a text file that records system information. 2. In Integration Services, a record of errors, warnings or events, which can be stored by multiple log providers, including SQL Server Profiler, the Windows Event log, and SQL Server.
error state number	A number associated with SQL Server messages that helps Microsoft support engineers find the specific code location that issued the message.
escape character	A character used to indicate that another character in an expression is meant literally and not as an operator.
ESRI .shp file	Open specification developed by Environmental Systems Research Institute, Inc. (ESRI) for spatial data.
ESRI Shapefile	Open specification developed by Environmental Systems Research Institute, Inc. (ESRI) for spatial data.
event	Any significant occurrence in the system or an application that requires users to be notified or an entry to be added to a log.
event handler	1. A software routine that executes in response to an event. 2. In Integration Services, a control flow that runs in response to a run-time event.
expression	1. In SQL, a combination of symbols and operators that evaluate to a single data value. 2. In Integration Services, a combination of literals, constants, functions, and operators that evaluate to a single data value.

TERM	DEFINITION
field	1. An area in a window or record that stores a single data value. 2. In Report Builder, dataset fields represent either numeric or non-numeric data from a data connection. For example, sales amounts, total sales, customer names, database identifiers, URLs, images, and spatial data. A report can have three types of fields: dataset fields, dataset calculated fields, and built-in fields.
filter	1. A set of criteria that controls the set of records returned as a result set. 2. In Full-Text Search, given a specified file extension, filters extract text from a file stored in a varbinary(max) or image column.
filter expression	An expression used for filtering data in the Filter operator.
flat file	A file consisting of records of a single record type, in which there is no embedded structure information governing relationships between records.
foreign key	The column or combination of columns whose values match the primary key (PK) or unique key in the same or another table. Also referred to as the referencing key.
foreign table	A table that contains a foreign key.
function	A piece of code that operates as a single logical unit. A function is called by name, accepts optional input parameters, and returns a status and optional output parameters.
gauge	A gauge is a data region that presents data as a range with an indicator pointing to a specific value within the range.
gauge panel	The gauge panel is the encompassing outer element that contains one or more radial and linear gauges. Use the gauge panel to define properties that apply for all gauges in a gauge panel.
gauge pointer	A gauge pointer is a needle, marker, or bar that indicates the current value on the gauge. A gauge pointer is always attached to a gauge scale.
gauge range	A gauge range is a colored area that represents a "critical zone," always bound by a start value and an end value, on the gauge scale.
gauge scale	A gauge scale is the range of values, bound by a minimum and a maximum, shown on the gauge. Typically, a gauge scale contains gauge labels and gauge tick marks to give accurate readings of what the gauge pointer is showing.
geodesic data	Data that specifies points, lines, or areas on a curved surface.
geography data	A SQL Server spatial data type that represents data in a round-earth coordinate system, such as GPS latitude and longitude coordinates.

TERM	DEFINITION
geometry data	A SQL Server spatial data type that supports planar, or Euclidean (flat-earth), data.
global assembly cache	A machine-wide code cache that stores assemblies specifically installed to be shared by many applications on the computer.
grid	A view type that displays data in a table.
grouping	A set of data that is grouped together in a report.
heat map	A type of map presentation where the intensity of color for each polygon corresponds to the related analytical data. For example, low values in a range appear as blue (cold) and high values as red (hot).
hierarchy	A logical tree structure that organizes the members of a dimension such that each member has one parent member and zero or more child members.
high availability	Status of a resource that remains operational and usable by clients most of the time with a very low percentage of failures that interrupt service.
HTML	See Other Term: Hypertext Markup Language
HTML Viewer	A user interface consisting of a report toolbar and other navigation elements used to work with a report.
Hypertext Markup Language	A text markup language used to create documents for the Web. HTML defines the structure and layout of a web document by using a variety of tags and attributes
identifier	The name of an object in a database.
identity column	A column in a table that has been assigned the identity property.
index	In a relational database, a database object that provides fast access to data in the rows of a table, based on key values.
inner aggregate	An aggregate function that is specified as a scope parameter by another aggregate. For example, in =Max(Sum([Quantity]),"Tablix1"), Sum is the inner aggregate.
inner join	An operation that retrieves rows from multiple source tables by comparing the values from columns shared between the source tables. An inner join excludes rows from a source table that have no matching rows in the other source tables.
instance	1. A copy of SQL Server running on a computer. 2. A specific copy of a report item, such as a report part, subreport, dynamic member, or group.

TERM	DEFINITION
integer	A numeric data type category that includes the bigint, int, smallint, and tinyint data types.
Internet Protocol security	Rules that computers follow to provide private and secure communication over Internet Protocol (IP) networks, through the use of cryptographic security services.
IPsec	See Other Term: Internet Protocol security
ISO	One of two international standards bodies responsible for developing international data communications standards. International Organization for Standardization (ISO) works closely with the International Electrotechnical Commission (IEC) to define standards of computing. They jointly published the ISO/IEC SQL-92 standard for SQL.
item-level role assignment	A security policy that applies to an item in the report server folder.
item-level role definition	a security template that defines a role used to control access to or interaction with an item in the report server folder namespace.
key performance indicator	(KPI) A quantifiable, standardized metric that reflects a critical business variable (for instance, market share), measured over time.
keyword	A reserved word in SQL Server that performs a specific function, such as to define, manipulate, or access database objects. For example: SELECT, FROM, and AND.
KPI	See Other Term: key performance indicator
label field	A report dataset field that identifies a label for a corresponding key field. Parameters and data regions can display the label for a key field instead of the value.
latency	The amount of time that elapses when a data change is completed at one server and when that change appears at another server.
latitude	The latitude coordinate as a decimal degree value in World Geodetic System (WGS 84) datum. Valid range is from –90.0 through +90.0.
layer	In a map report item, a layer specifies type and presentation of a single type of spatial data. A map report item is a container for multiple layers.
LCID	A number that identifies a Windows-based locale.
legend	In a report chart, map, or gauge data region, an element that provides a guide between the display presentation of data and the underlying data values.

TERM	DEFINITION
line layer	In a map report item, a layer that displays spatial data as lines, for example, for paths or routes.
linked report	A report that references an existing report definition by using a different set of parameter values or properties. A linked report is a report server item that provides an access point to an existing report. Conceptually, it is similar to a program shortcut that you use to run a program or open a file.
list data region	A data region on a report layout that repeats with each group or row in the report dataset. A list can be used to create free-form reports or forms, such as invoices, or in conjunction with other data regions.
local server	1. In SQL Server connections, an instance of SQL Server running on the same computer as the application. 2. In a Transact-SQL statement, when resolving references to database objects, the instance of SQL Server executing the statement. 3. In SQL Server distributed queries, the instance of SQL Server executing the distributed query. The local server then accesses any linked servers referenced in the query.
locale	The Windows operating system attribute that defines certain behaviors related to language.
locale identifier (LCID)	A number that identifies a Windows-based locale.
log file	A file or set of files that contain records of the modifications made in a database.
logical name	A name used by SQL Server to identify a file.
logical operators	The operators AND, OR, and NOT. Used to connect search conditions in WHERE clauses.
longitude	A coordinate as a decimal degree value in World Geodetic System (WGS 84) datum. Valid range is from –180.0 through +180.0.
many-to-many relationship	A relationship between two tables in which rows in each table have multiple matching rows in the related table. For example, each sales invoice can contain multiple products, but each product can appear on multiple sales invoices.
many-to-one relationship	A relationship between two tables in which one row in one table can relate to many rows in another table.
map	A report item that is a container for titles, various types of legends, and map layers.
map data	Map data consists of polygons, lines, points, and Bing map tiles. Map data can be embedded in a report or imported from ESRI shapefiles (.shp), SQL Server spatial data queries, or a Web service that returns images as Microsoft Bing map tiles.

TERM	DEFINITION
map viewport	The area of the map to display in the map report item. For example, a map for the entire United States might be embedded in a report, but only the area for the northwestern states are displayed.
marker	The symbol that is displayed on a map point layer at each point location.
master database	The system database that records all the system-level information for an instance of SQL Server.
master file	The file installed with earlier versions of SQL Server used to store the master, model, and tempdb system databases and transaction logs and the pubs sample database and transaction log.
match fields	A set of fields that are used to build a relationship between analytical data and spatial data.
matrix data region	A data region on a report layout that provides functionality similar to crosstabs and PivotTable reports. At run time, as the report data and data regions are combined, a matrix grows horizontally and vertically on the page. Values in matrix cells display aggregate values scoped to the intersection of the row and column groups to which the cell belongs.
MDX	See Other Term: Multidimensional Expressions (MDX)
Mercator projection	In a map report item, a cylindrical map projection devised by Gerardus Mercator in 1569.
metadata	Information about the properties of data, such as the type of data in a column (numeric, text, and so on) or the length of a column; information about a file, such as its title, description, date created, and date last modified; or information about the structure of data or information that specifies the design of objects such as cubes or dimensions.
method	A function that performs an action by using a COM object, as in SQL-DMO, OLE DB, and ActiveX Data Objects (ADO).
MIME type	The file format of an image, such as .bmp, .gif, or .jpeg.
model dependency	A relationship between two or more models in which one model is dependent on the information of another model.
Multidimensional Expressions (MDX)	A syntax used for defining multidimensional objects and querying and manipulating multidimensional data.
multidimensional OLAP	A storage mode that uses a proprietary multidimensional structure to store a partition's facts and aggregations or a dimension.
multiple instances	Multiple copies of SQL Server running on the same computer.

TERM	DEFINITION
named instance	An installation of SQL Server that is given a name to differentiate it from other named instances and from the default instance on the same computer.
named scope	In a report expression the calls an aggregate function, a value for the scope parameter that is the name of a dataset, data region, or data region group.
nested aggregates	An expression that contains an aggregate function that calls another aggregate function.
nested query	A SELECT statement that contains one or more subqueries, or another term for subquery.
nested table	A data mining model configuration in which a column of a table contains a table.
nesting	Placing one data region inside another data region, such as a sparkline inside a table. Nested data regions are based on the same report dataset, and the scope for data in the nested data region is automatically defined by its placement in the parent data region.
NULL	An entry that has no explicitly assigned value.
numeric expression	Any expression that evaluates to a number.
object	One of the components of a database, such as a table, index, or stored procedure.
object identifier	1. A unique name given to an object. 2. In Metadata Services, a unique identifier constructed from a globally unique identifier (GUID) and an internal identifier.
ODBC data source	1. The location of a set of data that can be accessed by using an ODBC driver. 2. A stored definition that contains all of the connection information that an ODBC application requires to connect to the data source.
ODBC driver	A dynamic-link library (DLL) that an ODBC-enabled application, such as Excel, can use to access an ODBC data source.
OGC	Open Geospatial Consortium
OLAP	See Other Term: online analytical processing
OLE DB	A COM-based application programming interface (API) for accessing data. OLE DB supports accessing data stored in any format for which an OLE DB provider is available.
OLE DB for OLAP	Formerly, the separate specification that addressed OLAP extensions to OLE DB. Beginning with OLE DB 2.0, OLAP extensions are incorporated into the OLE DB specification.

TERM	DEFINITION
one-to-many relationship	In relational databases, a relationship between two tables in which a single row in the first table can be related to one or more rows in the second table, but a row in the second table can be related only to one row in the first table.
one-to-one relationship	In relational database, a relationship between two tables in which a single row in the first table can be related only to one row in the second table, and a row in the second table can be related only to one row in the first table.
online analytical processing	A technology that uses multidimensional structures to provide rapid access to data for analysis.
operator	A sign or symbol that specifies the type of calculation to perform within an expression. There are mathematical, comparison, logical, and reference operators.
outer aggregate	An aggregate function that specifies a scope parameter that is another aggregate function.
padding	1. A string, typically added when the last plaintext block is short. 2. The space allotted in a cell to create or maintain a specific size.
page	In Report Builder, the page refers to the physical paper page. The paper size that you specify for the report controls how the report is rendered.
palette	A set of predefined or user-defined values for colors used in charts, maps, and gauges.
parameter	A built-in global collection in a report that enables users to set values that can vary report data, appearance, and connect related reports. Report parameters are created automatically from query parameters and dataset parameters, and manually by report authors.
parameterized report	A published report that accepts input values through parameters.
parameterized row filter	A row filter available with merge replication that allows you to restrict the data replicated to a Subscriber based on a system function or user-defined function.
parent	A member in the next higher level in a hierarchy that is directly related to the current member.
permission	A rule associated with an object to regulate which users can gain access to the object and in what manner.
pivot	1. To rotate rows to columns, and columns to rows, in a crosstabular data browser. 2. To choose dimensions from the set of available dimensions in a multidimensional data structure for display in the rows and columns of a crosstabular structure.

TERM	DEFINITION
placeholder	A character or symbol that is used in place of an actual value, text, or object. The actual value that the placeholder represents is unknown or unavailable at the current time, or is not displayed for security reasons. When a simple or complex expression is defined inside a text box, the resulting representation of this expression in design view is known as a placeholder.
planar data	Data that specifies points, lines, or areas on a flat geometric plane.
polygon layer	In a map report item, a layer that displays spatial data as areas, for example, geographical regions such as counties.
precision	The maximum total number of decimal digits that can be stored, both to the left and right of the decimal point.
primary key	A column or set of columns that uniquely identify all the rows in a table.
primary table	The "one" side of two related tables in a one-to-many relationship.
projection	A mapping of data from an N-dimensional coordinate system to an (N-1)-dimensional coordinate system. For example, mapping three dimensional geographic data onto a two dimensional display surface.
properties page	A dialog box that displays information about an object in the interface.
property	A named attribute of a control, field, or database object that you set to define one of the object's characteristics, such as size, color, or screen location; or an aspect of its behavior, such as whether it is hidden.
protocol	A standard set of formats and procedures that enable computers to exchange information.
provider	1. An OLE DB provider. 2. An in-process dynamic link library (DLL) that provides access to a database.
query parameters	Parameters that are specified in a dataset query, such as a Transact-SQL query.
RDL	See Other Term: Report Definition Language
record	A group of related fields (columns) of information treated as a unit. A record is more commonly called a row in a relational database.
rectangle	A report item that can be used as a container for multiple report items or as a graphical element on a report.

TERM	DEFINITION
recursive hierarchy	1. A hierarchy of data from a single report dataset that includes multiple hierarchical levels into a hierarchy structure, such as the report-to structure for manager-employee relationships in an organizational hierarchy. 2. In Master Data Services, a derived hierarchy that includes a recursive relationship. A recursive relationship exists when an entity has a domain-based attribute that is based on the entity itself.
relational database	A database or database management system that stores information in tables as rows and columns of data, and conducts searches by using the data in specified columns of one table to find additional data in another table.
relational database management system	A system that organizes data into related rows and columns.
relational OLAP	A storage mode that uses tables in a relational database to store multidimensional structures.
relationship	1. A link between tables that references the primary key in one table to a foreign key in another table. The relationship line is represented in a database diagram by a solid line if referential integrity between the tables is enforced, or a dashed line if referential integrity is not enforced for INSERT and UPDATE transactions. The endpoints of a relationship line show a primary key symbol to denote a primary key-to-foreign key relationship, or they show an infinity symbol to denote the foreign key side of a one-to-many relationship. 2. In Metadata Services, a relationship is an association between a pair of objects, where one object is an origin and the other object is a destination. The association repeats for each subsequent pair of objects, so that the destination of one relationship becomes the origin in the next relationship. In this way, all objects in an information model are associated through a chain of relationships that extend from one object to the next throughout the information model.
rendered report	A fully processed report that contains both data and layout information, in a format suitable for viewing.
rendering extension	A component in Reporting Services that is used to process the output format of a report.
rendering extension(s)	A plug-in that renders reports to a specific format.
rendering object model	Report object model used by rendering extensions.
report definition	The .rdl file that is the XML definition of a report and that conforms to the schema reportdefinition.xsd.
Report Definition Language	A set of instructions that describe layout and query information for a report.
report execution snapshot	A report snapshot that is cached.
report history	A collection of report snapshots that are created and saved over time.

TERM	DEFINITION
report history snapshot	A report snapshot that appears in report history.
report intermediate format	A static report history that contains data captured at a specific point in time.
report item	Any object, such as a text box, graphical element, or data region, that exists on a report layout.
report layout	The placement of data regions, report items, and text within a report and the applied formatting.
report layout template	A pre-designed table, matrix, or chart report template in Report Builder.
report link	A URL to a hyperlinked report.
report model	A metadata description of business data used for creating ad hoc reports in Report Builder.
report parameter	A parameter that is defined within a report definition. All query parameters have report parameters, but report parameters can be created independent of a query.
report parts	Report items that have been published separately to a report server and that can be reused in other reports. Report items such as tables, matrices, charts, and images can be published as report parts. Report parts have an .rsc file extension and they conform to the schema componentdefinition.xsd.
report processing extension	A component in Reporting Services that is used to extend the report processing logic.
report rendering	The action of combining the report layout with the data from the data source for the purpose of viewing the report.
report server	A location on the network where the ClickOnce version of Report Builder is launched from and a report is saved, managed, and published.
report server administrator	A user with elevated privileges who can access all settings and content of a report server.
report server database	A database that provides internal storage for a report server.
report server execution account	The account under which the Report Server Web service and Report Server Windows service run.
report server folder namespace	A hierarchy that contains predefined and user-defined folders. The namespace uniquely identifies reports and other items that are stored in a report server. It provides an addressing scheme for specifying reports in a URL.
Report Server service	A Windows service that contains all the processing and management capabilities of a report server.

TERM	DEFINITION
Report Server Web service	A Web service that hosts, processes, and delivers reports.
report snapshot	A static report that contains data captured at a specific point in time.
report-specific schedule	Schedule defined inline with a report.
ReportViewer controls	A Web server control and Windows Form control that provides embedded report processing in ASP.NET and Windows Forms applications.
resource	Any item in a report server database that is not a report, folder, or shared data source item.
result set	The set of rows returned from a SELECT statement.
role	1. A user or group to whom a set of specific permissions are granted. 2. In Analysis Services, a role uses Windows security accounts to limit scope of access and permissions when users access databases, cubes, dimensions, and data mining models. 3. In a database mirroring session, the principal server and mirror server perform complementary principal and mirror roles. Optionally, the role of witness is performed by a third server instance.
role assignment	A security policy that defines users and groups that can access specific items and perform specific operations.
role definition	A named collection of tasks that defines the operations a user can perform on a report server.
row	In an SQL table, a single occurrence of the object modeled by the table.
row aggregate function	A function that generates summary values, which appear as additional rows in the query results.
row filter	A filter that limits the rows to include in a dataset.
row identifier	1. A column or set of columns used to distinguish any single row from every other row in the table. 2. In a heap, a pointer to the row.
scalar	A single-value field, as opposed to an aggregate.
scalar aggregate	An aggregate function, such as MIN(), MAX(), or AVG(), that is specified in a SELECT statement column list that contains only aggregate functions.
schema	In the SQL-92 standard, a collection of database objects that are owned by a single user and form a single namespace. A namespace is a set of objects that cannot have duplicate names.

TERM	DEFINITION
scope	Used in multiple contexts. Scope can specify the data to use for evaluating an expression, the set of text boxes on a rendered page, or the set of report items that can be shown or hidden based on a toggle.
script	A collection of Transact-SQL statements used to perform an operation.
securable	Entities that can be secured with permissions.
Secure Sockets Layer (SSL)	A proposed open standard for establishing a secure communications channel to prevent the interception of critical information, such as credit card numbers. Primarily, it enables secure electronic financial transactions on the World Wide Web, although it is designed to work on other Internet services as well.
security extension	A component in Reporting Services that authenticates a user or group to a report server.
security principal	In Windows-based computers, an account (such as a user, security group, device, or computer) that can be granted or denied access to resources.
Semantic Model Definition Language (SMDL)	A set of instructions that describe layout and query information for reports created in Report Builder.
series	In a chart, a series is made up of more than one data point.
server aggregate	An aggregate value that is calculated on the data source. Depending on the data source, server aggregates can be treated as detail data or as aggregates based on the dataset option InterpretSubtotalsAsDetails.
server name	A name that uniquely identifies a server computer on a network.
service	A Service Broker object that defines a name for a specific task or set of tasks, and the contracts that other services can use to accomplish that task.
service principal name	The name by which a client uniquely identifies an instance of a service.
shared data source	A data source definition that is saved and managed as a file on a report server and can be accessed by multiple reports.
shared data source definition	The .rsd file that is the XML definition of a data source and that conforms to the schema shareddatasource.xsd.
shared dataset definition	The .rcd file that is the XML definition of a shared dataset and that conforms to the schema shareddatasetdefinition.xsd.
simple expression	An expression that contains a reference to a single field and appears in the report layout in brackets, e.g. [ProductID].

TERM	DEFINITION
SKU	A way of identifying different editions of one product; also, acronym for stock keeping unit (standard business terminology).
slice	A subset of the data in a cube, specified by limiting one or more dimensions by members of the dimension.
smart tag	A smart tag exposes key configurations directly on the design surface to enhance overall design-time productivity in Visual Studio 2005.
SMDL	See Other Term: Semantic Model Definition Language (SMDL)
snapshot	See Other Term: report snapshot
sort order	The set of rules in a collation that define how characters are evaluated in comparison operations, and the sequence in which they are sorted.
spatial data	Data that specifies locations of objects on a flat or curved surface. Spatial data can be points, lines, or polygons.
spatial data types	Specifies data that represents geometry (planar) or geography (geodesic) information.
SQL	See Other Term: Structured Query Language (SQL)
SQL query	An SQL statement, such as SELECT, INSERT, UPDATE, DELETE, or CREATE TABLE.
SQL Server Authentication	The mechanism to validate an attept to connect to an instance of SQL Server by specifying a SQL Server login ID and password.
SQL Server login	An account stored in SQL Server that allows users to connect to SQL Server.
SQL statement	An SQL or Transact-SQL command, such as SELECT or DELETE, that performs some action on data.
static connection string	A set of values that the report always uses to connect to the same data source each time the report runs.
string	A data type that contains letters, numbers, and most characters. Numbers in a String field, that is a field with a data type of String, are not available for numeric calculations. You must use a numeric data type, such as Integer or Float, to perform numeric calculations. For a parameter, String is the same as the Text data type.

TERM	DEFINITION
string functions	Functions that perform operations on character or binary strings.
Structured Query Language (SQL)	A language used to insert, retrieve, modify, and delete data in a relational database, designed specifically for database queries.
subreport	A subreport is a control embedded inside the body of a parent report. Conceptually, a subreport is similar to a frame in a Web page that provides a container for other Web page content. The subreport is rendered inside the parent report that contains it. Both reports are processed and displayed simultaneously.
symmetric key	A single key that is used with symmetric encryption algorithms for both encryption and decryption.
system role assignment	Role assignment that applies to the site as a whole.
system role definition	Role definition that conveys site-wide authority.
system stored procedures	A set of SQL Server-supplied stored procedures that can be used for actions such as retrieving information from the system catalog or performing administration tasks.
table	1. A two-dimensional object, which consists of rows and columns, that stores data about an entity modeled in a relational database. 2. A data region on a report layout that displays data in a columnar format.
table data region	A report item on a report layout that displays data in a columnar format.
tablix	A Reporting Services RDL data region that contains rows and columns resembling a table or matrix, possibly sharing characteristics of both.
task	A collection of permissions that constitute a task (manage reports, manage folders, etc)
Transact-SQL	The language containing the commands used to administer instances of SQL Server, create and manage all objects in an instance of SQL Server, and to insert, retrieve, modify and delete all data in SQL Server tables. Transact-SQL is an extension of the language defined in the SQL standards published by the International Standards Organization (ISO) and the American National Standards Institute (ANSI).
user instance	An instance of SQL Server Express that is generated by the parent instance on behalf of a user.
value (y) axis	The axis for displaying numeric data values in a chart. Usually the vertical axis. Exception: in bar charts, the axes are reversed, and the y axis displays grouping data.

TERM	DEFINITION
value expression	An expression in Multidimensional Expressions (MDX) that returns a value. Value expressions can operate on sets, tuples, members, levels, numbers, or strings.
variable	1. In Integration Services, stores values that can be used in scripts, expressions, and property expressions to set column values and the properties of package objects. 2. Defined entities that are assigned values. A local variable is defined with a DECLARE@localvariable statement and assigned an initial value within the statement batch where it is declared with either a SELECT or SET@localvariable statement.
variable interval	An option on a Reporting Services chart that can be specified to automatically calculate the optimal number of labels that can be placed on an axis, based on the chart width or height.
visualization	In maps, charts, and gauges, the way that a user chooses to visualize analytical data.
Web service	In Reporting Services, a service that uses Simple Object Access Protocol (SOAP) over HTTP and acts as a communications interface between client programs and the report server.
Windows Management Instrumentation	An interface that provides information about objects in a managed environment.
WKB	Well Known Binary data representation as specified by the Open Geospatial Consortium (OGC).
WKT	Well Known Text data representation as specified by the Open Geospatial Consortium (OGC).
WMI	See Other Term: Windows Management Instrumentation
x-axis	See Other Term: category (x) axis
y-axis	See Other Term: value (y) axis

See Also

Report Authoring Concepts (Report Builder and SSRS)