

Contents

Overview

Report Server (SharePoint)

SharePoint Service and service applications

- Manage a Reporting Services SharePoint service application

- Back up and restore Reporting Services SharePoint service applications

PowerShell cmdlets for Reporting Services SharePoint mode

Configuration and administration (SharePoint mode)

- Site collection features - Reporting Services

- Site collection features - Report Server and Power View

- Site settings and features - Reporting Services

- Site settings and features - Server File Sync in SharePoint

- Add Reporting Services content types to a SharePoint library

- Local mode vs. connected mode reports in the Report Viewer

- Upload documents to a SharePoint library

- Set processing options

Report Viewer web part on a SharePoint site

- Deploy the Report Viewer web part

- Add Report Viewer web part to a web page

- Connect Filter or Documents web part

- Customize the Report Viewer web part

- SharePoint site settings for the Report Viewer web part

High availability

Enable and disable RDL sandboxing

Configure Reporting Services to use a subject alternative name

Reporting Services report server

8/13/2018 • 10 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016 and later)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

Learn about the central piece of a SQL Server Reporting Services installation. It consists of a processing engine along with extensions to add functionality.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

A Reporting Services report server runs in one of two deployment modes; Native mode or SharePoint mode. See the [Feature Comparison of SharePoint and Native Mode](#) section for a comparison of features.

Installation: For information on Reporting Services installation, see [Install Reporting Services](#).

Overview of report server modes

Processing engines (processors) are the core of the report server. The processors support the integrity of the reporting system and cannot be modified or extended. Extensions are also processors, but they perform very specific functions. Reporting Services includes one or more default extensions for every type of supported extension. You can add custom extensions to a report server. Doing so allows you to extend a report server to support features that are not supported out of the box; examples of custom functionality might include support for single sign-on technologies, report output in application formats that are not already handled by the default rendering extensions, and report delivery to a printer or application.

A single report server instance is defined by the complete collection of processors and extensions that provide end-to-end processing, from the handling of the initial request to the presentation of a finished report. Through its subcomponents, the report server processes report requests and makes reports available for on-demand access or scheduled distribution.

Functionally, a report server enables report authoring experiences, report rendering, and report delivery experiences for a variety of data sources as well as extensible authentication and authorization schemes. Additionally a report server contains report server databases that store published reports, shared data sources, shared datasets, report parts, shared schedules and subscriptions, report definition source files, model definitions, compiled reports, snapshots, parameters, and other resources. A report server also enables administration experiences for configuring the report server to process report requests, maintain snapshot histories, and manage permissions for reports, data sources, datasets, and subscriptions.

A Reporting Services report server supports two modes of deployment for report server instances:

- **Native mode:** including native mode with SharePoint web parts, where a report server runs as an application server that provides all processing and management capability exclusively through Reporting Services components. You configure a native mode report server with Reporting Services configuration manager and SQL Server Management Studio.
- **SharePoint mode:** where a report server is installed as part of a SharePoint server farm. Deploy and configure SharePoint mode by using PowerShell commands or SharePoint content management pages.

In SQL Server Reporting Services you cannot switch a report server from one mode to the other. If you want to change the type of report server that your environment uses, you must install the desired mode of report server and then copy or move the report items or report server database from the older versioned report server to the new report server. This process is typically referred to as a 'migration'. The steps needed to migrate depend on the mode you are migrating to and the version you are migrating from. For more information, see [Upgrade and Migrate Reporting Services](#)

Feature comparison of SharePoint and native mode

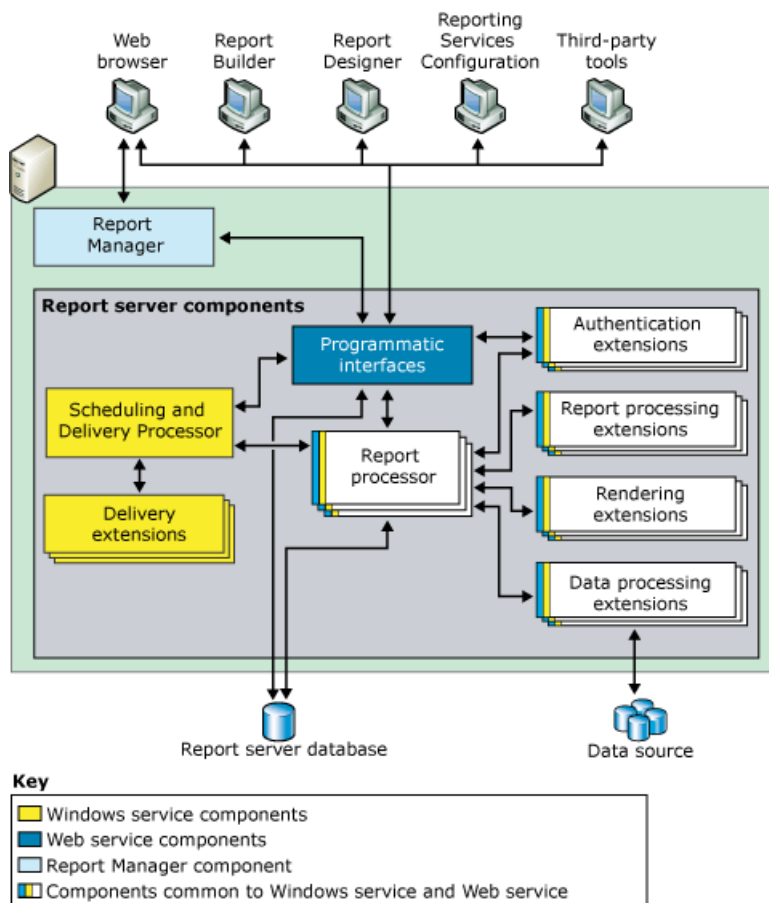
FEATURE OR COMPONENT	NATIVE MODE	SHAREPOINT MODE
URL addressing	Yes	<p>URL addressing is different in SharePoint integrated mode. SharePoint URLs are used to reference reports, report models, shared data sources, and resources. The report server folder hierarchy is not used. If you have custom applications that rely on URL access as supported on a native mode report server, that functionality will no longer work when the report server is configured for SharePoint integration.</p> <p>For more information on URL access, see URL Access Parameter Reference</p>
Custom security extensions	Yes	Reporting Services custom security extensions cannot be deployed or used on the report server. The report server includes a special-purpose security extension that is used whenever you configure a report server to run in SharePoint integrated mode. This security extension is an internal component, and it is required for integrated operations.
Configuration Manager	Yes	** Important ** Configuration Manager cannot be used to manage a SharePoint mode report server. Instead, use SharePoint central administration.
Report Manager	Yes	Report Manager cannot be used to manage SharePoint mode. Use the SharePoint application pages. For more information, see Reporting Services SharePoint Service and Service Applications .
Linked Reports	Yes	No.
My Reports	Yes	No
My Subscriptions and batching methods.	Yes	No
Data Alerts	No	Yes

FEATURE OR COMPONENT	NATIVE MODE	SHAREPOINT MODE
Power View	No	Yes Requires Silverlight in the client browser. For more information on browser requirements, see Browser Support for Reporting Services and Power View
.RDL reports	Yes	Yes .RDL reports can run on Reporting Services report servers in native mode or in SharePoint mode.
.RDLX reports	No	Yes Power View .RDLX reports can only run on Reporting Services report servers in SharePoint mode.
SharePoint user token credentials for the SharePoint list extension	No	Yes
AAM zones for internet facing deployments	No	Yes
SharePoint backup and recovery	No	Yes
ULS log support	No	Yes

Native mode

In native mode, a report server is a stand-alone application server that provides all viewing, management, processing, and delivery of reports and report models. This is the default mode for report server instances. You can install a native mode report server that is configured during setup or you can configure it for native mode operations once setup is complete.

The following diagram shows the three-tier architecture of a Reporting Services Native mode deployment. It shows the report server database and data sources in the data tier, the report server components in the middle tier, and the client applications and built-in or custom tools in the presentation tier. It shows the flow of requests and data among the server components and which components send and retrieve content from a data store.



The report server is implemented as a Microsoft Windows service, called the "Report Server service", that hosts a Web service, background processing, and other operations. In the Services console application, the service is listed as SQL Server Reporting Services (MSSQLSERVER).

Third-party developers can create additional extensions to replace or extend the processing capability of the report server. To learn more about the programmatic interfaces available to application developers, see the [Technical Reference](#).

Native mode with SharePoint web parts

Reporting Services provides two web parts that you can install and register on an instance of Windows SharePoint Services 2.0 or later, or SharePoint Portal Server 2003 or later. From a SharePoint site, you can use the web parts to find and view reports that are stored and processed on a report server that runs in native mode. These web parts were introduced in earlier releases of Reporting Services.

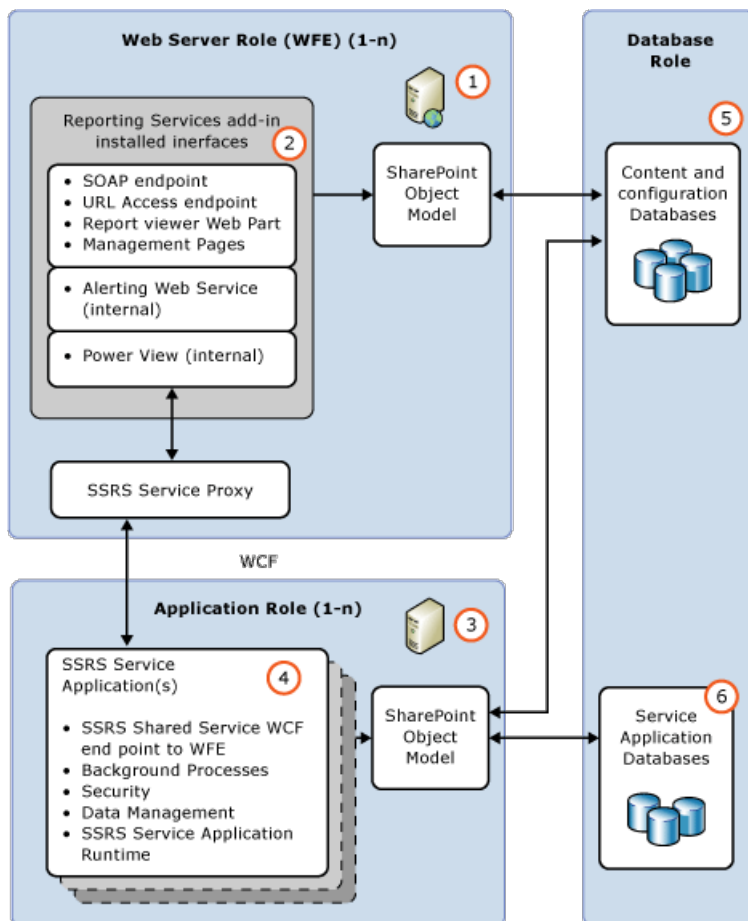
SharePoint mode

In SharePoint mode, a report server must run within a SharePoint server farm. The report server processing, rendering, and management features are represented by a SharePoint application server running the Reporting Services SharePoint shared service and one or more Reporting Services service applications. A SharePoint site provides the front-end access to report server content and operations.

SharePoint mode requires:

- SharePoint Foundation 2010 or SharePoint Server 2010.
- An appropriate version of the Reporting Services Add-in for SharePoint 2010 Products.
- A SharePoint application server with the Reporting Services shared service installed and at least one Reporting Services service application.

The following illustration shows a SharePoint mode Reporting Services environment:



	DESCRIPTION
(1)	Web servers or web front-ends (WFE). The Reporting Services add-in must be installed on each web server from which you want to utilize the web application features such as viewing reports or Reporting Services management pages for tasks such as managing data sources or subscriptions.
(2)	The add-in installs URL and SOAP endpoints for clients to communicate with the Application servers, through the Reporting Services service proxy.
(3)	Application servers running Reporting Services shared service. Scale-out of report processing is managed as part of the SharePoint farm and by adding the Reporting Services service to additional application servers.
(4)	You can create more than one Reporting Services service application, with different configurations including permissions, e-mail, proxy, and subscriptions.
(5)	Reports, data sources, and other items are stored in the SharePoint content databases.
(6)	Reporting Services service applications create three databases for report server, temp, and data alerting features. Configuration settings that apply to all SSRS service applications are stored in the RSReportserver.config file.

Report process and schedule and delivery process

The report server includes two processing engines that perform preliminary and intermediate report processing, and scheduled and delivery operations. The Report Processor retrieves the report definition or model, combines layout information with data from the data processing extension, and renders it in the requested format. The Scheduling and Delivery Process processes reports triggered from a schedule, and delivers reports to target destinations.

Report server database

The report server is a stateless server that stores all properties, objects, and metadata in a SQL Server database. Stored data includes published reports, compiled reports, report models, and the folder hierarchy that provides the addressing for all items managed by the report server. A report server database can provide internal storage for a single Reporting Services installation or for multiple report servers that are part of a scale-out deployment. If you configure a report server to run within a larger deployment of a SharePoint product or technology, the report server uses the SharePoint databases in addition to the report server database. For more information about data stores used in Reporting Services installation, see [Report Server Database \(SSRS Native Mode\)](#).

Authentication, rendering, data, and delivery extensions

The report server supports the following types of extensions: authentication extensions, data processing extensions, report processing extensions, rendering extensions, and delivery extensions. A report server requires at least one authentication extension, data processing extension, and rendering extension. Delivery and custom report processing extensions are optional, but necessary if you want to support report distribution or custom controls.

Reporting Services provides default extensions so that you can use all of the server features without having to develop custom components. The following table describes the default extensions that contribute to a complete report server instance that provides ready-to-use functionality:

TYPE	DEFAULT
Authentication	A default report server instance supports Windows Authentication, including impersonation and delegation features if they are enabled in your domain.
Data processing	A default report server instance includes data processing extensions for SQL Server, Analysis Services, Oracle, Hyperion Essbase, SAPBW, OLE DB, Parallel Data Warehouse, and ODBC data sources.
Rendering	A default report server instance includes rendering extensions for HTML, Excel, CSV, XML, Image, Word, SharePoint list, and PDF.
Delivery	A default report server instance includes an e-mail delivery extension and a file share delivery extension. If the report server is configured for SharePoint integration, you can use a delivery extension that saves reports to a SharePoint library.

NOTE

Reporting Services includes a complete set of tools and applications that you can use to administer the server, create content, and make that content available to users in your organization.

Related tasks

The following topics provide additional information on installing, using, and maintaining a report server:

TASK	LINK
Review Hardware and software requirements.	Hardware and Software Requirements for Reporting Services in SharePoint Mode.
Install Reporting Services in SharePoint mode.	Install Reporting Services SharePoint Mode for SharePoint 2010
If you are a Web developer or have expertise in creating cascading style sheets, you can modify the default styles at your own risk to change the colors, fonts, and layout of the toolbar or Report Manager. Neither the default style sheets nor instructions for modifying the style sheets are documented in this release.	Customize Style Sheets for HTML Viewer and Report Manager
Web developers who are familiar with HTML styles and Cascade Style Sheets (CSS) can use the information in this topic to determine which files can be modified to customize the appearance of Report Manager.	Configure the Web Portal to Pass Custom Authentication Cookies
Explains how to tune the memory settings for the Report Server Web service and Windows service.	Configure Available Memory for Report Server Applications
Explains recommended steps to configure a report server for remote administration.	Configure a Report Server for Remote Administration
Provides instructions for configuring the availability of My Reports on a Native report server instance.	Enable and Disable My Reports
Provides instructions for setting up the RSClientPrint control that provides print functionality from within supported browsers. For more information on browser requirements, see Browser Support for Reporting Services and Power View.	Enable and Disable Client-Side Printing for Reporting Services

Next steps

[Reporting Services Extensions](#)

[Reporting Services Tools](#)

[Subscriptions and Delivery \(Reporting Services\)](#)

[Report Server Database \(SSRS Native Mode\)](#)

[Implementing a Security Extension](#)


[Implementing a Data Processing Extension](#)

[Data Sources Supported by Reporting Services \(SSRS\)](#)

More questions? [Try asking the Reporting Services forum](#)

Reporting Services Report Server (SharePoint Mode)

5/3/2018 • 7 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

A Reporting Services report server configured for **SharePoint Mode** can run within a deployment of a SharePoint product. A report server in SharePoint mode can use the collaboration and management features of SharePoint for reports and other Reporting Services content types. SharePoint mode requires installing the appropriate version of the Reporting Services add-in for SharePoint products on your SharePoint Web Front Ends.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

For more information on installing and configuring, see the following:

- [Install Reporting Services SharePoint mode for SharePoint 2010](#).
- [Add an additional Report Server to a farm](#).

Feature summary

Configuring a report server to run in SharePoint integrated mode provides the following additional functionality that is only available when you deploy a report server in this mode:

- Use SharePoint document management and collaboration features, including alerts. A SharePoint site provides a unified portal for accessing and managing all report items in one place.
- Use SharePoint permissions and authentication providers to control access to reports, models, and other items.
- Use SharePoint deployment topologies to distribute reports over an Internet connection outside the firewall. A report server provides report and data processing services in the context of a larger SharePoint deployment that is configured for Internet access.
- Manage reports, models, data sources, schedules, and report history in custom application pages on a SharePoint site. You can set properties, define schedules and subscriptions, and create and manage report history on a SharePoint site the same way you create and manage them from other tools in SQL Server.
- Publish or upload reports, report models, resources, and shared data source files to a SharePoint library, including Report Center in Office SharePoint Server.

Use Report Designer, Model Designer, and Report Builder to create reports and data sources to be published directly to a SharePoint library. You can also use the Upload action on a SharePoint site to add any report definitions and report models to a SharePoint library.

Because the report server processes report definitions in the same way regardless of the server mode you use, the report data and layout is unaffected by server mode. Any report that you can run in a native mode report server can run on a report server that is configured for SharePoint integrated mode.

- Subscribe to and deliver reports to a SharePoint library using a new SharePoint delivery extension. You can

also deliver reports through e-mail or to a shared folder. The report server delivery extensions are used to deliver reports. You can create data-driven subscriptions for large-scale report distribution using subscriber data queried at run time.

- A Report Viewer web part you can add to SharePoint pages to view a report inside your SharePoint web application. The web part includes page navigation, search, print, and export features.
- Program against a new SOAP endpoint to create custom applications that integrate with a SharePoint site. You can also use the updated Windows Management Instrumentation (WMI) provider to programmatically configure a report server instance that runs in SharePoint integrated mode.
- Microsoft Access services reporting, in connected mode.
- AAM zones, internet facing deployments, and SharePoint user tokens for SharePoint lists.

Connected mode and local mode

The SQL Server 2008 R2 release introduced a new *local mode* for viewing reports from a SharePoint 2010 server that has the Microsoft SQL Server 2008 R2 or later Reporting Services Add-In for SharePoint 2010 products installed.

- *Local Mode*: Local mode allows reports to be rendered locally from the SharePoint document library, without integration with a Reporting Services report server. The Reporting Services add-in for SharePoint products is required but a Reporting Services report server is not. The add-in can be installed several different ways, including the SharePoint 2010 products preparation tool. For more information on local mode, see [Local mode vs. connected mode reports in the Report Viewer](#) and [Where to find the Reporting Services add-in for SharePoint products](#).
- *Connected Mode*: Connected mode is supported by integrating a Reporting Services report server into the SharePoint farm using SharePoint Central Administration. The integration with a report server enables full end-to-end reporting, providing the collaboration features of SharePoint 2010 and the server based features of a report server including: Subscriptions, Snapshots, and server based processing.

Unsupported sharePoint features

Not all SharePoint features are available for integrated operations. The following is a list of the SharePoint features Reporting Services does not directly integrate with:

- Secure Store Service.
- You cannot use the SharePoint Outlook Calendar integration or the SharePoint scheduling for reporting services files in a document library.
- SharePoint Business Data catalog.
- SharePoint personalization is also not supported on the Reporting Services pages. Report Server integration is not supported if the SharePoint Web application is enabled for Anonymous access.
- SQL Server Reporting Services does **not** support SharePoint document library version control. If you save report items in a document library that is configured with "Document Version History" enabled, Reporting Services features will not function correctly and generate errors in the ULS log. The following is an example of an error in the ULS log:
 - "...a data source associated with the report has been disabled".

Document library version history is configured on the "Versioning Settings" page of "Library Settings".

Supported combinations of the SharePoint add-in and report server

Not all features are supported in all combinations of report server, Reporting Services add-in for SharePoint, and SharePoint Products. For more information, see [Supported combinations of SharePoint and Reporting Services Server and add-in](#)

NOTE

The correct version of the Reporting Services add-in must be used with the corresponding version of SharePoint Products.

Components that provide integration

To combine the servers in a single deployment, you integrate an installation of SQL Server Reporting Services with an instance of SharePoint products

Integration is provided through SQL Server and the Reporting Services Add-in for SharePoint Products. The Reporting Services Add-in is a freely distributable component that you can download and then install on a server that is running the appropriate version of SharePoint.

TIP

Not all features are supported in all combinations of report server, Reporting Services add-in for SharePoint, and SharePoint Products. For more information see, [Supported combinations of SharePoint and Reporting Services Server and add-in](#).

- On SharePoint, the Reporting Services Add-in provides the ReportServer proxy endpoint, a Report Viewer web part, and application pages so that you can view, store, and manage report server content on a SharePoint site or farm.
- On Reporting Services provides updated program files, a SOAP endpoint, and custom security and delivery extensions. The report server must be configured to run in SharePoint integrated mode, dedicated exclusively to supporting report access and delivery through your SharePoint site.

After you install the Reporting Services Add-in on SharePoint and configure the two servers for integration, you can upload or publish report server content types to a SharePoint library, and then view and manage those documents from a SharePoint site. Uploading or publishing report server content is an important first step; the web part and pages become available when you select report definitions (.rdl), report models (.smdl) and shared data sources (.rsds) on a SharePoint site.

Language considerations

SharePoint Foundation 2010 and SharePoint Server 2010 products are available in many more languages than SQL Server

When you configure a report server to run within a deployment of a SharePoint product, you might see a combination of languages. The user interface, documentation, and messages will appear in the following languages:

- All application pages, tools, errors, warnings, and messages that originate from Reporting Services will appear in the language used by the Reporting Services instance in one of the SQL Server languages.
- Application pages that you open on a SharePoint site, the Report Viewer web part, and Report Builder will appear in one of the supported languages for the Reporting Services Add-in. To view the list of supported languages, go to [SQL Server downloads](#) and find the download page for the SQL Server 2016 Reporting Services Add-in.

- SharePoint sites, SharePoint Central Administration, online help, and messages are available in the languages supported by Office Server products.

If the language of your SharePoint product or technology differs from the report server language, Reporting Services will try to use a language from the same language family that provides the closest match. If a close substitute is not available, the report server uses English.

Related tasks

The following table summarizes tasks related to a Reporting Services SharePoint mode report server:

TASK	LINK
Detailed steps for installing and configuring Reporting Services in SharePoint mode.	Install Reporting Services SharePoint mode for SharePoint 2010 and Add an additional Report Server to a farm .
Scale-out your Reporting Services SharePoint deployment by adding additional report servers.	Add an additional Report Server to a Farm and Deployment topologies for SQL Server BI features in SharePoint .
Add additional SharePoint web front-ends that have the Reporting Services components installed for viewing and report items.	Add an additional Reporting Services web front-end to a farm
Configure e-mail for your report server within SharePoint.	Configure e-mail for a Reporting Services service application
Recent information for this release, found on the TechNet Wiki.	SQL Server 2012 Reporting Services tips, tricks, and troubleshooting .

Next steps

[Install or uninstall the Reporting Services sdd-in for SharePoint](#)




[Report Viewer web part on a SharePoint site](#)

[Quiz: Configuring SSRS 2012 for SharePoint integration](#)

More questions? [Try asking the Reporting Services forum](#)

Reporting Services SharePoint service and service applications

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

Reporting Services SharePoint mode is architected on the SharePoint service architecture and utilizes a SharePoint service and one to many service applications. Creating a service application makes the service available and generates the service application database. You can create multiple Reporting Services service applications but one service application is sufficient for most deployment scenarios.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

Creating a Reporting Services service application

You can use SharePoint Central Administration or PowerShell scripts to create the Reporting Services services applications. For more information on using SharePoint Central Administration, see the "Create a Reporting Services Service Application" section in [Install Reporting Services SharePoint Mode for SharePoint 2010](#). See the PowerShell section later in this topic for a sample PowerShell script for creating service applications.

Modify the associations of the service application with a proxy group

The New page for creating a services application contains the section **Web Application Association**. The section allows you to associate your service application as you create it. Use the following steps to change the association and assign a customer configuration to the service application. The same general process can also be used to add the proxy to the default group rather than changing the association of the service application to a custom group.

1. In SharePoint Central Administration, in the Application Management, click **Configure Service Application Associations**.
2. On the Service application Associations page, change the view to **Service Applications**.
3. Find and click the name of your new Reporting Services Service application. You could also click the application proxy group name **default** to add the proxy to default group rather than completing the following steps.
4. Select **Custom** in the selection box **Edit the following group of connections**.
5. Check the box for your proxy and click **Ok**.

Edit service application properties

You can reopen the property page of the service application to modify the properties.

1. In SharePoint Central Administration, in the Application Management group, click **Manage service applications**.

2. Select the service application by clicking the type column to select the entire row. If you click the name of the application it, the Management options page for the service opens instead of opening the properties of the service application.
3. In the Service Applications ribbon, click **Properties**.

Create a Reporting Services service application using PowerShell

You can use PowerShell to create the Service application and proxy. The sample below assumes that you know what application pool you want to configure the service application to use.

1. Add the application pool object of your application pool name to a variable that is passed into the New action.

```
$appPoolName = get-spserviceapplicationpool "<application pool name>"
```

2. Create the service application with a name and application pool name you provide.

```
New-SPRSServiceApplication -Name 'MyServiceApplication' -ApplicationPool $appPoolName -DatabaseName 'MyServiceApplicationDatabase' -DatabaseServer '<Server Name>'
```

3. Get the new service application object, and pipe the object into the Pipe the new proxy cmdlet.

```
Get-SPRSServiceApplication -name MyServiceApplication | New-SPRSServiceApplicationProxy "MyServiceApplicationProxy"
```




Related tasks

TASK	LINK
Manage the settings of your Service Application.	Manage a Reporting Services SharePoint Service Application
Backup and restore the service application and related components such as encryption keys and proxy.	Backup and Restore Reporting Services SharePoint Service Applications

More questions? [Try asking the Reporting Services forum](#)

Manage a Reporting Services SharePoint service application

5/3/2018 • 8 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

Reporting Services service applications are managed from SharePoint Central Administration. The Management and Properties pages allow you to update the configuration of the service application as well as common administration tasks.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

Open service application properties page

To open the properties page for a Reporting Services service application, complete the following:

1. In Central Administration, in the Application Management group, click **Manage service applications**.
2. Click near the name of your service application or on the **type** column, which will select the entire row, and then click **Properties** in the SharePoint ribbon.

For more information on service application properties, see [Step 3: Create a Reporting Services Service Application](#).

Open service application management pages

To open the management pages for a Reporting Services service application, complete the following:

1. In Central Administration, in the Application Management group, click **Manage service applications**.
2. Click the name of your service application and the **Manage Reporting Services Application** page will open.
3. Alternatively, you could click near the name or on the **type** column for your service application, which will select the entire row, and then click **Manage** in the SharePoint ribbon.

System settings page

The systems settings page allows you to configure the behavior and user experience of your service application including various timeouts.

Report settings

SETTING	COMMENTS
External Images Timeout	Default is 600 seconds.

SETTING	COMMENTS
Snapshot Compression	Default is SQL
System Report Timeout	<p>Default is 1800 seconds.</p> <p>Specify whether report processing times out on the report server after a certain number of seconds. This value applies to report processing on a report server. It does not affect data processing on the database server that provides the data for your report. The report processing timer clock begins when the report is selected and ends when the report opens. The value that you specify must be sufficient to complete both data processing and report processing.</p>
System Snapshot Limit	<p>Default is -1, which is no limit.</p> <p>Set a site-wide default value for the number of copies of report history to retain. The default value provides an initial setting that establishes the number of snapshots that can be stored for each report. You can specify different limits in property pages for specific reports.</p>
Stored Parameters Lifetime	Default is 180
Stored Parameters Threshold	Default is 1500 days.

Session settings

SETTING	COMMENTS
Session Timeout	Default is 600 seconds.
Use Session Cookies	Default is TRUE.
EDLX Report Timeout	Default is 1800 seconds.

System settings for logging

SETTING	COMMENTS
Enable Execution Logging	<p>Default is TRUE.</p> <p>specify whether the report server generates trace logs and the number of days the log is kept. . The logs are stored on the report server computer in the \Microsoft SQL Server\MSSQL.n\ReportServer\Log folder. A new log file is started each time the service is restarted. For more information about log files, see Report Server Service Trace Log</p>
Execution Log Days Kept	Default is 60 days.

Reporting Services all supports SharePoint ULS logging. For more information, see [Turn on Reporting Services events for the SharePoint trace log \(ULS\)](#)

Security settings

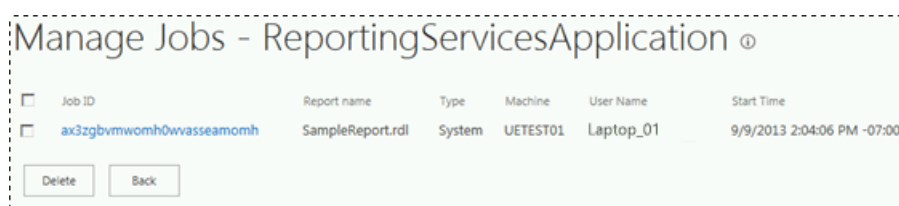
SETTING	COMMENTS
Enable Integrated Security	Default is TRUE. Specifies whether a connection to a report data source can be made using the Windows security token of the user who requested the report.
Enable Load Report Definition	Default is TRUE..
Enable Remote Errors	Default is FALSE
Enable Test Connection Detailed Errors	Default is TRUE.

Client settings

SETTING	COMMENTS
Enable Report Builder Download	Default is TRUE. Specifies whether clients are able to see the button for downloading the report builder application.
Report Builder Launch URL	Specify a custom URL when the report server does not use the default Report Builder URL. This setting is optional. If you do not specify a value, the default URL will be used, which launches Report Builder. To launch Report Builder 3.0 as a Click-Once application, enter the following value: http://<computername>/ReportServer/ReportBuilder/ReportBuilder_3_0_0_0.application.
Enable Client Printing	The Default is TRUE. Specifies whether users can download the client side control, which provides printing options.
Edit Session Timeout	Default is 7200 seconds.
Edit Session Cache Limit	Default is 5.

Manage jobs

You can view and delete the running jobs, for example jobs that are created by report subscriptions and data-driven subscriptions. The page is not used to manage subscriptions, but rather jobs that are triggered by a subscription. For example a subscription that is scheduled to run once an hour will generate a job once an hour that appears on the **Manage Jobs** page.



Key management

The following table summarizes the Kay Management pages

IMPORTANT

Periodically changing the Reporting Services encryption key is a security best practice. A recommended time to change the key is immediately following a major version upgrade of Reporting Services. Changing the key after an upgrade minimizes additional service interruption caused by changing the Reporting Services encryption key outside of the upgrade cycle.

PAGE	DESCRIPTION
Backup Encryptions Key	<p>1) Type a password in to the Password: and Confirm Password: boxes and click Export. You will see a warning if the password you typed is does not meet the complexity requirements of the domain policy.</p> <p>2) You will be prompted for a file location of where to save the key file. You should consider storing the key file on a separate computer from the one that is running Reporting Services. The default file name is the same name as the service application.</p>
Restore Encryption Key	<p>1) Type or browse to the key file in the File Location box</p> <p>2) In the Password box, type the password that was used to back up the encryption file.</p> <p>3) Click OK</p>
Change Encryption Key	<p>This operation will create a new key and re-encrypt your encrypted content. If you have a lot of content, this operation make take several hour.</p> <p>When the change encryption key operation is complete, it is recommended you make a backup of your new key.</p>
Deleted Encrypted Content	<p>Deleted content cannot be recovered.</p> <p>** Important * The action of deleting and recreating the symmetric key cannot be reversed or undone. Deleting or recreating the key can have important ramifications on your current installation. If you delete the key, any existing data encrypted by the symmetric key will also deleted. Deleted data includes connection strings to external report data sources, stored connection strings, and some subscription information.</p>

Execution account

Use this page to configure an account to use for unattended processing. This account is used under special circumstances when other sources of credentials are not available:

- When the report server connects to a data source that does not require credentials. Examples of data sources that might not require credentials include XML documents and some client-side database applications.
- When the report server connects to another server to retrieve external image files or other resources that are referenced in a report.

Setting this account is optional, but not setting it limits your use of external images and connections to some data sources. When retrieving external image files, the report server checks to see if an anonymous

connection can be made. If the connection is password protected, the report server uses the unattended report processing account to connect to the remote server. When retrieving data for a report, the report server either impersonates the current user, prompts the user to provide credentials, uses stored credentials, or uses the unattended processing account if the data source connection specifies **None** as the credential type. The report server does not allow its service account credentials to be delegated or impersonated when connecting to other computers, so it must use the unattended processing account if no other credentials are available.

The account you specify should be different from the one used to run the service account. If you are running the report server in a scale-out deployment, you must configure this account the same way on each report server.

You can use any Windows user account. For best results, choose an account that has read permissions and network logon permissions to support connections to other computers. It must have read permissions on any external image or data file that you want to use in a report. Do not specify a local account unless all report data sources and external images are stored on the report server computer. Use the account only for unattended report processing.

PowerShell command

The following is an example PowerShell command to return the list of Reporting Services service applications with the UEAccount property:

```
Get-SPRSServiceApplication | select typename, name, service, ueaccountname
```

For more information, see [PowerShell cmdlets for Reporting Services SharePoint Mode](#).

Options

Specify an execution account

Select to specify an account.

Account

Enter a Windows domain user account. Use this format: *<domain>\<user account>*.

Password

Type the password.

Confirm password

Retype the password.

E-mail settings

Use this page to specify Simple Mail Transport Protocol (SMTP) settings that enable report server e-mail delivery from the report server. You can use the Report Server E-Mail delivery extension to distribute reports or report processing notifications through e-mail subscriptions. The Report Server E-Mail delivery extension requires an SMTP server and an e-mail address to use in the From: field.

Options

Use SMTP Server

Specifies that report server e-mail is routed through an SMTP server.

Outbound SMTP Server

Specify the SMTP server or gateway to use. You can use a local server or an SMTP server on your network.

From Address

Specifies the e-mail address to use in the From: field of a generated e-mail. You must specify a user account that has permission to send mail from the SMTP server.

Provision subscriptions and alerts

Use this page to verify if SQL Server Agent is running and to provision access for reporting services to use SQL Server Agent. SQL Server Agent is required for Reporting Services subscriptions, schedules, and data alerts.

[Provision Subscriptions and Alerts for SSRS Service Applications](#)

Proxy association




When you created the Reporting Services service application you selected the web application to associate and provision permissions for access by the Reporting Services service application. If you chose to not associate or you want to change the association, you can use the following steps.

1. In SharePoint Central Administration, in the Application Management, click **Configure Service Application Associations**.
2. On the Service application Associations page, change the view to **Service Applications**.
3. Find and click the name of your new Reporting Services Service application. You could also click the application proxy group name **default** to add the proxy to default group rather than completing the following steps.
4. Select **Custom** in the selection box **Edit the following group of connections**.
5. Check the box for your proxy and click **Ok**.

More questions? [Try asking the Reporting Services forum](#)

Back up and restore Reporting Services SharePoint service applications

5/3/2018 • 5 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

This topic describes how to back up and restore a Reporting Services services application using SharePoint Central Administration or PowerShell.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

Before you begin

Limitations and restrictions

NOTE

Reporting Services service applications can partially be backed up and restored using the SharePoint backup and restore functionality. **Additional steps are required** and the steps are documented in this topic. Currently the backup process **does not** back up encryption keys and credentials for unattended execution accounts (UEA) or windows authentication to the Reporting Services database.

Recommendations

- Back up the encryption keys before starting the SharePoint backup. If you do not back up the encryption keys, then you will not be able to access your encrypted data, following the restore of the service application. You will need to delete your encrypted data.
- Verify if your Reporting Services service application is using UEA or Windows authentication for database access. If it is using either, verify what the proper credentials are so you can correctly configure the service application after the restore process.
- Review that the SharePoint backup log is created in the same folder as the backup file. The file is typically named **spbackup.log**

Back up the service application

Complete the following steps in order:

1. Back up the encryption keys
2. Back up the service application
3. Verify if you service application uses an UEA or Windows authentication for database access. If it does, make a note of the credentials so you can use them to configure the service application after it is restored.

Back up the encryption keys using SharePoint Central Administration

For information on backing up the Reporting Services encryption keys, see the "Encryption Keys" section of [Manage a Reporting Services SharePoint Service Application](#).

Back up the service application using SharePoint Central Administration

To back up the Service Application, complete the following steps:

1. In SharePoint Central Administration select **Perform a backup** in the **Backup and Restore** group.
2. Under the **Shared Services** node, expand **Shared Service Applications** and select your service application. It will have a type of **SQL Server Reporting Services Service Application**.
3. Select **Next**.
4. Type the path for the **Backup location:** and select **Start Backup**
5. Repeat the process above but instead of selecting the service application, expand the **Shared Services Proxies** node, and select service application proxy. It will have a type of **SQL Server Reporting Services Service Application Proxy**.

For more information, see the following topics in the SharePoint documentation:

[Back up a service application \(SharePoint Foundation 2010\) in the SharePoint documenttation](#).

[Back up a service application \(SharePoint Server 2010\)](#)

Verify execution account and database authentication

Execution Account: To verify if your service application is using an execution account:

1. In SharePoint Central Administration, select **Manage Service Applications** in the **Application Management** group.
2. Select the name of your service application and then select **Manage** in the SharePoint Ribbon.
3. Select **Execution Account**.
4. If an execution account is configured, you need to know the credentials when it is time to restore the service application backup. Do not proceed with the backup and restore procedure until you know the correct credentials.

Database Authentication: To verify if your service application is using Windows Authentication for the database authentication:

5. In SharePoint Central Administration, select **Manage Service Applications** in the **Application Management** group.
6. Select the name of your service application and then select **Properties** in the SharePoint Ribbon.
7. Review the **Reporting Services (SSRS) Service Database** section.
8. If Windows Authentication is configured, you need to know the credentials so you can configure the service application after you restore it. Do not proceed with the backup and restore procedure until you know the correct credentials.

Restore the service application

Complete the following steps in order:

1. Restore the Reporting Services service application.
2. Restore the encryption keys

3. If your service application was using an execution account or Windows authentication for database access, configure the credentials.

Restore the service application using SharePoint Central Administration

1. In SharePoint Central Administration select **Restore from a backup** in the **Backup and Restore** group.
2. Type the path to your backup file in **Backup Directory Location** box and select **Refresh**.
3. Select your service application backup from the **Top Component** list and then select **Next**.
4. Select your Reporting Services application and then select **Next**.
5. In the **Login Names and Passwords** section type the password for the login name. The login name box should be populated with the login the service application was using prior to the back up.
6. Select **Start Restore**.
7. Repeat the process above but instead of restoring the service application, expand the **Shared Services** node and then expand the **Shared Service Applications** node.

For more information, see the following topics in the SharePoint documentation:

[Restore a service application \(SharePoint Foundation 2010\)](#).

[Restore a service application \(SharePoint Server 2010\)](#).

Restore the encryption keys using SharePoint Central Administration

For information on restoring the Reporting Services encryption keys, see the "Encryption Keys" section of [Manage a Reporting Services SharePoint Service Application](#).

Configure the execution account and database authentication

Execution Account: If your service application was using an execution account complete the following steps to configure it:

1. In SharePoint Central Administration, select **Manage Service Applications** in the **Application Management** group.
2. Select the name of your service application and then select **Manage** in the SharePoint Ribbon.
3. Select **Execution Account**.
4. Type the account, password, and select the **Specify and Execution Account** box.
5. Select **OK**.



Database Authentication: If your service application was using Windows Authentication for the database authentication complete the following steps:

6. In SharePoint Central Administration select **Manage Service Applications** in the **Application Management** group.
7. Select the name of your service application and then select **Properties** in the SharePoint Ribbon.
8. Review the **Reporting Services (SSRS) Service Database** section.
9. Select **Windows Authentication**.
10. Type the account and password. Select **Use as Windows Credentials** if appropriate.
11. Select **Ok**

More questions? [Try asking the Reporting Services forum](#)

PowerShell cmdlets for Reporting Services SharePoint mode

5/16/2018 • 9 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

When you install SQL Server 2016 Reporting Services SharePoint mode, PowerShell cmdlets are installed to support report Servers in SharePoint mode. The cmdlets cover three categories of functionality.

- Installation of the Reporting Services SharePoint shared service and proxy.
- Provisioning and management of Reporting Services service applications and associated proxies.
- Management of Reporting Services features, for example extensions and encryption keys.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

Cmdlet summary

To run the cmdlets you need to open the SharePoint Management Shell. You can also use the graphical user interface editor that is included with Microsoft Windows, **Windows PowerShell Integrated Scripting Environment (ISE)**. For more information, see [Starting Windows PowerShell on Windows Server](#). In the following cmdlet summaries, the references to service application 'databases', refer to all of the databases created and used by a Reporting Services service application. This includes the configuration, alerting, and temp databases.

If you see an error message similar to the following when you type the PowerShell examples:

- Install-SPRSService : The term 'Install-SPRSService' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.

One of the following issues is occurring:

- Reporting Services SharePoint mode is not installed and therefore the Reporting Services cmdlets are not installed.
- You ran the PowerShell command in Windows PowerShell or Windows PowerShell ISE instead of the SharePoint Management Shell. Use the SharePoint Management shell or add the SharePoint Snap-in to the Windows PowerShell window with the following command:

```
Add-PSSnapin Microsoft.SharePoint.PowerShell
```

For more information see [Use Windows PowerShell to administer SharePoint 2013](#).

Open the SharePoint Management Shell and run cmdlets

1. Click the **Start** button
2. Click the **Microsoft SharePoint Products** group.
3. Click the **SharePoint Management Shell**.

To view command line help for a cmdlet use the PowerShell 'Get-Help' command at the PowerShell command prompt. For example:

```
Get-Help Get-SPRSServiceApplicationServers
```

Shared service and proxy cmdlets

The following table contains the PowerShell cmdlets for the Reporting Services SharePoint shared service.

CMDLET	DESCRIPTION
Install-SPRSService	<p>Installs and registers, or uninstalls, the Reporting Services shared service. This can be done only on the machine that has an installation of SQL Server Reporting Services in SharePoint mode. For installation, two operations occur:</p> <ul style="list-style-type: none">-The Reporting Services service is installed in the farm.-The Reporting Services service instance is installed to the current machine. <p>For Uninstallation, two operations occur:</p> <ul style="list-style-type: none">-The Reporting Services service is uninstalled from the current machine.-The Reporting Services service is uninstalled from the farm. <p>If there are any other machines in the farm that have the Reporting Services service installed, or if there are still Reporting Services service applications running in the farm, a warning message is displayed.</p>
Install-SPRSServiceProxy	<p>Installs and registers, or uninstalls, the Reporting Services service proxy in the SharePoint farm.</p>
Get-SPRSProxyUrl	<p>Gets the URL(s) for accessing the Reporting Services service.</p>
Get-SPRSServiceApplicationServers	<p>Gets all servers in the local SharePoint farm that contain an installation of the Reporting Services shared service. This cmdlet is useful for Reporting Services upgrades, to determine which servers run the shared service and therefore need to be upgraded.</p>

Service application and proxy cmdlets

The following table contains the PowerShell cmdlets for Reporting Services service applications and their associated proxies.

CMDLET	DESCRIPTION
Get-SPRSServiceApplication	Gets one or more Reporting Services service application objects.
New-SPRSServiceApplication	<p>Create a new Reporting Services service application and associated databases.</p> <p>LogonType Parameter: Specifies if the report server uses the SSRS Application Pool account or a SQL Server login to access the report server database. Valid values are:</p> <p>0 Windows Authentication</p> <p>1 SQL Server</p> <p>2 Application Pool Account (default)</p>
Remove-SPRSServiceApplication	Removes the specified Reporting Services service application. This will also remove the associated databases.
Set-SPRSServiceApplication	Edits the properties of an existing Reporting Services service application.
New-SPRSServiceApplicationProxy	Creates a new Reporting Services service application proxy.
Get-SPRSServiceApplicationProxy	Gets one or more Reporting Services service application proxies.
Dismount-SPRSDatabase	Dismounts the service application databases for a Reporting Services service application.
Remove-SPRSDatabase	Remove the service application databases for a Reporting Services service application.
Set-SPRSDatabase	Sets the properties of the databases associated to a Reporting Services service application.
Mount-SPRSDatabase	Mounts databases for a Reporting Services service application.
New-SPRSDatabase	Create new service application databases for the specified Reporting Services service application.
Get-SPRSDatabaseCreationScript	Outputs the database creation script to the screen for a Reporting Services service application. You can then run the script in SQL Server Management Studio.
Get-SPRSDatabase	<p>Gets one or more Reporting Services service application databases. Use the command to get the ID of service application database so you can use the Set-SPRSDatabase cmdlet to modify properties, for example the <code>querytimeout</code>.</p> <p>. See the example in this topic, Get and set Properties of the Reporting Service application database.</p>

CMDLET	DESCRIPTION
Get-SPRSDatabaseRightsScript	Outputs the database rights script to the screen for a Reporting Services service application. It will prompt for desired user and database then returns transact SQL you can run to modify permissions. You can then run this script in SQL Server Management Studio.
Get-SPRSDatabaseUpgradeScript	Outputs a database upgrade script to the screen. The script will upgrade Reporting Services service application databases to the database version of the current Reporting Services installation.

Reporting Services custom functionality cmdlets

CMDLET	DESCRIPTION
Update-SPRSEncryptionKey	Updates the encryption key for the specified Reporting Services service application and re-encrypts its data.
Restore-SPRSEncryptionKey	Restores a previously backed up encryption key for a Reporting Services service application.
Remove-SPRSEncryptedData	Delete the encrypted data for the specified Reporting Services service application.
Backup-SPRSEncryptionKey	Backs up the encryption key for the specified Reporting Services service application.
New-SPRSExtension	Registers a new extension with a Reporting Services service application.
Set-SPRSExtension	Sets the properties of an existing Reporting Services extension.
Remove-SPRSExtension	Removes an extension from a Reporting Services service application.

CMDLET	DESCRIPTION
Get-SPRSExtension	<p>Gets one or more Reporting Services extensions for a Reporting Services service application.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> Delivery DeliveryUI Render Data Security Authentication EventProcessing ReportItems Designer ReportItemDesigner ReportItemConverter ReportDefinitionCustomization
Get-SPRSSite	<p>Gets the SharePoint sites based on whether the "ReportingService" feature is enabled. By default, sites that enable the "ReportingService" feature are returned.</p>

Basic samples

Return a list of cmdlets that contain 'SPRS' in the name. This will be the full list of Reporting Services cmdlets.

```
Get-command -noun *SPRS*
```

Or with a little more detail, piped to a text file named commandlist.txt

```
Get-command -noun *SPRS* | Select name, definition | Format-List | Out-File c:\commandlist.txt
```

Install the Reporting Services SharePoint service and service proxy.

```
Install-SPRSService
```

```
Install-SPRSServiceProxy
```

Start the Reporting Services service

```
get-spserviceinstance -all |where {$_.TypeName -like "SQL Server Reporting*"} | Start-SPServiceInstance
```

Type the following command from the SharePoint Management Shell to return a filtered list of rows from the a log file. The command will filter for lines that contain "ssrscustomactionerror". This example is looking at the log file created when the rssharepoint.msi was installed.

```
Get-content -path C:\Users\testuser\AppData\Local\Temp\rs_sp_0.log | select-string "ssrscustomactionerror"
```

Detailed samples

In addition to the following samples, see the section "Windows PowerShell Script" in the topic [Windows PowerShell script for Steps 1–4](#).

Create a reporting services service application and proxy

This sample script completes the following tasks:

1. Create a Reporting Services service application and proxy. The script assumes the application pool "My App Pool" already exists.
2. Add the proxy to the default proxy group
3. Grant the service app access to the port 80 web app's content database. The script assumes site `http://sitename` already exists.

```
# Create service application and service application proxy
$appPool = Get-SPServiceApplicationPool "My App Pool"
$serviceApp = New-SPRSServiceApplication "My RS Service App" -ApplicationPool $appPool
$serviceAppProxy = New-SPRSServiceApplicationProxy -Name "My RS Service App Proxy" -ServiceApplication $serviceApp

# Add service application proxy to default proxy group. Any web application that uses the default proxy group
will now be able to use this service application.
Get-SPServiceApplicationProxyGroup -default | Add-SPServiceApplicationProxyGroupMember -Member $serviceAppProxy

# Grant application pool account access to the port 80 web application's content database.
$webApp = Get-SPWebApplication "http://sitename"
$appPoolAccountName = $appPool.ProcessAccount.LookupName()
$webApp.GrantAccessToProcessIdentity($appPoolAccountName)
```

Review and update a reporting services delivery extension

The following PowerShell script example, updates the configuration for the report server e-mail delivery extension for the service application named `My RS Service App`. Update the values for the SMTP server (

`<email server name>`) and the FROM email alias (`<your FROM email address>`).

```
$app=get-sprsserviceapplication -Name "My RS Service App"
$emailCfg = Get-SPRSExtension -identity $app -ExtensionType "Delivery" -name "Report Server Email" | select -
ExpandProperty ConfigurationXml
$emailXml = [xml]$emailCfg
$emailXml.SelectSingleNode("//SMTPServer").InnerText = "<email server name>"
$emailXml.SelectSingleNode("//SendUsing").InnerText = "2"
$emailXml.SelectSingleNode("//SMTPAuthenticate").InnerText = "2"
$emailXml.SelectSingleNode("//From").InnerText = '<your FROM email address>'
Set-SPRSExtension -identity $app -ExtensionType "Delivery" -name "Report Server Email" -ExtensionConfiguration $emailXml.OuterXml
```

In the above example if you did not know the exact name of the service application, you could rewrite the first statement to get the service application based on a search of the partial name. For example:

```
$app=get-sprsserviceapplication | where {$_.name -like " ssrs_testapp *"}
```

The following script will return the current configuration values for the report server e-mail delivery extension for the service application named "Reporting Services Application". The first step sets the value of the variable \$app to the object of the service application that has a name of " My RS Service App "

The second statement will Get the 'Report Server Email' delivery extension for the service application object in variable \$app, and select the configurationXML

```
$app=get-sprsserviceapplication -Name "Reporting Services Application"  
Get-SPRSExtension -identity $app -ExtensionType "Delivery" -name "Report Server Email" | select -  
ExpandProperty ConfigurationXml
```

You can also rewrite the above two statements as one:

```
get-sprsserviceapplication -Name "Reporting Services Application" | Get-SPRSExtension -ExtensionType  
"Delivery" -name "Report Server Email" | select -ExpandProperty ConfigurationXml
```

Get and set properties of the Reporting Service application database

The following example first returns a list of the databases and properties so you can determine the database guid (ID) that you then supply to the set command. For a full list of the properties, use `Get-SPRSDatabase | format-list`.

```
get-SPRSDatabase | select id, querytimeout,connectiontimeout, status, server, ServiceInstance
```

The following is an example of the output. Determine the ID for the database you want to modify and use the ID in the SET cmdlet.

- Id : 56f8d1bc-cb04-44cf-bd41-a873643c5a14

QueryTimeout : 120

ConnectionTimeout : 15

Status : Online

Server : SPServer Name=uetestb01

ServiceInstance : SPDatabaseServiceInstance

```
Set-SPRSDatabase -identity 56f8d1bc-cb04-44cf-bd41-a873643c5a14 -QueryTimeout 300
```

To verify the value is set, run the GET cmdlet again.

```
Get-SPRSDatabase -identity 56f8d1bc-cb04-44cf-bd41-a873643c5a14 | select id, querytimeout,connectiontimeout,  
status, server, ServiceInstance
```

List Reporting Services data extensions

The following example loops through each Reporting Services service application and lists the current data extensions for each.

```
$apps = Get-SPRSServiceApplication
foreach ($app in $apps)
{
Write-host -ForegroundColor "yellow" Service App Name $app.Name
Get-SPRSExtension -identity $app -ExtensionType "Data" | select name,extensiontype | Format-Table -AutoSize
}
```

Example output:

- Name ExtensionType

```
-----
```

```
SQL Data
```

```
SQLAZURE Data
```

```
SQLPDW Data
```

```
OleDb Data
```

```
OleDb-MD Data
```

```
ORACLE Data
```

```
ODBC Data
```

```
XML Data
```

```
SHAREPOINTLIST Data
```

Change and list Reporting Services subscription owners

See [Use PowerShell to change and list Reporting Services subscription owners and run a subscription.](#)

Next steps

[Use PowerShell to change and list Reporting Services subscription owners and run a subscription](#)




[CheckList: Use PowerShell to verify Power Pivot for SharePoint](#)

[Get help SQL Server PowerShell](#)

More questions? [Try asking the Reporting Services forum](#)

Configuration and administration of a SQL Server Reporting Services (SSRS) report server

8/20/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

SQL Server Reporting Services is a server-based reporting platform that provides a full range of ready-to-use tools and services to help you create, deploy, and manage reports for your organization, as well as programming features that enable you to extend and customize your reporting functionality. You can integrate your reporting environment with a SharePoint product to experience the benefits of using the collaborative environment provided by SharePoint sites.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

Use the following sections to help you understand concepts, deployment scenarios, procedures, and more for integrating your Reporting Services environment with a SharePoint product or technology:

- Menu options in a SharePoint document library
 - [Data Alert Manager for SharePoint Users](#)
 - [Create and Manage Subscriptions for SharePoint Mode Report Servers](#)
 - [Update Credentials in Report Data Sources from a SharePoint Site](#)
 - [Manage Shared Datasets](#)
 - [Set Parameters on a Published Report \(Reporting Services in SharePoint Integrated Mode\)](#)
 - [Set Processing Options \(Reporting Services in SharePoint Integrated Mode\)](#)
- [Reporting Services Site Collection Features](#)
- [Activate the Report Server and Power View Integration Features in SharePoint](#)
- [Reporting Services Site Settings and Site Features\(SharePoint Mode\)](#)
- [Activate the Report Server File Sync Feature in SharePoint Central Administration](#)
- [Add Reporting Services Content Types to a SharePoint Library](#)
- [Local Mode vs. Connected Mode Reports in the Report Viewer \(Reporting Services in SharePoint Mode\)](#)
- [Upload Documents to a SharePoint Library \(Reporting Services in SharePoint Mode\)](#)
- [Set Processing Options \(Reporting Services in SharePoint Integrated Mode\)](#)

More questions? [Try asking the Reporting Services forum](#)

Reporting Services site collection features

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

Reporting Services SharePoint mode provides three SharePoint site collection features. The features support the general Reporting Services SharePoint mode reporting environment, Power View, a feature of the SQL Server 2016 Reporting Services Add-in, and management operations for Reporting Services in SharePoint Central Administration.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

Site collection features

The following table describes the Reporting Services site collection features.

FEATURE	DESCRIPTION
Report Server Central Administration Feature	<p>Enables Features for managing integration with a Reporting Services report server. This feature is only installed and used in the SharePoint Central Administration site collection.</p> <p>The Report server integration feature is automatically activated in SharePoint Central Administration Site collection after you install the Microsoft SQL Server 2016 Reporting Services (SSRS) Add-in for SharePoint products. In some situations, you need to manually activate the feature. To activate the report server feature, use the Reporting Services pages in SharePoint Central Administration's Site Settings page.</p> <p>The SQL Server 2008 R2 Reporting Services version and later of the Add-in for SharePoint products activate the report server integration feature for all existing site collections when the Add-in is installed. Additionally, the feature is automatically active for new site collections.</p>
Report Server Integration Feature	<p>Enables rich reporting using Microsoft Reporting Services</p> <p>This feature is Active by default.</p>

FEATURE	DESCRIPTION
Power View Integration Feature	<p>Enables interactive data exploration and visual presentation against Power Pivot work books and Analysis services tabular databases.</p> <p>The feature can be accessed by the context menus of the following data sources:</p> <p>.rdlx</p> <p>.rsds</p> <p>.bism connection file</p> <p>If Power View does not appear in the context menus, verify the Power View Integration Feature is activated.</p> <p>This feature is deactivated by default.</p>

Next steps




[Activate the Report Server and Power View Integration Features in SharePoint Reporting Services Site Settings and Site Features\(SharePoint Mode\)](#)

[Activate the Report Server File Sync Feature in SharePoint Central Administration](#)

More questions? [Try asking the Reporting Services forum](#)

Activate the report server and Power View integration features in SharePoint

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

The Reporting Services site collection features are activated by default after you install the Microsoft SQL Server 2016 Reporting Services (SSRS) Add-in for SharePoint products. In some situations, you need to manually activate the features.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

If you install the Reporting Services add-in for SharePoint 2010 Products after the installation of the SharePoint product, then the Report Server integration feature and the Power View integration feature will only be activated for root site collections. For other site collections, you need to manually activate the features. For example, if you have a site collection of **[http://\[my server name\]/sites/\[site collection name\]](http://[my server name]/sites/[site collection name])** you need to manually activate the Reporting Services site collection features.

When there is no root site collection, the Reporting Services add-in will log a message similar to the following.

"SharePoint web app 80 does not have root site collection"

The message is found in the add-in installation log, named "RS_SP_#.log" where # is an incrementing number. The log file is found in the current users Temp folder, for example C:\Users\[user name]\AppData\Local\Temp. For more information on logging options with the add-in, see [Install or Uninstall the Reporting Services Add-in for SharePoint](#).

Activate the Report Server and Power View integration site collection features

1. Open your browser to the site where you want the Reporting Services features active.
2. Click **Site Actions**.
3. Click **Site Settings**.
4. Click **Site Collection Features** in the Site Collection Administration Group.
5. Find **Report Server Integration Feature** or **Power View Integration Feature** in the list.
6. Click **Activate**.

To deactivate the features, you can use the same procedure, but click **Deactivate** rather than **Activate**.

Activate or Deactivate Reporting Services central administration site collection feature

1. Open your browser to SharePoint Central Administration.
2. Click **Site Actions**.
3. Click **Site Settings**.
4. Click **Site Collection Features** in the Site Collection Administration Group.
5. Find **Report Server Central Administration Feature** in the list.
6. Click **Activate**.

To deactivate the feature, you can use the same procedure, but click **Deactivate** rather than **Activate**.


Next steps

After the feature is activated, you can continue with server integration.

More questions? [Try asking the Reporting Services forum](#)

Reporting Services site settings and site features (SharePoint mode)

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

Reporting Services SharePoint mode has several site level custom features and site feature that can be managed from the SharePoint Site Settings page. The settings are site wide and affect all Reporting Services service applications. You must have Content Manager and System Administrator permissions to view this page.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

SITE SETTING	DESCRIPTION
Reporting Services Site Settings	Site-wide settings described in this topic.
Manage Data Alerts	Management of the Data Alerting feature.
Report Server File Sync	<p>A Site level feature that is deactivated by default.</p> <p>Synchronizes Report Server files (.rdl, .rsds, .smdl, .rsd, .rsc, .rdlx) from a SharePoint document library to the report server when files are added or updated in the directly within the document library.</p> <p>For more information, see Activate the Report Server File Sync Feature in SharePoint Central Administration</p>

Open the Reporting Services site settings page

1. From the SharePoint site's **Site Actions** menu, select **Site Settings**.
2. In the **Reporting Services** section, select **Reporting Services Site Settings**.

Options for Reporting Services site settings

OPTION	DESCRIPTION
Enable RSClientPrint ActiveX control download	The control displays a custom print dialog box that supports features common to other print dialog boxes, including print preview, page selections for specifying specific pages and ranges, page margins, and orientation. For more information on the control, see Using the RSClientPrint Control in Custom Applications




OPTION	DESCRIPTION
Enable remote errors in local mode	<p>Show or hide detailed error messages on remote computers when running in local mode. If you see an error message similar to the following, then enabling remote errors may be useful:</p> <div data-bbox="826 331 1434 412"> <p>For more information about this error navigate to the report server on the local server machine or enable remote errors</p> </div>
Enable accessibility metadata for reports	<p>Turn on accessibility metadata in the HTML output for reports</p>
Enable Exact Data Visualization Fit Sizing for Reports	<p>Configure data visualization fit behavior when inside a tablix, to fix exactly. This includes chart, gauge, and map. When disabled the behavior is for data visualizations to fit approximately, which may leave some white space. This setting only applies to rendering in the Report Viewer web part. To manage this behavior for server-side rendering, you need to modify the rsreportserver.config file. For more information, see the following:</p> <p>RsReportServer.config Configuration File.</p> <p>Customize Rendering Extension Parameters in RSReportServer.Config.</p> <p>HTML Device Information Settings.</p> <p>Enabling Exact may have performance impact because the processing to determine the exact size may take longer than an approximate fit.</p>

See also

[Manage a Reporting Services SharePoint Service Application](#)

Activate the report server file sync feature in SharePoint

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016 and later)  SharePoint  Power BI Report Server

The Reporting Services Report Server File Sync feature utilizes SharePoint event handlers to synchronize the report server catalog with items in document libraries. This feature is beneficial when users frequently upload published report items directly to SharePoint document libraries. If the file Sync feature is not activated, content will still be synchronized but not as frequently.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

The File Sync feature can be activated in SharePoint Site Administration after you install the SQL Server 2016 Reporting Services (SSRS) Add-in for SharePoint products.

This feature can be manually activated and deactivated per site but not at the site collection level.

Prerequisites

The Reporting Services Add-in for SharePoint must be installed. If the add-in is not installed the file sync feature will not be visible in the site feature list.

To verify installation, view the list of installed applications in Microsoft Windows **Control Panel**. If the Reporting Services Add-in is installed, follow the instructions in this topic to activate the report server file sync feature.

To activate or deactivate the Reporting Services file sync feature on a site

1. From the main page of your site, click the **Site Actions** menu and click **Site Settings**.
2. In the **Site Actions** click **Manage Site Features**.
3. Find **Report Server File Sync** in the list.
4. Click **Activate**.

NOTE

To deactivate the Report Server file sync feature, you can use the same procedure but click **Deactivate**.

See also

[Troubleshoot Report Parts \(Report Builder and SSRS\)](#)

[Activate the Report Server and Power View Integration Features in SharePoint](#)




[Install or Uninstall the Reporting Services Add-in for SharePoint](#)

[Install or Uninstall the Reporting Services Add-in for SharePoint](#)

More questions? [Try asking the Reporting Services forum](#)

Add Reporting Services content types to a SharePoint library

5/3/2018 • 5 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

Reporting Services provides predefined SharePoint content types that are used to manage shared data source (.rsds) files, report models (.smdl), and Report Builder report definition (.rdl) files. Adding a **Report Builder Report**, **Report Model**, and **Report Data Source** content type to a library enables the **New** command so that you can create new documents of that type.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

To add content types to a library, you must be a site administrator or have Full Control level of permission.

The Reporting Services content types and content type management will automatically be enabled in all document libraries for existing site collections created from the following site template types:

- **Business Intelligence Center**

Sites created after the Reporting Services integration will not have the Reporting Services content types enabled.

TIP

If you have **not** previously configured content types for a library, first enable management of content types, then enable the Reporting Services content types. See the procedures on enabling content type management in a single document library.

Short video: (SSRS) Enabling Content Types in SharePoint2010.wmv (<http://www.youtube.com/watch?v=yqhm3DrtT1w>).

In this topic:

- [Enable content types in all document libraries in an existing BI center](#)
- [To enable content type management for a single document library \(SharePoint 2013\)](#)
- [To add Reporting Services content types \(SharePoint 2013\)](#)
- [To enable content type management for a single document library \(SharePoint 2010\)](#)
- [To add report server content types \(SharePoint 2010\)](#)
- [To enable Content types and content management for multiple BI sites](#)

Enable content types in all document libraries in an existing BI center

1. To enable the content types and content management in all document libraries in an existing **Business**

Intelligence Center site, you can toggle the Reporting Services integration feature.

2. Go to **Site settings**.

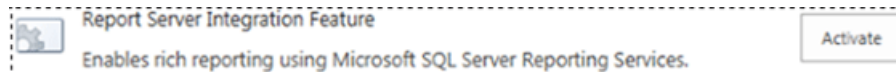
- In SharePoint 2013, click the **Settings** icon. ⚙
- In SharePoint 2010, click **Site Actions**, then click **Site Settings**.

3. Click **Site collection features**.

4. Find the **Report Server Integration Feature** and click **Deactivate**.



5. Refresh the browser then click **Activate** for the **Report Server Integration Feature**.



To enable content type management for a single document library (SharePoint 2013)

1. Open the library for which you want to enable multiple content types.

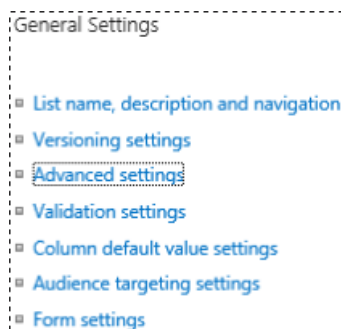
2. Click **Library** in the ribbon.



3. On the **Library** ribbon, click **Library Settings**. If you do not see **Library Settings** or the button is disabled, you do not have permission to configure library settings, including content types.



4. In the **General Settings** section, click **Advanced settings**.



5. In the **Content Types** section, select **Yes** to allow management of content types.

6. Click **OK**.

To add Reporting Services content types (SharePoint 2013)

1. Open the library for which you want to add Reporting Services content types.

2. On the ribbon, click **Library**.

3. Click **Library Settings**.

4. Under **Content Types**, click **Add from existing site content types**.

5. In **Select site content types from**, select **SQL Server Reporting Services Content Types**.
6. In the **Available Site Content Types** list, click **Report Builder**, and then click **Add** to move the selected content type to the **Content types to add** list.
7. To add **Report Model** and **Report Data Source** content types, repeat the previous step.
8. When you finish adding content types, click **OK**.

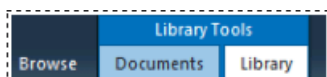
NOTE

If the Reporting Services content type group **SQL Server Reporting Services Content Types** is not visible on the **Add Content Types** page, one of the following conditions is true:

- The Reporting Services add-in for SharePoint products has not been installed. For more information, see [Install or Uninstall the Reporting Services Add-in for SharePoint](#). The topic includes information on installing the add-in and stepping through a files only installation of the add-in to work around issues.
- The add-in is installed but the site collection feature **Report Server Integration Feature** is not active. Verify the site collection feature in **Site Settings**.
- All of the Reporting Services content types have already been added to the library. If all the content types are part of a library, then the group is removed from the **Add Content Types** page. If you delete one or more of the Reporting Services content types, then the group **SQL Server Reporting Services Content Types** will be visible on the **Add Content Types** page.

To enable content type management for a single document library (SharePoint 2010)

1. Open the library for which you want to enable multiple content types. On the library menu bar, you should see the following menus: **New**, **Upload**, **Actions**, and **Settings**. If you do not see **Settings**, you do not have permission to add a content type.
2. On the **Library Tools** ribbon, click **Library**.



3. On the **Settings** ribbon group, click **Library Settings**.
4. Under **General Settings**, click **Advanced settings**.
5. In the **Content Types** section, select **Yes** to allow management of content types.
6. Click **OK**.

To add report server content types (SharePoint 2010)

1. Open the library for which you want to add Reporting Services content types.
2. On the **Library Tools** ribbon tabs, click the **Library** tab.
3. On the **Settings** ribbon group, click **Library Settings**.
4. Under **Content Types**, click **Add from existing site content types**.
5. In the **Select Content Types** section, in **Select site content types from**, click the arrow to select **SQL**

Server Reporting Services Content Types.

6. In the **Available Site Content Types** list, click **Report Builder**, and then click **Add** to move the selected content type to the **Content types to add** list.
7. To add **Report Model** and **Report Data Source** content types, repeat the previous step.
8. When you finish adding content types, click **OK**.

To enable content types and content management for multiple BI sites

1. For SQL Server Reporting Services 2008 and 2008 R2 report servers, you can enable content types and content management for multiple Business Intelligence Center sites:
2. In SharePoint Central Administration, click **General Applications settings**. In the **SQL Server Reporting Services (2008 and 2008 R2)** section, click **Reporting Services Integration**.



3. Click **Activate feature in all existing site collections**.





4. Click **Ok**.

See also

[SharePoint Site and List Permission Reference for Report Server Items](#)
[Start Report Builder](#)

Local mode vs. connected mode reports in the Report Viewer

5/3/2018 • 3 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

Reporting Services reports can be configured to run in either *local mode* or *connected mode*, which leverages a Reporting Services report server. Instead, you can use the Report Viewer to directly render reports from SharePoint when the data extension supports local mode reporting. This approach is called *local mode*. In previous versions of Reporting Services, the SharePoint farm was required to be connected to a Reporting Services report server configured in SharePoint mode so the Report Viewer control could render reports. This approach is called *remote mode* or *connected mode*.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

In *local mode* there is no Reporting Services report server. You must install the Reporting Services add-in for SharePoint products, but no Reporting Services report server is required. With Local mode, users can view reports but will not have access to server side features such as subscriptions and data alerts.

Local mode vs connected mode and supported extensions

Local mode: When you have a data extension that supports local mode, the Report Viewer directly renders reports from SharePoint. In *local mode* there is no Reporting Services report server. You must install the Reporting Services add-in for SharePoint products, but no Reporting Services report server is required. With Local mode, users can view reports but will **not** have access to server side features such as subscriptions and data alerts.

Connected mode, also called *remote mode* requires a Reporting Services report server in SharePoint mode, connected to the SharePoint farm so the Report Viewer control could render reports..

The following is a list of the data processing extensions that support local mode reporting:

- Microsoft Access 2010 reporting extension. For more information on Access Services, see [Use Access Services with SQL Reporting Services: Installing SQL Server 2008 R2 Reporting Services Add-In \(SharePoint Server 2010\)](#).
- The Reporting Services SharePoint list data extension. For more information on the SharePoint List Data Extension, see [Data Sources Supported by Reporting Services \(SSRS\)](#)

Custom data processing extensions can also be developed to support local mode. For more information, see [Implementing a Data Processing Extension](#).

Local mode supports rendering reports that have an embedded data source or a shared data source from an .rsds file. However, you cannot manage the report or its associated data source. If you try to do this, you will receive an error that this is not supported in local mode. Managing data sources in the SharePoint site is supported in only connected mode.

NOTE

As with previous versions, you cannot embed user names and passwords in the .rsds file.

Configure local mode and access services with SharePoint 2013

You can configure your SharePoint 2013 farm to support existing Access 2010 web databases and Reporting Services local mode. For more information, see [Set up and configure Access Services 2010 for web databases in SharePoint Server 2013](#).

It is not possible to create new Access web databases for SharePoint 2013. Access 2013 uses a new type of database, *Access web app* that you build in Access, then use and share with others as a SharePoint app in a web browser.

For more information, see the following.

- [What's new in Access 2013](http://office.microsoft.com/access-help/what-s-new-in-access-2013-HA102809500.aspx) (<http://office.microsoft.com/access-help/what-s-new-in-access-2013-HA102809500.aspx>).
- [Basic tasks for an Access app](http://office.microsoft.com/access-help/basic-tasks-for-an-access-app-HA102840210.aspx?CTT=5&origin=HA102809500) (<http://office.microsoft.com/access-help/basic-tasks-for-an-access-app-HA102840210.aspx?CTT=5&origin=HA102809500>).

Configure local mode reporting with SharePoint 2010

Local mode requires ASP.NET session state. The installation of Access services will enable ASP.Net sessions state. You can also enable using PowerShell.

1. Open the SharePoint 2010 Management Shell.
2. Type the following command:

```
- Enable-SPSessionStateService
```

3. When prompted, type the name of your database.
4. Perform an IIS reset.

For more information, see [Use Access Services with SQL Reporting Services: Installing SQL Server 2008 R2 Reporting Services Add-In \(SharePoint Server 2010\)](#) and [Enable-SPSessionStateService](#).

Connected mode

For the latest information on using ADS extension with Reporting Services connected mode, see [Access Services Report in SharePoint Site shows error in data extension 'ADS'](#).




See also

[Data sources supported by Reporting Services](#)

More questions? [Try asking the Reporting Services forum](#)

Upload documents to a SharePoint library (Reporting Services in SharePoint mode)

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

You can upload report definitions and report models to a SharePoint library. When uploading a report server item, you must select a library or a folder within a library. You cannot upload a report server item to a list or page.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

You cannot upload a data source (.rds) file. However, you can publish .rds files from a design tool, such as Report Designer, to a SharePoint library. During publication, a new .rds file is created from the original .rds file in the solution. You can also create a new .rds file in a SharePoint library and then set data source connection properties in the uploaded reports and models to use the new connection.

NOTE

The report server must be configured for SharePoint mode, and the instance of the SharePoint product must have the Reporting Services Add-in that provides program files for storing and accessing report server items from a SharePoint site.

To upload a document to a library, you must have the "Add Items" permission at the site level. If you are using default security settings, this permission is granted to members of the **Owners** group who have the Full Control level of permission and to the **Members** group who have the Contribute level of permission.

Add a report definition or report model to a library

1. Open the library or a folder within a library. If the library is not already open, click its name on the Quick Launch. If the name of your library does not appear, click **View All Site Content**, and then click the name of your library.
2. On the **Upload** menu, click **Upload document**.
3. To upload a single report or report model file, select a report definition (.rdl) or report model (.smdl) file and then click **OK**.

If the report definition uses a shared data source (.rds) file to store connection information to an external data source, you can upload the .rdl and the .rds file at the same time. To do this, click **Upload Multiple Documents**, specify both files, and then click **OK**.

If you upload a report that contains references to shared data sources, report models, or subreports, the references will be broken in the report when the files are uploaded. For more information about how to reset the references, see [Create and Manage Shared Data Sources \(Reporting Services in SharePoint Integrated Mode\)](#).

When you upload a report, it runs on demand when you open it, retrieving live data from the data source.

You can configure the report to retrieve data on a schedule or use cached data. For more information, see [Set Processing Options \(Reporting Services in SharePoint Integrated Mode\)](#).

A report can contain parameters so that users can filter data. You can configure the parameters to use specific values or change how they appear to the user. For more information, see [Set Parameters on a Published Report \(Reporting Services in SharePoint Integrated Mode\)](#).

See also

[Publish a Report to a SharePoint Library](#)




[Publish a Shared Data Source to a SharePoint Library](#)

[Granting Permissions on Report Server Items on a SharePoint Site](#)

More questions? [Try asking the Reporting Services forum](#)

Set processing options (Reporting Services in SharePoint integrated mode)

5/3/2018 • 4 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

You can set processing options on a Reporting Services report to determine when data processing occurs. You can also set a time-out value for report processing, and options that determine whether report history is enabled for the current report.

- You can run a report as a report snapshot to prevent the report from being run at arbitrary times (for example, during a scheduled backup). A report snapshot is usually created and subsequently refreshed on a schedule, allowing you to time exactly when report and data processing will occur. If a report is based on queries that take a long time to run, or on queries that use data from a data source that you prefer no one access during certain hours, you should run the report as a snapshot.
- A report server can cache a copy of a processed report and return that copy when a user opens the report. Caching is a performance-enhancement technique. Caching can shorten the time required to retrieve a report if the report is large or accessed frequently.
- Report history is a collection of previously run copies of a report. You can use report history to maintain a record of a report over time. Report history is not intended for reports that contain confidential or personal data. For this reason, report history can include only those reports that query a data source using a single set of credentials (either stored credentials or credentials used for unattended report execution) that are available to all users who run a report.

Reporting Services integration with SharePoint uses the check out and check in content management features of SharePoint to save updates to Reporting Services content types. This includes the creation of report snapshots. Therefore if you have enabled versioning on a document library, you will see the report version updated when a new report history snapshot is created. This is a side-effect of updating snapshots. When a snapshot is updated it causes the LastExecution property of the report to change and that will cause a change in the version of the report.

- You can specify time-out values to set limits on how system resources are used.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

Set data refresh options

1. Point to a report in the library.
2. Click the down arrow, and select **Manage processing options**.
3. In **Data Refresh Options**, click **Use snapshot data**. If you see "This report can not run from a snapshot because one or more of the data sources credentials are not stored", the report is not configured to run unattended and you must modify the data sources to use stored credentials before setting this option.

4. In **Data Snapshot Options**, select **Schedule data processing**.
5. Select **On a shared schedule** if you have an existing shared schedule that you want to use, otherwise click **On a custom schedule**, and then click **Configure**.
6. Select frequency, schedule, and start and end dates, and then click **OK**.
7. In **Data Snapshot Options**, select **Create or update the snapshot when this page is saved** if you want to immediately create snapshot data to use with the report, without waiting for the scheduled data processing to occur.

Set report caching options

1. Point to a report in the library.
2. Click the down arrow, and select **Manage processing options**.
3. In **Data Refresh Options**, click **Use cached data**. If you see "This report can not be cached because one or more of the data sources credentials are not stored", the report is not configured to run unattended and you must modify the data sources to use stored credentials before setting this option.
4. In **Cache Options**, specify how the cache will expire:
 - Enter a number of minutes after which the cache will expire.
 - Use a shared schedule to clear the cache at times specified in the schedule.
 - Create a custom schedule to clear the cache at a time that you specify.

Set processing time-out values

1. Point to a report in the library.
2. Click the down arrow, and select **Manage processing options**.
3. In **Processing Time-out**, select **Use site default setting** if you want to use the value specified at the report server level. Otherwise, select **Do not time out report processing** or **Limit report processing in seconds** if you want to override that value with no time-out or different time-out values.

Set report history options and limits

1. Point to a report in the library.
2. Click the down arrow, and select **Manage processing options**.
3. In **History Snapshot Options**, specify how and when data processing occurs and is saved.
4. In **History Snapshot Limits**, select **Use site default setting** if you want to use the value specified at the report server level. Otherwise, select **Do not limit the number of snapshots** or **Limit number of snapshots** to specify a custom value.

Set database timeout

- Use Windows PowerShell to set the database timeout of a SharePoint report server. For more information, see the "Get and set Properties of the Reporting Service Application Database" section of [PowerShell cmdlets for Reporting Services SharePoint Mode](#).

Next steps

[Set Report Processing Properties](#)




[Caching Reports](#)

[Setting Time-out Values for Report and Shared Dataset Processing](#)

More questions? [Try asking the Reporting Services forum](#)

Report Viewer web part on a SharePoint site - Reporting Services

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016 and later)  SharePoint  Power BI Report Server

The Report Viewer web part is a custom web part. You can use the web part to view, navigate, print, and export reports on a report server within a SharePoint site. The Report Viewer web part is associated with report definition (.rdl) files that are processed by a Microsoft SQL Server Reporting Services report server.

The latest Report Viewer web part can also service paginated reports deployed to Power BI Report Server. The web part does not work with Power BI reports.

Why the Report Viewer web part is re-introduced

The Report Viewer web part was available as part of the Reporting Services Add-in for SharePoint products. The web part was specific for report servers in SharePoint integrated mode. SharePoint integrated mode was deprecated after SQL Server 2016.

Starting with SQL Server 2017, there's only one installation mode for Reporting Services: **Native mode**. You could embed all reports types using a Page Viewer web part using the *rs:Embed=true* URL parameter. Embedding reports into SharePoint pages is an integration story requested by customers and the updated Report Viewer web part enables this scenario for paginated reports.

While the Page Viewer web part suffices to embed a paginated report into a SharePoint page, the updated Report Viewer web part offers additional features.

- Show/hide specific toolbar buttons
- Override report parameter values
- Connect Filter web parts to report parameters

Download the Report Viewer web part solution package

The Report Viewer web part is available on the Microsoft Download Center.

[Download Report Viewer web part solution package](#)

Considerations and limitations

The items listed are specific to the updated Report Viewer web part.

- The web part can only be used on *classic* SharePoint pages.
- Only paginated (RDL) reports are supported for embedding in the Report Viewer web part. If you are looking to embed Power BI reports or mobile reports, you can use the *rs:Embed=true* URL parameter.

Next steps

To get started with the updated Report Viewer web part, see [Deploy the Report Viewer web part on a SharePoint site](#).

Deploy the SQL Server Reporting Services Report Viewer web part on a SharePoint site

5/3/2018 • 4 minutes to read • [Edit Online](#)

APPLIES TO: ✓ SQL Server Reporting Services (2016 and later) ✓ SharePoint ✓ Power BI Report Server

The Report Viewer web part is a custom web part that can be used to view SQL Server Reporting Services (native mode) reports within your SharePoint site. You can use the web part to view, navigate, print, and export reports on a report server. The Report Viewer web part is associated with report definition (.rdl) files that are processed by a SQL Server Reporting Services report server or a Power BI Report Server. This Report Viewer web part cannot be used with Power BI reports hosted in Power BI Report Server.

Use the following instructions to manually deploy the solution package that add the Report Viewer web part to a SharePoint Server 2013 or SharePoint Server 2016 environment. Deploying the solution is a required step for configuring the web part.

The Report Viewer web part is a standalone solution package and is not associated with SharePoint integrated mode for SQL Server Reporting Services.

Requirements

IMPORTANT

You currently cannot install this web part if you already have Reporting Services SharePoint integrated mode configured.

Support SharePoint Server versions:

- SharePoint Server 2016
- SharePoint Server 2013

Support Reporting Services versions:

- SQL Server 2008 Reporting Services (Native mode) and later.
- Power BI Report Server

Download the Report Viewer web part solution package

The Report Viewer web part is available on the Microsoft Download Center.

[Download Report Viewer web part solution package](#)

Deploy the farm solution

This section shows you how to deploy the solution package to your SharePoint farm. This task only needs to be performed once.

1. On a SharePoint server, open a SharePoint Management Shell using the **Run as Administrator** option.
2. Run [Add-SPSolution](#) to add the farm solution.

```
Add-SPSolution -LiteralPath "{path to file}\ReportViewerWebPart.wsp"
```

The cmdlet returns the name of the solution, its solution ID, and Deployed=False. In the next step, you will deploy the solution.

3. Run the [Install-SPSolution](#) cmdlet to deploy the farm solution.

SharePoint 2013

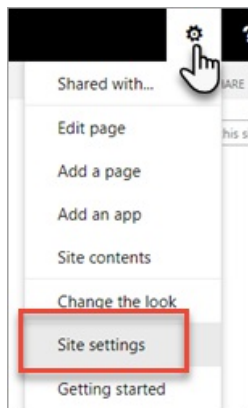
```
Install-SPSolution -Identity ReportViewerWebPart.wsp -CompatibilityLevel "14,15" -GACDeployment -  
WebApplication {URL to web application}
```

SharePoint 2016

```
Install-SPSolution -Identity ReportViewerWebPart.wsp -GACDeployment -WebApplication {URL to web  
application}
```

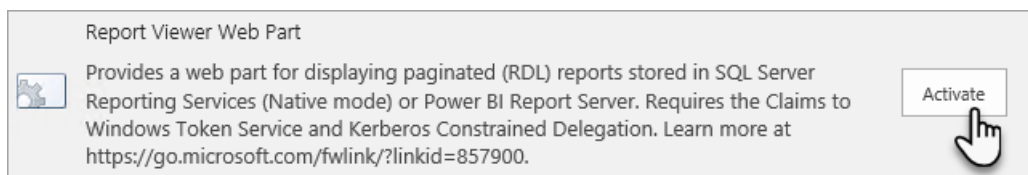
Activate feature

1. In your SharePoint site, select the **gear** icon in the upper left and select **Site Settings**.



By default, SharePoint web applications are accessed through port 80. This means that you can often access a SharePoint site by entering <http://> to open the root site collection.

2. In **Site Collection Administration**, select **Site collection features**.
3. Scroll down the page until you find the **Report Viewer web part** Feature.
4. Select **Activate**.



5. Repeat for additional site collections by opening each site and clicking Site Actions.

Optionally, you can also use PowerShell to enable this feature on all sites using the [Enable-SPFeature](#) cmdlet.

```
Get-SPWebApplication "<web application url>" | Get-SPSite -Limit ALL |  
    ForEach-Object {  
        Write-Host "Enabling feature for $($_.URL)"  
        Enable-SPFeature -identity "ReportViewerWebPart" -URL $_.URL -ErrorAction Continue  
    }
```

Remove the solution

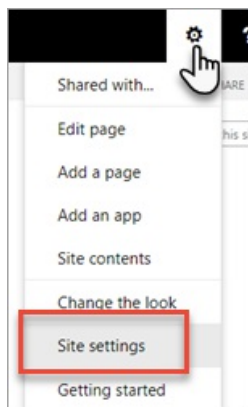
Although SharePoint Central Administration provides solution retraction, you do not need to retract the **ReportViewerWebPart.wsp** file unless you are systematically troubleshooting an installation or patch deployment problem.

1. In SharePoint Central Administration, in **System Settings**, select **Manage farm solutions**.
2. Select **ReportViewerWebPart.wsp**.
3. Select **Retract Solution**.

Remove the web part from Site settings

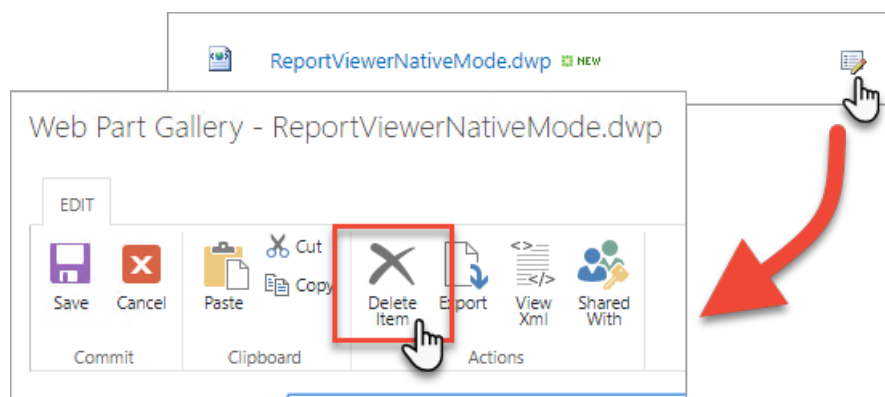
Retracting the solution does not remove the Report Viewer web part from the list of web parts within your SharePoint site. To remove the Report Viewer web part, do the following.

1. In your SharePoint site, select the **gear** icon in the upper left and select **Site Settings**.



By default, SharePoint web applications are accessed through port 80. This means that you can often access a SharePoint site by entering *http://* to open the root site collection.

2. Under **Web Designer Galleries**, select **web parts**.
3. Select the **edit icon** next to **ReportViewerNativeMode.dwp**. It may not be listed on the first page of results.
4. Select **Delete Item**.



Deletion of the web part can be attempted by using PowerShell, but there is not a direct command for it. For a script example, see [How to delete web parts from the web part Gallery](#).

Supported languages

The following languages are supported with the web part:

- English (en)
- German (de)
- Spanish (sp)
- French (fr)
- Italian (it)
- Japanese (ja)
- Korean (ko)
- Portuguese (pt)
- Russian (ru)
- Chinese (Simplified - zh-HANS and zh-CHS)
- Chinese (Traditional - zh-HANT and zh-CHT)

Troubleshoot

- Error when uninstalling SSRS if you have SharePoint integrated mode configured:

Install-SPRSService : [A]

Microsoft.ReportingServices.SharePoint.SharedService.Service.ReportingWebService cannot be cast to [B]Microsoft.ReportingServices.SharePoint.SharedService.Service.ReportingWebService. Type A originates from 'Microsoft.ReportingServices.SharePoint.SharedService, Version=14.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91' in the context 'Default' at location 'C:\Windows\assembly\GAC_MSIL\Microsoft.Reporting Services.SharePoint.SharedService.dll'. Type B originates from 'Microsoft.ReportingServices.SharePoint.SharedService, Version=12.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91' in the context 'Default' at location 'C:\Windows\assembly\GAC_MSIL\Microsoft.Reporting Services.SharePoint.SharedService.dll'.

Solution:

1. Remove the Report Viewer web part
 2. Uninstall SSRS
 3. Reinstall the Report Viewer web part
- Error when trying to upgrade SharePoint if you have SharePoint integrated mode configured:

Could not load file or assembly 'Microsoft.ReportingServices.Alerting.ServiceContract, Version=14.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91' or one of its dependencies. The system cannot find the file specified. 00000000-0000-0000-0000-000000000000

Solution:

1. Remove the Report Viewer web part
2. Uninstall SSRS
3. Reinstall the Report Viewer web part


Next steps

After the Report Viewer web part has been deployed and activated, you can add the web part to a SharePoint page. For more information, see [Add Report Viewer web part to a SharePoint page](#).

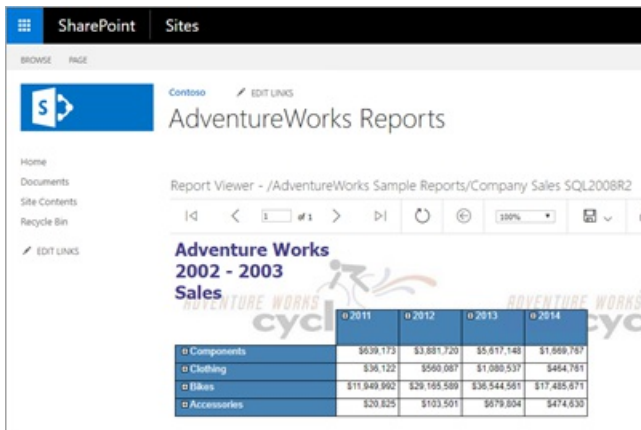
More questions? [Try asking the Reporting Services forum](#)

Add SQL Server Reporting Services Report Viewer web part to a SharePoint page

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016 and later)  SharePoint  Power BI Report Server

Display a report, from SQL Server Reporting Services or Power BI Report Server, by adding a Report Viewer web part to a SharePoint page.

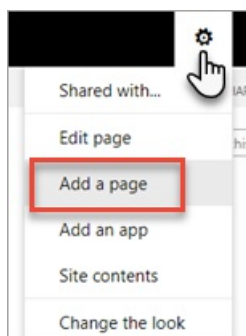


Prerequisites

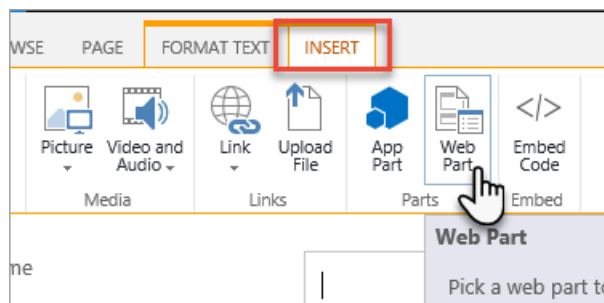
- For reports to load successfully, the Claims to Windows Token Service (C2WTS) needs to be configured for Kerberos constrained delegation. For more information on how to configure C2WTS, see [Claims to Windows Token Service \(C2WTS\) and Reporting Services](#).
- The Report Viewer web part must be deployed to your SharePoint farm. For information on how to deploy the Report Viewer web part solution project, see [Deploy the Report Viewer web part on a SharePoint site](#).
- To add a web part to a web page, you must have the Add and Customize Pages permission at the site level. If you are using default security settings, this permission is granted to members of the **Owners** group who have the Full Control level of permission.

Add web part

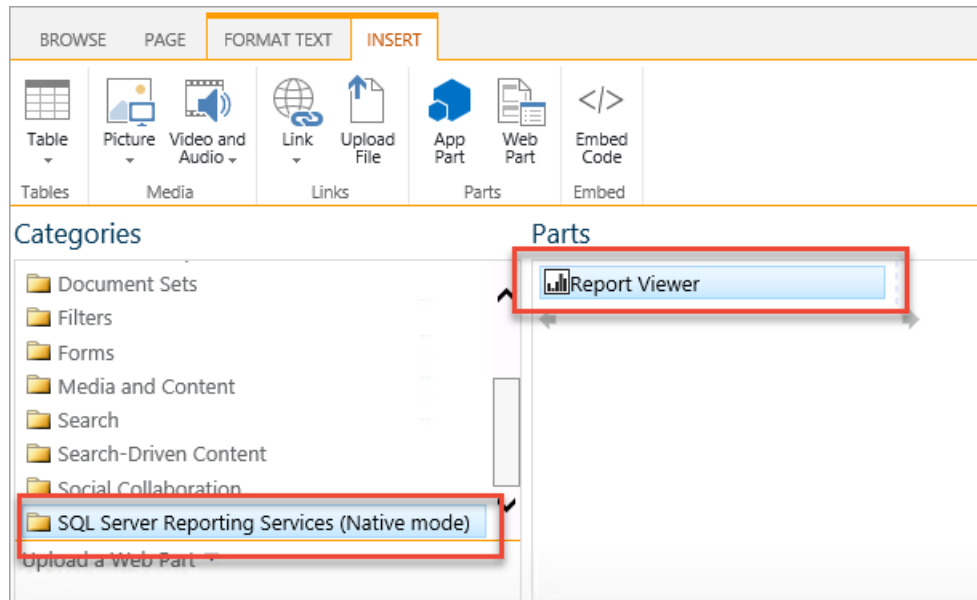
- In your SharePoint site, select the **gear** icon in the upper left and select **Add a page**.



- Give your page a name and select **Create**.
- Within the page designer, select the **Insert** tab in the ribbon. Then select **web part** within the **Parts** section.



- Under **Categories**, select **SQL Server Reporting Services (Native mode)**. Under **Parts**, select **Report Viewer**. Then select **Add**.

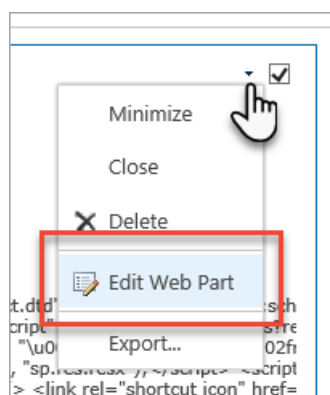


This may initially appear with an error. The error is because the default report server URL is set to <http://localhost> and may not be available at that location.

Configure the Report Viewer web part

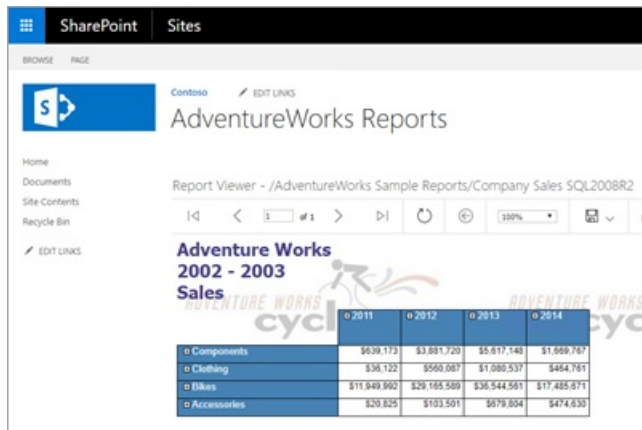
To configure the web part to point to your specific report, do the following.

- When editing the SharePoint page, select the down arrow in the upper right of the web part and select **Edit web part**.



- Enter the **Report Server URL** for the report server hosting your report. This should look similar to <http://myrsserver/reportserver>.
- Enter the path and name of the report you want to display within the web part. This will look similar to `/AdventureWorks Sample Reports/Company Sales`. In this example, the report *Company Sales* is in a folder called *AdventureWorks Sample Reports*.

4. If your report requires parameters, after you have supplied the report server URL and the name of the report, select **Load Parameters** within the **Parameters** section.
5. Select **Ok** to save your changes to the web part configuration.
6. Select **Save**, within the Office ribbon, to save the changes to the SharePoint page.






Considerations and limitations

- The Report Viewer web part cannot be used on modern pages within SharePoint.
- Power BI reports can't be used with the Report Viewer web part.
- If you don't see the Report Viewer web part, to add to your page, make sure you have [deployed the Report Viewer web part](#).

More questions? [Try asking the Reporting Services forum](#)

Connect Filter or Documents web part with a Reporting Services Report Viewer web part

5/3/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

If you are using a SharePoint product, you can create a dashboard or web part Page that includes a Filter web part or Documents web part and a Report Viewer web part. Supported versions are SharePoint Foundation 2010 or SharePoint Server 2010. Also supported are Windows SharePoint Services 3.0 or Office SharePoint Server 2007. By connecting a Filter web part, users who select filter values in a Filter web part can send the value to a parameterized report on the same page. By connecting a Documents web part, users who click on reports in the Documents library can view the report in an adjacent Report Viewer web part.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

The Filter web part is used to send values to one or more parameters on a report. To use a Filter web part, the report must have parameters defined for it that are compatible with the values, data type, and format sent by the web part.

The Documents web part is associated with the Documents library of the Home site. To view, add, or remove items from the Documents library, click **View All Site Content**. In Libraries, click **Documents**. You can use the **New**, **Upload**, and **Actions** menu to manage the items in the Documents library.

Connect a Filter web part

1. Open or create the web part page or dashboard.
2. On the **Site Actions** menu, click **Edit Page**.
3. Click **Add a web part**.
4. In **All web parts**, in the **Miscellaneous** category, select **SQL Server Reporting Services Report Viewer**.
5. Click **Add**. The web part is added at the top of the zone.
6. On another zone in the same web part page or dashboard, click **Add a web part**.
7. In **All web parts**, in the **Filters** section, select a web part.
8. Click **Add**. The web part is added at the top of the zone.
9. In the zone that contains the web part, click the web part **edit** menu, point to **Connections**, point to **Send Filter Values To**, and then select **Report Viewer - report name**.
10. Check in your changes and save the page.

Connect a Documents web part

1. Open or create the web part page or dashboard.
2. On the **Site Actions** menu, click **Edit Page**.
3. Click **Add a web part**.
4. In **All web parts**, in the **Lists and Library** section, select **Documents**.
5. Click **Add**. The web part is added at the top of the zone.
6. Click **Apply** at the bottom of the tool pane, and then click **OK** to close the pane.
7. On another zone in the same web part page or dashboard, click **Add a web part**.
8. In **All web parts**, in the **Miscellaneous** category, select **SQL Server Reporting Services Report Viewer**.
9. Click **Add**. The web part is added at the top of the zone.
10. In the zone that contains the web part, click the web part **edit** menu, point to **Connections**, point to **Get report definitions from**, and then select **Documents**.
11. Check in your changes and save the page.

See also

[Add the Report Viewer web part to a web page](#)



[Report Viewer web part on a SharePoint Site](#)

[Customize the Report Viewer web part](#)

More questions? [Try asking the Reporting Services forum](#)

Customize the Report Viewer web part

5/3/2018 • 9 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

You can use the Report Viewer web part to view reports that run on a report server that is configured for SharePoint integration. Reports that you can display include report definition (.rdl) files and Report Builder reports. Reports open in the Report Viewer web part in a new page automatically, but you can also add a Report Viewer web part to an existing web page or site if you want a particular report to always be visible on that page.

NOTE

Although they have identical names, the Report Viewer web part that is installed through the Reporting Services Add-in is different from the Report Viewer web part that is included in the RSWebParts.cab file. The instructions in this topic are specifically for the Report Viewer web part that is installed through the Reporting Services Add-in.

You can customize the Report Viewer web part in the following ways:

- Change the appearance of the web part by setting properties.
- Choose which interactive reporting features are available on the report toolbar.
- Specify which view areas are available. The Report Viewer web part has a report view area, a Parameters area, and a Credentials area.

You cannot extend the Report Viewer web part to support different file types, and you cannot replace the report toolbar with a custom toolbar or add new functionality to the existing toolbar. If you require customization of the standard features, you should create a custom web part.

Setting web part properties

A web part has custom properties that are used to configure specific functionality. A web part also has common properties that are standard for every web part.

Change default properties

The Report Viewer web part has default properties that are ideally suited for opening reports on demand from a library or folder. By default, all available controls are displayed on the toolbar, and height and width are set to use all of the available space on the web page. If you want to modify the default properties, you can customize the web part through **Site Settings**.

1. On the **Site Actions** menu, click **Site Settings**.
2. Under Galleries, click **web parts**.
3. Click **ReportViewer.dwp**.
4. Open the tool pane and set the properties you want to use.

Customize an embedded Report Viewer in a web page

You can set properties to make the Report Viewer fit within a web page. The report viewer can use the same style

and colors as the page that contains it. You can hide all or part of the toolbar, document map, and parameters area to maximize the report viewing area within the allocated space. The report always uses the styles that are defined for it when it was created; you cannot customize report appearance after it is published to a SharePoint library.


If you are embedding the Report Viewer web part in a web page, you should set the **Report URL** property to a specific report. If you do not, the Report Viewer will display instructions for linking to a report. You cannot customize or remove the instructions.

Custom properties of the Report Viewer web part

When setting custom properties, be aware that some properties are used only when the Report Viewer web part is embedded in a page. Examples include Title, Height, Width, Chrome type, and Zone. Other properties, such as Toolbar settings and Parameters settings, are used regardless of whether the Report Viewer appears within a page or opens a report in full-page mode.

The custom properties of the Report Viewer web part are listed below.

PROPERTY	DESCRIPTION
Report	A fully-qualified path to a report that is on the current SharePoint site, or on a site within the same Web application or farm. For best results when setting additional properties, click Apply after you specify the report URL.
Hyperlink Target	Standard HTML that specifies the target frame for displaying linked content within the current document. For reports that include hyperlinks to external Web sites, you can specify whether a target document replaces the existing report within the current window, or opens in a new browser window. Valid values include _Top , _Blank , and _Self . _Top uses the current window, _Blank loads the document in a new browser window, and _Self opens the document within the current frame. Although _Parent is a valid value for the Target attribute in HTML, do not use it for a Report Viewer web part that is embedded in a page.
Auto-Generate web part Title	A generated title that includes the name of the Report Viewer web part plus the name of the report, separated by a dash. If the report does not have a title, the report file name is used. The title is visible when you add a web part to a page. If this check box is selected, the title will be generated each time the page is refreshed.
Auto-Generate web part Title Detail Link	A generated hyperlink that appears above the web part. You can click the link to open the report in a new page, in full-page mode.
Show report builder menu item	Shows or hides the Actions menu option to open Report Builder.
Show subscription menu item	Shows or hides the Actions menu option to create a subscription for the report.
Show print menu item	Shows or hides the Actions menu option to print the report.
Show export menu item	Shows or hides the Actions menu option to export the report.
Show refresh button	Shows or hides the refresh button on the toolbar.

PROPERTY	DESCRIPTION
Show page navigation controls	Shows or hides the report navigation buttons on the toolbar. This option changes the visibility of all the navigation controls.
Show back button	Shows or hides the back button on the toolbar.
Show find controls	Shows or hides the find controls on the toolbar. The find controls allow a user to search for text in the rendered report. This option changes the visibility of all the find controls.
Show zoom control	Shows or hides the zoom control on the toolbar.
Show ATOM feed button	Shows or hides the ATOM feed button on the toolbar. 
ToolBar location	Determines the location of the toolbar within the report viewer. Valid values include Top and Bottom .
Prompt area	Valid values include Displayed , Collapsed , and Hidden . Displayed displays the Parameters area for reports that include parameterized values and that require user input before the report will run. Use Hidden if all of the report parameters are specified and you do not want the parameters area to be visible to users.
Parameters Area Width	You can choose the measurement and value. The default is 200 pixels. The only requirement for this property is that it is greater than zero.
Document Map	A report navigation control that is defined in the report and used to provide one-click access to specific sections of a report. It is available in HTML reports. The document map is displayed in a collapsible area next to the report viewing area. Valid values include Displayed , Collapsed , and Hidden . If a document map is defined for a report, the area is expanded by default unless marked as hidden or collapsed in the web part properties. If the document map is collapsed, you can click the arrow to expand it.
Document Map Area Width	You can choose the measurement and value. The default is 200 pixels. The only requirement for this property is that it is greater than zero.

PROPERTY	DESCRIPTION
Load Parameters	<p>Retrieve parameter properties for the report. Not all reports have parameters. If the report does not have parameters, no values will be returned. If you are setting properties for a report that you just uploaded, you might get an error indicating that the data source connection has been deleted. If this occurs, reset the connection and then finish setting parameter properties after the connection is specified. For more information about how to set the connection, see Create and Manage Shared Data Sources (Reporting Services in SharePoint Integrated Mode).</p> <p>For best results, click Apply before clicking Load Parameters.</p> <p>After you load parameter properties, you can set them the same way you would in the parameter property pages of the report. For more information about how to set parameters, see Set Parameters on a Published Report (Reporting Services in SharePoint Integrated Mode).</p>

Customizing the toolbar

The toolbar appears beneath the title and extends across the top of the report. The toolbar provides an **Actions** menu, page navigation for paginated reports, refresh, and zoom. It includes a document map control for reports that have a document map. The **Actions** menu includes commands for exporting the report, searching for text or numbers within a report, printing the report, and, opening the report in Report Builder.

You cannot add new commands to the **Actions** menu but you can customize it by changing the options that are visible to users. To change the visibility of toolbar buttons and controls, you change options in the **ToolBar Items Visibility** section of the web part. You can also remove the **Print** command or specific export formats by making these features unavailable on the report server. Page navigation controls are available for reports that have page breaks; otherwise, the report is a single page of variable length. **Refresh** re-processes the report using the parameters that are current for the report. To display all the controls on one line, set the overall width of the web part to at least 400 pixels.

Customizing the viewing area

The view area is used to display reports. The report view area is shared with the Parameters and Credentials area, if they are used. If credentials are required, the Credentials area is displayed next to an empty report view area. The Credentials area closes after the user provides credentials and runs the report. To customize the text that prompts users to set credentials, modify the data source connection properties. For more information, see [Create and Manage Shared Data Sources \(Reporting Services in SharePoint Integrated Mode\)](#).

The Parameters area provides fields for entering values before running the report. It is only used when a report definition includes parameters. When either the Parameters or Credentials areas are displayed, the report view is adjusted to use the remaining width of the web part. You can set properties on the web part to customize the width of Parameters. You can also define the labels that appear next to individual parameters on the page. For more information about how to modify parameter labels, see [Set Parameters on a Published Report \(Reporting Services in SharePoint Integrated Mode\)](#).




See also

[Report Viewer web part on a SharePoint Site](#)
[Add the Report Viewer web part to a web page](#)

More questions? [Try asking the Reporting Services forum](#)

SharePoint site settings for the Report Viewer web part - Reporting Services

6/25/2018 • 2 minutes to read • [Edit Online](#)

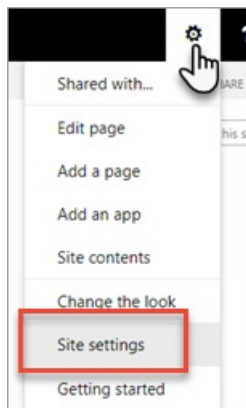
APPLIES TO:  SQL Server Reporting Services (2016 and later)  SharePoint  Power BI Report Server

The Report Viewer web part has a couple of settings that can be configured. These settings can be enabled and disabled on the SharePoint site settings page by a site administrator. Note that each site has its own settings. Additionally, these settings will not be reset after reinstalling the Report Viewer web part.

Accessing the site settings page

The site settings can be accessed by:

1. In your SharePoint site, select the **gear** icon in the upper left and select **Site Settings**.



2. Clicking **Report Viewer Web Part Settings** in the **Reporting Services** site settings group.

NOTE

The site settings can also be reached by navigating directly to

```
<site>/_layouts/15/ReportViewerWebPart/ReportViewerWebPartSettings.aspx
```

Report Viewer web part settings

SETTING	COMMENTS
Collect usage data	Enables error and usage information to be sent to Microsoft to help improve our products. For the Microsoft error reporting data collection policy, see the Microsoft SQL Server Privacy Statement .
Enable Accessibility Metadata for Reports	Sets the AccessibleTablix device info for rendered reports.

High availability in SQL Server Reporting Services

5/3/2018 • 2 minutes to read • [Edit Online](#)

A Reporting Services report server is a stateless server that stores application data, content, properties, and session information in two SQL Server relational databases. As such, the best way to ensure the availability of Reporting Services functionality is to do the following:

- Use the high availability features of the SQL Server Database Engine to maximize the uptime of the report server databases. If you configure a Database Engine instance to run in a failover cluster, you can select that instance when you create a report server database.
- Use SQL Server Always On availability groups with the Reporting Services databases and for data sources, as possible. For more information, see [Reporting Services with Always On Availability Groups](#).
- Configure multiple report servers to run in a scale-out deployment, where all the servers share a single report server database. Deploying multiple report server instances, preferably on different servers, in a scale-out deployment can help provide uninterrupted service in the event one of the report server instances goes down.

A scale-out deployment provides a way to share a database. If one report server goes down, other servers in the same deployment will continue to work.




Reporting Services is not cluster-aware. By itself, a scale-out deployment does not provide load balancing; it does not detect the processing loads on a report server and route new processing requests to the least busy server. It does not re-route processing requests that failed before completion. To get load balancing features, you must configure load balancing for the Web servers that host the report servers, and then configure the report servers in a scale-out deployment so that they share the same report server database.

The Report Server Web service and Windows service are tightly integrated and run together as a single report server instance. You cannot configure availability for one service separately from the other.

More questions? [Try asking the Reporting Services forum](#)

Enable and disable RDL sandboxing for Reporting Services in SharePoint integrated mode

5/3/2018 • 7 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Reporting Services (2016)  SharePoint  Power BI Report Server

For content related to previous versions of SQL Server Reporting Services, see [SQL Server Reporting Services 2014](#).

The RDL (Report Definition Language) sandboxing feature lets you detect and restrict the usage of specific types of resources, by individual tenants, in an environment of multiple tenants that use a single web farm of report servers. An example of this is a hosting services scenario where you might maintain a single web farm of report servers that are used by multiple tenants, and perhaps different companies. As a report server administrator, you can enable this feature to help achieve the following objectives:

- Restrict external resource sizes. External resources include images, .xslt files, and map data.
- At report publish time, limit types and members that are used in expression text.
- At report processing time, limit the length of the text and the size of the return value for expressions.

NOTE

Reporting Services integration with SharePoint is no longer available after SQL Server 2016.

When RDL Sandboxing is enabled, the following features are disabled:

- Custom code in the **<Code>** element of a report definition.
- RDL backward compatibility mode for SQL Server 2005 Reporting Services (SSRS) custom report items.
- Named parameters in expressions.

This topic describes each element in the **<RDLSandboxing>** element in the RSReportServer.Config file. For more information about how to modify this file, see [Modify a Reporting Services Configuration File \(RSreportserver.config\)](#). A server trace log records activity related to the RDL Sandboxing feature. For more information about trace logs, see [Report Server Service Trace Log](#).

Example configuration

The following example shows the settings and example values for the **<RDLSandboxing>** element in the RSReportServer.Config file.

```

<RDL sandboxing>
  <MaxExpressionLength>5000</MaxExpressionLength>
  <MaxResourceSize>5000</MaxResourceSize>
  <MaxStringResultLength>3000</MaxStringResultLength>
  <MaxArrayResultLength>250</MaxArrayResultLength>
  <Types>
    <Allow Namespace="System.Drawing" AllowNew="True">Bitmap</Allow>
    <Allow Namespace="TypeConverters.Custom" AllowNew="True">*</Allow>
  </Types>
  <Members>
    <Deny>Format</Deny>
    <Deny>StrDup</Deny>
  </Members>
</RDL sandboxing>

```

Configuration settings

The following table provides information about configuration settings. Settings are presented in the order in which they appear in the configuration file.

SETTING	DESCRIPTION
MaxExpressionLength	Maximum number of characters allowed in RDL expressions. Default: 1000
MaxResourceSize	Maximum number of KB allowed for an external resource. Default: 100
MaxStringResultLength	Maximum number of characters allowed in a return value for an RDL expression. Default: 1000
MaxArrayResultLength	Maximum number of items allowed in an array return value for an RDL expression. Default: 100
Types	The list of members to allow within RDL expressions.
Allow	A type or set of types to allow in RDL expressions.
Namespace	Attribute for Allow that is the namespace that contains one or more types that apply to Value. This property is case-insensitive.
AllowNew	Boolean attribute for Allow that controls whether new instances of the type are allowed to be created in RDL expressions or in an RDL <Class> element. When RDL sandboxing is enabled, new arrays cannot be created in RDL expressions, regardless of the setting of AllowNew .

SETTING	DESCRIPTION
Value	Value for Allow that is the name of the type to allow in RDL expressions. The value * indicates that all types in the namespace are allowed. This property is case-insensitive.
Members	For the list of types that are include in the <Types> element, the list of member names that are not allowed in RDL expressions.
Deny	<p>The name of a member that is not allowed in RDL expressions. This property is case-insensitive.</p> <p>When Deny is specified for a member, all members with this name for all types are not allowed.</p>

Working with expressions when RDL sandboxing is enabled

You can modify the RDL Sandboxing feature to help manage the resources that are used by an expression in the following ways:

- Restrict the number of characters that are used for an expression.
- Restrict the size of the result returned by an expression.
- Allow a specific list of types that can be used in an expression.
- Restrict the list of members by name for the list of allowed types that can be used in an expression.
- The RDL Sandboxing feature enables you to create a list of approved types and a list of denied members. The list of approved types is called an allow list. The list of denied members is called a block list.

NOTE

In the report definition, a computer cannot know the type of each instances of an expression reference. When you add a member to the block list, you are denying all members of that name across all types in the allow list.

RDL expression results are verified at run time. RDL expressions are verified in the report definition when the report is published. Monitor the report server trace log for violations. For more information, see [Report Server Service Trace Log](#).

Working with types

When you add a type to the allow list, you are controlling the following entry points to access RDL expressions:

- Static members of a type.
- The Visual Basic **New** method.
- The **<Classes>** element in the report definition.
- Members that you have added to the block list for a type in the allow list.

The allow list does not control the following entry points:

- Report datasets. Fields in report datasets that are returned from queries might contain any valid RDL type.
- Report parameters. User-supplied parameter values might contain any valid RDL type.
- Members of an enabled type that are not in the block list. By default, all members of all types in the allow list

are enabled. When you add a member name to the block list, you are denying all members with that name across all types that are in the allow list.

To enable a member of one type but deny a member with the same name for a different type, you must do the following:

- Add a **<Deny>** element for the member name.
- Create a proxy member with a different name on a class in a custom assembly for the member that you want to enable.
- Add that new class to the allow list.

To add Visual Basic .NET Framework functions to the allow list, add the corresponding types from the `Microsoft.VisualBasic` namespace to the allow list.

To add Visual Basic .NET Framework type keywords to the allow list, add the corresponding CLR type to the allow list. For example, to use the Visual Basic .NET Framework keyword **Integer**, add the following XML fragment to the **<RDLSandboxing>** element:

```
<Allow Namespace="System">Int32</Allow>
```

To add a generic or a Visual Basic .NET Framework nullable type to the allow list, you must do the following:

- Create a proxy type for the generic or Visual Basic .NET Framework nullable type.
- Add the proxy type to the allow list.

Adding a type from a custom assembly to the allow list does not implicitly grant execute permission on the assembly. You must specifically modify the code access security file and provide execute permission to your assembly. For more information, see [Code Access Security in Reporting Services](#).

Maintaining the **<Deny>** list of members

When you add a new type to the allow list, use the following list to determine when you might have to update the block list of members:

- When you update a custom assembly with a version that introduces new types.
- When you add members to the types in the allow list.
- When you update the .NET Framework on the report server.
- When you upgrade the report server to a later version of Reporting Services.
- When you update a report server to handle a later RDL schema, because new members might have been added to RDL types.

Working with operators and new

By default, Visual Basic .NET Framework language operators, except for **New**, are always allowed. The **New** operator is controlled by the **AllowNew** attribute on the **<Allow>** element. Other language operators, such as the default collection accessor operator **!** and Visual Basic .NET Framework cast macros such as **CInt**, are always allowed.

Adding operators to a block list, including custom operators, is not supported. To exclude operators for a type, you must do the following:

- Create a proxy type that does not implement the operators that you want to exclude.
- Add the proxy type to the allow list.

To create a new array in an RDL expression, create the array in a method on a class that you define, and add that class to the allow list.

To create a new array in an RDL expression, you must do the following:

- Define a new class and create the array in a method on that class.
- Add the class to the allow list.

See also

[RsReportServer.config configuration file](#)

[Report Server Service trace log](#)

More questions? [Try asking the Reporting Services forum](#)

Configure Reporting Services to use a subject alternative name

5/3/2018 • 2 minutes to read • [Edit Online](#)

This topic explains how to configure Reporting Services (SSRS) to use a subject alternative name (SAN) by modifying the rsreportserver.config file and using the Netsh.exe tool.

The instructions apply to the Reporting Service URL as well as a Web Service URL.

To use a SAN, the SSL certificate must be registered on the server, signed, and have the private key. You cannot use a self-signed certificate

URLs in Reporting Services can be configured to use an SSL certificate. A certificate normally has just a subject name, which allows only one URL for an SSL (Secure Sockets Layer) session. The SAN is an additional field in the certificate that allows an SSL service to listen for many URLs, and to share the SSL port with other applications. The SAN looks something like `www.s2.com`.

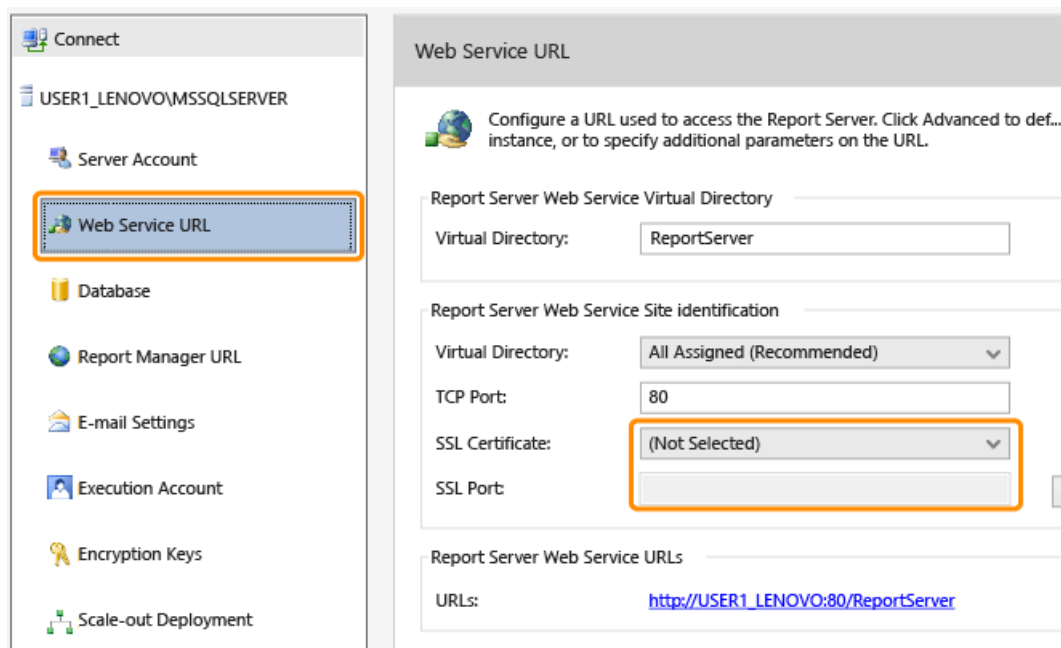
For more information about SSL settings for Reporting Services, see [Configure SSL Connections on a Native Mode Report Server](#).

Configure SSRS to use a subject alternative name for web service URL

1. Start Reporting Services Configuration Manager.

For more information, see [Reporting Services Configuration Manager \(Native Mode\)](#).

2. On the **Web Service URL** page, select an SSL port and SSL Certificate.



The configuration manager registers the SSL certificate for the port.

3. Open the rsreportserver.config file.

For SSRS Native mode, the file is located by default in the following folder:

```
\Program Files\Microsoft SQL Server\MSRS11.MSSQLSERVER\Reporting Services\ReportServer
```

4. Copy the URL section for the Report Server Web Service application.

For example, the following original URL section is:

```
<URL>
<UrlString>https://localhost:443</UrlString>
<AccountSid>S-1-5-20</AccountSid>
<AccountName>NT Authority\NetworkService</AccountName>
</URL>
```

The following modified URL section is:

```
<URL>
<UrlString>https://www.s1.com:443</UrlString>
<AccountSid>S-1-5-20</AccountSid>
<AccountName>NT Authority\NetworkService</AccountName>
</URL>
<URL>
<UrlString>https://www.s2.com:443</UrlString>
<AccountSid>S-1-5-20</AccountSid>
<AccountName>NT Authority\NetworkService</AccountName>
</URL>
```

5. Save the rsreportserver.config file.
6. Start a command prompt as an administrator, and run the Netsh.exe tool.

```
C:\windows\system32\netsh
```

7. Switch to the http context by typing the following.

```
Netsh>http
```

8. Show the existing urlacl's by typing the following:

```
Netsh http>show urlacl
```

An entry such as the following appears.

```
Reserved URL           : https:// www.s1.com:443/ReportServer/
User: NT SERVICE\ReportServer
Listen: Yes
Delegate: No
SDDL: D:(A;;GX;;;S-1-5-80-1234567890-123456789-123456789-1234567890)
```

An urlacl is a DACL (Discretionary Access Control List) for a reserved URL.

9. Create a new entry for the subject alternative name, with the same user and SDDL as the existing entry, by typing the following.

```
netsh http>add urlacl url=https://www.s2.com:443/ReportServer
user="NT Service\ReportServer" sddl=D:(A;;GX;;;S-1-5-80-1234567980-12346579-123456789-1234567890)
```

10. On the **Report Server Status** page of the Reporting Services Configuration Manager, Click **Stop** and then click **Start** to restart the report server.

See also

[RsReportServer.config configuration file](#)

[Reporting Services Configuration Manager](#)

[Modify a Reporting Services configuration file](#)

[Configure Report Server URLs](#)

More questions? [Try asking the Reporting Services forum](#)