**Types of Reports**

In Reporting Services, you can use reports in a variety of ways. This topic describes the terminology used to describe the various types of reports and the ways reports get created and used. A single report can have characteristics from more than one type; for example, snapshot reports can be parameterized, ad hoc reports incorporate clickthrough report functionality due to the report models upon which they are based, and subreports can be linked reports.

With Reporting Services, you can create the following types of reports:

* [Parameterized reports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Param)
* [Linked reports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Link)
* [Snapshot reports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Snapshot)
* [Cached reports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Cache)
* [Ad hoc reports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#AdHoc)
* [Clickthrough reports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Click)
* [Drilldown reports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Drilldown)
* [Drillthrough reports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Drill)
* [Subreports](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Sub)

[http://i.msdn.microsoft.com/Global/Images/clear.gif](http://msdn.microsoft.com/en-us/library/bb630404.aspx#Sub) Report Appearance, Processing, and Delivery Terminology

There are several ways to think about report type. You might think about it as the way data appears in the report. In Reporting Services, the appearance of data in a report depends on the type of data region you use; for example, tabular reports and chart reports use different data regions. For more information about how to display data, see [Data Regions and Maps (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd207125.aspx). Likewise, the functionality that is available in a report depends on the output format; for example, interactive features like drillthrough reports are available in Web-based export formats but not in all Image-based export formats. A report's final output format affects which features you can include in a report. For more information about design considerations for various export formats, see [Exporting Reports (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd239307.aspx).

There is also terminology associated with the stage of processing a report is in. For more information about the differences between reportdefinitions, publishedreports, and renderedreports, see [Reports, Report Parts, and Report Definitions (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd207054.aspx). Finally, for information about report scheduling and on-demand reports, see [Scheduling Reports, Shared Datasets, and Subscriptions](http://msdn.microsoft.com/en-us/library/ms159767.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Parameterized Reports

A parameterized report uses input values to complete report or data processing. With a parameterized report, you can vary the output of a report based on values that are set when the report runs. Parameterized reports are frequently used for drillthrough reports, linked reports, and subreports, connecting and filtering reports with related data.

**Using Parameters**

Parameters are used in dataset queries to select report data, to filter the result set that the query returns, or to set layout properties used to display or hide parts of a report. You can also specify cascading parameters that populate a series of dependent, drop-down parameter lists. For example, a drop-down list of Region parameter values can be used to populate a drop-down list of City parameter values.

You can use parameters with linked reports by pairing a specific parameter with each linked report to change the outcome. For example, you can create a single regional sales report that shows the sales for all regions, and then use a parameter for each linked report to filter data for a particular region. Specific parameter values can be stored with the report so that users do not have to type values.

Not all parameters may be visible in the report at run time. A report author, report server administrator, or content manager can specify which values to use and then hide the input fields on the report.

**Query Parameters and Report Parameters**

Reporting Services supports two kinds of parameters: query parameters and report parameters. *Query parameters* are used during data processing to select or filter data. Query parameters are specified in the syntax of a data processing extension. If a query parameter is specified, a value must be provided either by the user or by default properties to complete the SELECT statement or stored procedure that retrieves data for a report. *Report parameters* are used during report processing to show a different aspect of the data. A report parameter is usually used to filter a large set of records, but it can have other uses depending on the queries and expressions used in the report. Report parameters differ from query parameters in that they are defined in a report and processed by the report server, while query parameters are defined as part of the dataset query and processed on the database server. For more information, see [Parameters (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd220464.aspx) and [Setting Parameter Properties for a Published Report](http://msdn.microsoft.com/en-us/library/ms155908.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Linked Reports

A linked report is a report server item that provides an access point to an existing report. Conceptually, it is similar to a program shortcut that you use to run a program or open a file.

A linked report is derived from an existing report and retains the original's report definition. A linked report always inherits report layout and data source properties of the original report. All other properties and settings can be different from those of the original report, including security, parameters, location, subscriptions, and schedules.

You can create a linked report on the report server when you want to create additional versions of an existing report. For example, you could use a single regional sales report to create region-specific reports for all of your sales territories.

Although linked reports are typically based on parameterized reports, a parameterized report is not required. You can create linked reports whenever you want to deploy an existing report with different settings. For more information, see [Adding, Modifying, and Deleting Linked Reports](http://msdn.microsoft.com/en-us/library/ms155889.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Snapshot Reports

A report snapshot is a report that contains layout information and query results that were retrieved at a specific point in time. Unlike on-demand reports, which get up-to-date query results when you select the report, report snapshots are processed on a schedule and then saved to a report server. When you select a report snapshot for viewing, the report server retrieves the stored report from the report server database and shows the data and layout that were current for the report at the time the snapshot was created.

Report snapshots are not saved in a particular rendering format. Instead, report snapshots are rendered in a final viewing format (such as HTML) only when a user or an application requests it. Deferred rendering makes a snapshot portable. The report can be rendered in the correct format for the requesting device or Web browser.

Report snapshots serve three purposes:

* Report history. By creating a series of report snapshots, you can build a history of a report that shows how data changes over time.
* Consistency. Use report snapshots when you want to provide consistent results for multiple users who must work with identical sets of data. With volatile data, an on-demand report can produce different results from one minute to the next. A report snapshot, by contrast, allows you to make valid comparisons against other reports or analytical tools that contain data from the same point in time.
* Performance. By scheduling large reports to run during off-peak hours, you can reduce processing impact on the report server during core business hours.

For more information about creating report snapshots, see [Setting Report Processing Properties](http://msdn.microsoft.com/en-us/library/ms159241.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Cached Reports

A cached report is a saved copy of a processed report. Cached reports are used to improve performance by reducing the number of processing requests to the report processor and by reducing the time required to retrieve large reports. They have a mandatory expiration period, usually in minutes. For more information about how to use cached reports, see [Caching Reports (SSRS)](http://msdn.microsoft.com/en-us/library/ms155927.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Clickthrough Reports

A clickthrough report is a report that displays related data from a report model when you click the interactive data contained within your model-based report. These reports are generated by the report server based on the information contained within the report model. The person who created the model determines which fields are interactive and which fields are returned when a clickthrough report is opened. These field settings cannot be changed in the report authoring tools.

Clickthrough reports are autogenerated. However, you can create an alternative customized report to the model for interactive data items that is displayed instead. The custom report is a standard Reporting Services report. For more information, see [Working with Clickthrough Reports](http://msdn.microsoft.com/en-us/library/ms345252.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Drilldown Reports

Drilldown reports initially hide complexity and enable the user to toggle conditionally hidden report items to control how much detail data they want to see. Drilldown reports must retrieve all possible data that can be shown in the report. For more information, see [Hiding and Showing Report Items by Adding Drilldown (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd207042.aspx).

For reports with large amounts of data, consider drillthrough reports instead.

http://i.msdn.microsoft.com/Global/Images/clear.gif Drillthrough Reports

Drillthrough reports are standard reports that are accessed through a hyperlink on a text box in the original report. Drillthrough reports work with a main report and are the target of a drillthrough action for a report item such as placeholder text or a chart. The main report displays summary information, for example in a matrix or chart. Actions defined in the matrix or chart provide drillthrough links to reports that display greater details based on the aggregate in the main report. Drillthrough reports can be filtered by parameters, but they do not have to be. Drillthrough reports differ from subreports in that the report does not display within the original report, but opens separately. They differ from clickthrough reports in that they are not autogenerated from the data source, but are instead custom reports that are saved on the report server. They differ from drilldown reports in that they retrieve the report data only for the specified parameters or for the dataset query. For more information, see [Drillthrough Reports (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/ff519554.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Subreports

A subreport is a report that displays another report inside the body of a main report. Conceptually, a subreport is similar to a frame in a Web page. It is used to embed a report within a report. Any report can be used as a subreport. The subreport can use different data sources than the main report. The report that the subreport displays is stored on a report server, usually in the same folder as the parent report. You can set up the parent report to pass parameters to the subreport.

Although a subreport can be repeated within data regions using a parameter to filter data in each instance of the subreport, subreports are typically used with a main report as a briefing book or as a container for a collection of related reports. For more information, see [Subreports (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd239314.aspx).

For reports with many instances of subreports, consider using drillthrough reports instead.

**Reporting Services Extensions**

An extension is a .NET Assembly that is invoked by the Report Processor to perform certain processing functions. There are several types of extensions: Data Processing, Delivery, Rendering, Security (authentication and authorization), SemanticQuery, ModelGeneration, and EventProcessing.

For an extension to be used by a report server, it has to be installed (assuming default SSRS configuration) to the C:\Program Files\Microsoft SQL Server\MSSQL.3\Reporting Services\ReportServer\bin directory and configured in C:\Program Files\Microsoft SQL Server\MSSQL.3\Reporting Services\ReportServer\rsreportserver.config.

The last part of an extension filename usually implies the extension's functionality. For example, the HTML rendering extension's filename is Microsoft.ReportingServices.HtmlRendering.dll.

Custom extensions allow developers to add complementing functionality that is not available in SSRS "out-of-the-box." For example, a company can implement an extension that delivers reports to a phone or a fax. You can learn more about extensions in Chapter 26, "Writing Custom Reporting Services Extensions."

**Note**

**This release of SSRS does not allow custom SemanticQuery, ModelGeneration, or EventProcessing extensions.**

#### Data-Processing Extensions

Data-processing extensions retrieve data from the report data source. Some of the tasks performed by data-processing extensions include open connection to a data source, analyze query and return field names, pass parameters, and retrieve and iterate data set. Table 3.1 outlines the data-processing extensions included and configured with SSRS.

#### Table 3.1. Data-Processing Extensions Configured with SSRS

| Extension | Description/Notes |
| --- | --- |
| SQL Server | Connects to and retrieves data from the SQL Server database engine versions 7.0 through 2005. |
| OLE DB | Connects to and retrieves data from OLE DB-compliant data sources. |
| Microsoft SQL Server Analysis | Connects to and retrieves data from the SQL Server Analysis Services Services 2000 and 2005. For Analysis Services 2005, this extension supports both Multidimensional Expressions (MDX) and Data Mining Expressions (DMX). For Analysis Services 2000, this extension supports nonparameterized MDX only. |
| Oracle | Connects to and retrieves data from an Oracle database; requires Oracle client 8i Release 3 (8.1.7) to be installed on a computer on which Reporting Server is installed. |
| ODBC | Connects to and retrieves data from ODBC-compliant data sources. |
| XML | Retrieves XML data from any XML web source (such as a web server) that can be accessed through a URL. |

All extensions, which are installed with SSRS (except XML), leverage corresponding .NET data providers. Microsoft.ReportingServices.DataExtensions library provides wrapper classes that supply SSRS data-processing extension interfaces to .NET data providers.

Developers can create additional custom data-processing extensions.

#### Delivery Extensions

Delivery extensions deliver reports to specific devices or formats. Extensions included with RS **include email and file share delivery**. The delivery method and, therefore, corresponding extension are selected when a user (or an administrator) creates a subscription.

A sample of printer delivery extension is included with SQL Server samples and discussed in Chapter 26. Table 3.2 outlines the delivery extensions included and configured with SSRS.

#### Table 3.2. Delivery Extensions Included with SSRS

| Extension | Purpose |
| --- | --- |
| Email delivery | Delivers a rendered report to an email inbox. Allows setting delivery options that control an output format and whether the report is delivered as a link or as an attachment. |
| File share delivery | Delivers a rendered report to a shared folder. Allows setting delivery options that control a destination folder path, an output format, and whether the report overrides an older version or is added as a new version. |

Developers can create additional custom delivery extensions.

#### Rendering Extensions

Report Server Rendering extensions transform a report's layout and data into a device-specific format. Extensions included with RS include HTML (3.2 and 4.0), Microsoft Excel, Text/CSV, XML, Image (BMP, EMF, GIF, JPEG, PNG, TIFF, WMF), and PDF rendering.

**Note**

**Unlike SSRS 2000, which rendered Excel files as an MHTML file carrying special Excel metatags, SSRS 2005 renders reports to Excel's native binary format.**

Because the final rendering phase is only loosely coupled with data processing, it enables users to choose different rendering options for the same report without the need to requery data sources.

Developers can create additional custom rendering extensions.

#### Security Extensions

This book frequently uses the term "security extension" as if it refers to a single unit. In actuality, there are two interrelated extensions:

* Authentication extension, which handles a process that establishes user's identity
* Authorization extension, which handles a process that checks if an identity has access to a particular SSRS resource

SSRS includes a security extension based on Windows authentication. After a user's identity is established, an authorization process determines whether a Windows user (or a Windows group that contains a user) is configured to access a particular resource on a reporting server.

Developers can create additional custom security extensions. An instance of SSRS can use only one security extension. In other words, either the Windows or a custom extension can be used, but not both at the same time.

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

* [Exporting to a CSV File (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd255223.aspx)
* [Exporting to Microsoft Excel (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd255234.aspx)
* [Exporting to Microsoft Word (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd283105.aspx)
* [Rendering to HTML (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd255269.aspx)
* [Exporting to a PDF File (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd255291.aspx)
* [Exporting to an Image File (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd283101.aspx)
* [Exporting to XML (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/dd283112.aspx)
* [Generating Data Feeds from Reports (Report Builder 3.0 and SSRS)](http://msdn.microsoft.com/en-us/library/ee240754.aspx)

[SQL Server Reporting Services](http://en.wikipedia.org/wiki/SQL_Server_Reporting_Services) will gain charting capabilities from the integration of the data visualization products from [Dundas Data Visualization, Inc.](http://en.wikipedia.org/wiki/Dundas_Data_Visualization,_Inc.), which was acquired by Microsoft

**Model Designer Object Properties**

A report model consists of three parts:

* Semantic Model: A semantic model contains your business model of the data, which is described in terms of familiar names such as Products or Customers.
* Physical Model: A physical model contains a physical description of the database with optional transformations, such as defined queries within the Data Source View, to more closely correspond to the business model that you want.
* Mapping: A mapping describes how the semantic model is represented within a physical model and binds semantic objects to their corresponding physical objects.

Each report model created by Report Model Designer contains exactly one semantic model, one physical model, and one mapping.

A report model is written in Semantic Model Definition Language (SMDL), which is XML-based. A SMDL file is used to create models when calling the report server Web service. The namespace URI for SMDL is http://schemas.microsoft.com/sqlserver/*YYYY*/*MM*/semanticmodeling, where YYYY/MM indicates the date of the release of that version of SMDL. The standard file extension for SMDL files is .smdl. The MimeType to use for SMDL files is simply text/xml. All identifiers in SMDL are case-sensitive and match using Invariant culture. In this documentation, SMDL elements are discussed as objects and collections.

This section discusses the different properties associated with a model and that can be assigned within Model Designer.

#### Types of data sources

| **Source of**  **Report data** | **Reporting Services Data Source Type** | **Name of Data Processing Extension/Data Provider** | **Underlying Data Provider version**  **(Optional)** | **Data**  **Source**  **Platform x86** | **Data**  **Source**  **Platform x64** | **Data**  **Source**  **Platform ia64** | **Version of data source** | **RS**  **Platform x86** | **RS**  **Platform x64** | **RS**  **Platform ia64** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SQL Server relational database | [Microsoft SQL Server](http://msdn.microsoft.com/en-us/library/ms159219.aspx#MicrosoftSQLServer) | Built-in Reporting Services data processing extension | Extends System.Data.SqlClient | Y | Y | Y | SQL Server 2008 and later  SQL Server 2005  SQL Server 2000  SQL Server 7.0, x86 only | Y | Y | Y |
| SQL Server relational database | [OLEDB](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OLEDBSQL) | Built-in Reporting Services data processing extension | Extends System.Data.OledbClient | Y | Y | Y | SQL Server 2008 and later  SQL Server 2005  SQL Server 2000  SQL Server 7.0, x86 only  SQL Server 6.5 and earlier, x86 only | Y | Y | Y |
| SQL Server relational database | [ODBC](http://msdn.microsoft.com/en-us/library/ms159219.aspx#ODBC) | Built-in Reporting Services data processing extension | Extends System.Data.OdbcClient | Y | Y | Y | SQL Server 2008 and later  SQL Server 2005  SQL Server 2000  SQL Server 7.0 | Y | Y | Y |
| SQL Azure database | [Microsoft SQL Azure](http://msdn.microsoft.com/en-us/library/ms159219.aspx#Azure) | Built-in Reporting Services data processing extension | Extends System.Data.SqlClient | N/A | N/A | N/A | SQL Azure | Y | Y | Y |
| SQL Server PDW appliance | [Microsoft Parallel Data Warehouse](http://msdn.microsoft.com/en-us/library/ms159219.aspx#PWD) | Built-in Reporting Services data processing extension | N/A | N/A | N/A | N/A | SQL Server 2008 R2 Parallel Data Warehouse | Y | Y | Y |
| Analysis Services multidimensional database | [Microsoft SQL Server Analysis Services](http://msdn.microsoft.com/en-us/library/ms159219.aspx#AnalysisServices) | Built-in Reporting Services data processing extension | Uses ADOMD.NET | Y | Y | Y | SQL Server 2008 Analysis Services and later  SQL Server 2005 Analysis Services | Y | Y | Y |
| Analysis Services multidimensional database | [Microsoft SQL Server Analysis Services](http://msdn.microsoft.com/en-us/library/ms159219.aspx#AnalysisServices) | Built-in Reporting Services data processing extension | Uses ADOMD.NET | Y | N | N | SQL Server 2000  Analysis Services | Y | N | Y |
| Analysis Services multidimensional database | [OLEDB](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OLEDBAS9) | Built-in Reporting Services data processing extension | Extends System.Data.OledbClient  Version 10.0 | Y | Y | Y | SQL Server 2008 Analysis Services | Y | Y | Y |
| Analysis Services multidimensional database | [OLEDB](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OLEDBAS9) | Built-in Reporting Services data processing extension | Extends System.Data.OledbClient  Version 9.0 | Y | Y | Y | SQL Server 2005 Analysis Services | Y | Y | Y |
| Analysis Services multidimensional database | [OLEDB](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OLEDBAS8) | Built-in Reporting Services data processing extension | Extends System.Data.OledbClient  Version 8.0 | Y | N | Y | SQL Server 2000  Analysis Services | Y | N | Y |
| SharePoint lists | [Microsoft SharePoint List](http://msdn.microsoft.com/en-us/library/ms159219.aspx#SharePointList) | Built-in Reporting Services data processing extension | Gets data from Lists.asmx or the SharePoint object model API interfaces.  See [Note](http://msdn.microsoft.com/en-us/library/ms159219.aspx#SharePointList). | N | Y | N | SharePoint 2010 Products | Y | Y | Y |
| SharePoint lists | [Microsoft SharePoint List](http://msdn.microsoft.com/en-us/library/ms159219.aspx#SharePointList) | Built-in Reporting Services data processing extension | Gets data from Lists.asmx or the SharePoint object model API interfaces.  See [Note](http://msdn.microsoft.com/en-us/library/ms159219.aspx#SharePointList). | Y | Y | N | Windows SharePoint Services 3.0 and Office SharePoint Server 2007 | Y | Y | Y |
| XML | [XML](http://msdn.microsoft.com/en-us/library/ms159219.aspx#XML) | Built-in Reporting Services data processing extension | XML data sources do not have platform dependencies. |  |  |  | XML Web Services or documents | Y | Y | Y |
| Report Server Model | [Report Model](http://msdn.microsoft.com/en-us/library/ms159219.aspx#Model) | Built-in Reporting Services data processing extension for a published SMDL file | Data sources for a model use Built-in data processing extensions.  Oracle-based models requires Oracle client components.  Teradata-based models require .NET Data Provider for Teradata from Teradata.  See Teradata documentation for platform support. |  |  |  | Models can be created from:  SQL Server 2008 and later  SQL Server 2005  SQL Server 2000  SQL Server 2005 Analysis Services  Oracle 9.2.0.3 or later  Teradata v13, v12, and v6.2 | Y | Y | Y |
| SAP multidimensional database | [Sap BI NetWeaver](http://msdn.microsoft.com/en-us/library/ms159219.aspx#SapBINetWeaver) | Built-in Reporting Services data processing extension | See SAP documentation for platform support. |  |  |  | SAP BI NetWeaver 3.5 | Y |  |  |
| Hyperion Essbase | [Hyperion Essbase](http://msdn.microsoft.com/en-us/library/ms159219.aspx#Hyperion) | Built-in Reporting Services data processing extension | See Hyperion documentation for platform support. | Y |  |  | Hyperion Essbase 9.3.1 | Y |  |  |
| Oracle relational database | [Oracle](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OracleClient) | Built-in Reporting Services data processing extension | Extends System.Data.OracleClient  Requires Oracle client components. | Y |  |  | Oracle 10g, 9, 8.1.7 | Y | Y | Y |
| Oracle relational database | Part of [OLEDB](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OLEDBStandard) | Built-in  [OLE DB for Oracle](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OracleOLDEB) | MSDAORA  Part of MDAC from downloads.microsoft.com  Requires Oracle client components.  See Oracle documentation for platform support. | Y |  |  | Depends on version of MDAC.  For more information, search http://support.microsoft.com for "Oracle". | Y |  |  |
| Teradata relational database | [Teradata](http://msdn.microsoft.com/en-us/library/ms159219.aspx#Teradata) | Built-in Reporting Services data processing extension | Extends .NET Data Provider for Teradata from Teradata.  Requires .NET Data Provider for Teradata from Teradata.  See Teradata documentation for platform support. | Y |  |  | Teradata v13  Teradata v12  Teradata v6.20 | Y | N | Y |
| DB2 relational database | Customized registered data extension name | Microsoft OLE DB Provider for DB2 | Install from [Microsoft SQL Server 2008 Feature Pack download site](http://go.microsoft.com/fwlink/?LinkId=158389).  See HI Server documentation for platform support. | Y |  |  |  | Y | N | Y |
| DB2 relational database | Customized registered data extension name |  | 2004 Host Integration Server  See HI Server documentation. | Y |  |  |  | Y | N | Y |
| Generic OLE DB data source | [OLEDB](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OLEDBStandard) | Built-in Reporting Services data processing extension | Any data source that supports OLE DB.  See the data source documentation for platform support. | Y |  |  | Any data source that supports OLE DB. See [Note](http://msdn.microsoft.com/en-us/library/ms159219.aspx#OLEDBStandard). | Y |  |  |
| Generic ODBC data source | [ODBC](http://msdn.microsoft.com/en-us/library/ms159219.aspx#ODBCGeneric) | Built-in Reporting Services data processing extension | Any data source that supports ODBC.  See the data source documentation for platform support. | Y |  |  | Any data source that supports ODBC. See [Note](http://msdn.microsoft.com/en-us/library/ms159219.aspx#ODBCGeneric). | Y | Y | Y |

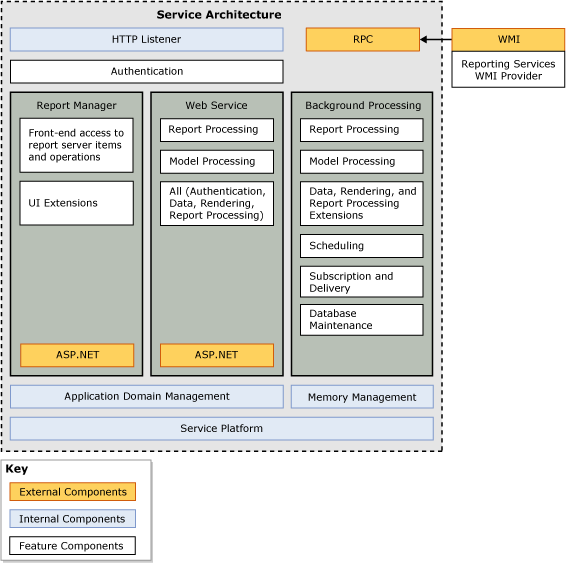
Many standard .NET Framework data providers are available from third parties. For more information, search the third-party Web sites or forums.

In Reporting Services, a report server is implemented as a Windows service that consists of distinct feature areas that run in separate application domains. The service hosts Report Manager, the Report Server Web service, and background processing feature areas.

This topic describes the composition of the service so that you can make informed choices about which features to enable and how to troubleshoot any issues that occur.

http://i.msdn.microsoft.com/Global/Images/clear.gif Architecture Diagram

The following diagram shows the service architecture in Reporting Services.



**Creating a Report Server Database**

the databases are named **reportserver** and **reportservertempdb**.

* SQL Server 2005 or SQL Server 2008 can be used to host the databases. Do not use SQL Server 2000 or earlier versions of SQL Server.
* The SQL Server Database Engine instance can be a local or remote instance.

**SQL Server Reporting Services** (**SSRS**) is a server-based **report** generation **...** **NET** ReportViewer **web** control or the ReportViewer **Windows Forms** control.

**How to: Turn Reporting Services Features On or Off**

Similarly, you can run just the Report Server Web service if you only want interactive, on-demand reporting.

* [Report Server Web service](http://msdn.microsoft.com/en-us/library/bb677363.aspx#RSWebSvc)
* [Scheduled events and processing](http://msdn.microsoft.com/en-us/library/bb677363.aspx#Sched)
* [Report Manager](http://msdn.microsoft.com/en-us/library/bb677363.aspx#ReportManager)
* [Report Builder](http://msdn.microsoft.com/en-us/library/bb677363.aspx#ReportBuilder)
* [Windows Integrated security for report data sources](http://msdn.microsoft.com/en-us/library/bb677363.aspx#WinIntSec)

1. [Applied Microsoft *SQL Server* 2008 *Reporting Services*](http://www.google.co.in/url?sa=t&rct=j&q=SQL+Server+Reporting+Services+Leverages+the+infrastructure+of+the+.NET+framework+to+sandbox+custom+code+execution+and+prevent+malicious+code+from+performing+unauthorized+access&source=web&cd=7&ved=0CE8QFjAG&url=http%3A%2F%2Fwww.scribd.com%2Fdoc%2F18671165%2FApplied-Microsoft-SQL-Server-2008-Reporting-Services&ei=PJMaT6-NAcKGrAfc9enGDQ&usg=AFQjCNGoRSgnU1m3uy1hprV-2_fLhhPFjg)

www.scribd.com/.../Applied-Microsoft-**SQL**-**Server**-2008-**Reporting**-...

Previously, rendering was **performed** entirely on the **server** and the report clients, **......** **Reporting Services leverages** the **Code Access** Security (CAS) **infrastructure** **...** **NET Framework** to **sandbox custom code execution** and to **prevent malicious** **...**

**Interactive Sorting, Document Maps, and Links (Report Builder 3.0 and SSRS)**

In Web-based environments, you can add a number of features that let your users interact with reports. Your users can change the sort order of values in your report, show or hide items in the report, or click links that go to other reports or Web pages. You can also add a table of contents or document map. Your report users can click items in the table of contents or document map to jump to areas within a report.

Report Builder and Report Designer support three types of links with the following actions:

* **Bookmark links**   Jump to other areas within the report.
* **Hyperlinks**   Jump to URLs that specify the address of Web pages or reports on a report server by using URL access.
* **Drillthrough report links**   Jump to other reports on the same report server. For more information, see [Drillthrough Reports (Report Builder 3.0 and SSRS)](http://technet.microsoft.com/en-us/library/ff519554.aspx).

#### Table 1.2 RS supports various report types

| **Report Type** | **Purpose** | **Example** |
| --- | --- | --- |
| Tabular | Displays data in a table format with a fixed number or rows and columns. | Excel-type reports |
| Freeform | Data regions are positioned arbitrarily on the page by the report author. | Invoice-invoice details report |
| Chart | Presents data graphically. | Employee performance chart |
| Crosstab (matrix) | Data is rotated to present row data as columns. | A report that shows products on rows and time on columns |
| Drilldown | Includes expandable sections. | A company performance crosstab report where product can be expanded by category and brand |
| Drillthrough | Generated from clicking on a hyper-link. | Customer Order History with hyperlinks on the order identifier to show the order details report |
| Interactive | Includes interactive features, such as document maps, hyperlinks, visible-on-demand sections, and so forth. | Adobe Acrobat.type reports with document |

**XML Using Web Services**

To support service-oriented architectures, the XML data provider can query Web services directly by parsing the XML structure of the SOAP response directly. This requires knowledge of the Web service structure, which includes the namespace, method, SOAP Action, parameters, and schema of the response body.

When you access these data sources, you must specify what data is needed for retrieval. You can use the XML Data Provider Query language outlined in the next section to do this.

| **RunningRequestsDbCycle** | Specifies how often the report server evaluates running jobs to check whether they have exceeded report execution time outs, and when to present running job information in the Manage Jobs page of Report Manager. This value is specified in seconds. Valid values range from 0 to 2147483647. The default is 60. |
| --- | --- |

http://msdn.microsoft.com/en-us/library/ms157273.aspx

**Reporting Services** ships with 6 built-in **roles**: System Administrator, System **User**, Browser, **Content Manager**, My **Reports**, and Publisher.

Initially, only users who are members of the local administrators group can access a report server. Reporting Services is installed with two default role assignments that grant item-level and system-level access to members of the local administrators group. These built-in role assignments local Administrators to grant report server access to other users and manage report server items. The built-in role assignments cannot be deleted. A local administrator always has permission to fully manage a report server instance.

Because full permissions on a report server include item-level and system-level permissions, a local administrator is assigned to the following roles:

* [System Administrator Role](http://msdn.microsoft.com/en-us/library/ms156470.aspx)
* [Content Manager Role](http://msdn.microsoft.com/en-us/library/ms159693.aspx)

Additional configuration is required before you can administer a report server instance on a local computer that runs Windows Vista or Windows Server 2008.

The **Content Manager** role is often used with the **System Administrator** role. Together, the two role definitions provide a complete set of tasks for users who require full access to all items on a report server. Although the **Content Manager** role provides full access to reports, report models, folders, and other items within the folder hierarchy, it does not provide access to site-level items or operations. Tasks such as creating and managing shared schedules, setting server properties, and managing role definitions are system-level tasks that are included in the **System Administrator** role

http://msdn.microsoft.com/en-us/library/ms159693.aspx