

# Reporting Services Architecture

SSRS is a comprehensive and extensible reporting platform, which includes an integrated set of processing components, programmatic interfaces, and tools. Processing components are the basis for the multilayered architecture of SSRS and interact with each other to retrieve data, process layout, render, and deliver a report to a target destination. SSRS supports two categories of components:

- **Processors**—Ensure integrity of SSRS and supply an infrastructure that enables developers to add a new functionality (extensions). Processors themselves are not extendable in this release of SSRS.
- **Extensions**—Assemblies that are invoked by processors and perform specific processing functionality, such as data retrieval. Developers can write custom extensions.

Reporting Services architecture diagram is depicted in Figure 3.1; components are described in more details later in this chapter. Arrows on the diagram show a data flow between components “within” and “outside” of SSRS (external tools, applications, services, and utilities).

## CHAPTER 3

### IN THIS CHAPTER

- Report Server Web and Windows Service
- Programmatic Interfaces
- Report Processor
- Command-Line Utilities
- Reporting Services Extensions
- Report Server Databases
- Scheduling and Delivery Processor
- Report Builder NEW in 2005
- Report Model Designer NEW in 2005
- Report Designer
- Report Manager
- SQL Server Management Studio NEW in 2005
- Reporting Services Configuration Tool NEW in 2005
- RSPrintClient Control NEW in 2005
- WMI Provider
- Performance Monitoring Objects

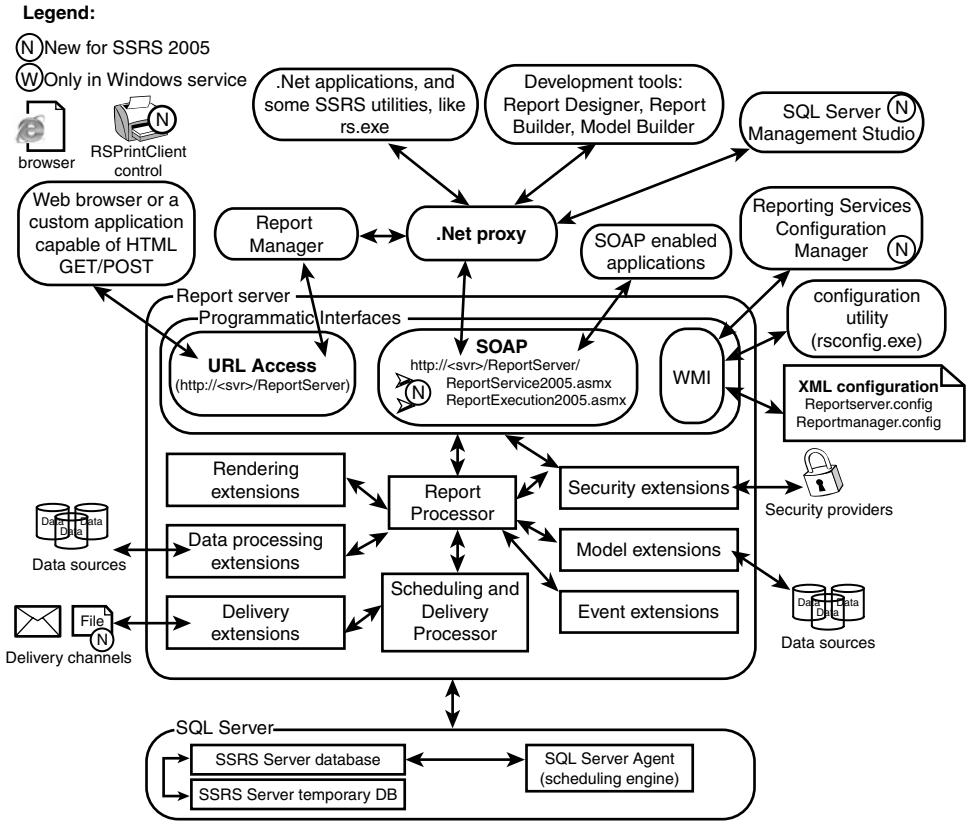


FIGURE 3.1 Reporting Services architecture.

## Report Server Web and Windows Service

The Report Server is implemented as a symbiosis between a web service and Windows service. Services work together to host, process, and deliver reports.

### NOTE

In SSRS 2005, Report Server supports multiple instances on a single computer.

When SSRS Windows service starts, it initially loads basic supporting assemblies (such as SQL data provider and interface library), but it does not load extensions. In this release, SSRS Windows service handles encryption and decryption and serves as a host for the Scheduling and Delivery Processor. The Scheduling and Delivery Processor uses a `PollingInterval` configuration setting to monitor SSRS catalog's Event table for "events" (such as timed subscription). Events instruct SSRS to "wake up" and start processing.

Similar to any .NET web service, SSRS web service is invoked (“wakened up”) via Internet Information Server (IIS) and executes in the ASP.NET worker process (`aspnet_wp.exe`). “Wake up” events are facilitated by programmatic interfaces.

## Programmatic Interfaces

The Reporting Services programmatic interfaces accept SOAP (via SSRS web service) and HTTP requests (via URL Access). For more information about SSRS web service, please see Chapter 25, “How to Use Reporting Services Web Service.” For more information about SSRS URL Access, please see Chapter 24, “How to Use URL Access.”

SSRS 2005 web service provides three endpoints:

- `http://localhost/ReportServer/ReportService.asmx` is provided for backward compatibility with SSRS 2000.
- `http://<server>/ReportServer/ReportService2005.asmx` is the new management endpoint in SSRS 2005.
- `http://<server>/ReportServer/ReportExecution2005.asmx` is the new execution endpoint.

The name of an endpoint describes the purpose. For example, the execution endpoint is designed to provide report execution (processing) interfaces and, with that, an access to functions, such as `Render`. Function `Render` returns a rendered report as a stream with a specified format, such as HTML.

Programmatic interfaces facilitate retrieval of information from the SSRS catalog and information exchange between SSRS components.

When a report is requested, either interactively or by the Scheduling and Delivery Processor, programmatic interfaces initialize the Report Processor component and start processing a report.

## Report Processor

The Report Processor ties components of a report server together and takes care of caching within SSRS. Caching refers to an ability of the SSRS to keep a copy of a processed report and return that copy when a user opens the report. Caching can shorten the time required to retrieve a report, especially if the report is large or is accessed frequently. All report caches are stored in the SSRS catalog (specifically `ReportServerTempDB` database) and survive both SQL Server and Report Server restarts.

Report Processor performs the following operations:

- **Execution**—Retrieves a report definition and combines it with data retrieved by the data-processing extension. This operation generates an intermediate format.
- **Rendering**—Renders the intermediate format to a requested output format using rendering extensions.

- **Processing of models**—This is similar to Execution operation for reports that are generated by Report Builder and contain a semantic model (or simply a model, which serves as a data source for a report) and a semantic query. Semantic query refers to a query performed against a model that, in turn, just like a SQL query, generates a report's data set.

This is how Report Processor responds to users' requests:

- **New interactive report request**—Intermediate format is generated and passed to the rendering extension; user receives rendered report.
- **Request to generate cache or snapshot**—Intermediate format is generated and stored in the database.
- **Request for cached report or snapshot**—Intermediate format is retrieved from cache (or snapshot) and passed to the rendering extension; user receives rendered report.

## Command-Line Utilities

Three administration assistance utilities are installed automatically during the Reporting Services install:

- **rs.exe**—Host scripting operations. Developers can, for example, create VB.NET script to deploy a set of reports. You can find more details about this utility in Chapter 25.
- **rsconfig.exe**—Use to modify encrypted connection information to the Report Server database.
- **rskeymgmt.exe**—Back up/restore symmetric keys for encrypted data used by a report server or delete encrypted data if the key is lost. For more details, please see Chapter 21, "Deploying and Configuring SSRS."

---

### NOTE

SSRS 2005 discontinued the `rsactivate.exe` utility, which was used in the previous release to activate new SSRS instances in a web farm. In SSRS 2005, activation is performed using the Reporting Services Configuration Tool.

## 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.

## Report Server Databases

The SSRS catalog encompasses two databases: Report Server database (the default name is ReportServer) and Report Server temporary database (the default name is ReportServerTempDB). Report Server database is a SQL Server database that stores part of SSRS configuration, report definitions, report metadata, report history, cache policy, snapshots, resources, security settings, encrypted data, scheduling and delivery data, and extension information.

---

### NOTE

Although users can certainly directly access databases in the SSRS catalog and directly modify objects that SSRS uses, this is not a recommended (or supported) practice. Underlying data and structures within the SSRS catalog are not guaranteed to be compatible between different releases of SSRS, service packs, or patches.

Please treat the ReportServer database as one of the production databases. Although many developers store report definitions (RDL) in a separate repository, and, thus, RDL is often recoverable, a loss of snapshot data can carry a negative business impact. For example, users might make some business decisions using snapshot’s capabilities to report “frozen-in-time” data.

Another database that SSRS uses is the Report Server temporary database. This database is responsible for storing intermediate processing products, such as cached reports, and session and execution data.

**NOTE**

To store temporary snapshots in the file system, instead of the database, administrators should complete the following steps. First modify `RSReportServer.config` and set `WebServiceUseFileShareStorage` and `WindowsServiceUseFileShareStorage` to `True`.

Then set `FileShareStorageLocation` to a fully qualified path. The default path is `C:\Program Files\Microsoft SQL Server\MSSQL.3\Reporting Services\RSTempFiles`.

Unlike SQL Server's `tempdb`, data in `ReportServerTempDB` survives SQL Server and Report Server restarts. Report Server periodically cleans expired and orphan data in `ReportServerTempDB`.

All data in `ReportServerTempDB` can be deleted at any time with minimal or no impact. The minimal impact that a user might experience, for example, is a temporary performance reduction due to lost cache data and a loss of an execution state. The execution state is stored in the table `SessionData`. Loss of the execution state results in an error: "Execution 'j4j3vfb1canzv3qzcqhvm155' cannot be found (rsExecutionNotFound)." To resolve the loss of the execution state, a user would need to reopen a report.

**TIP**

SSRS does not recover deleted `ReportServerTempDB` or tables within this database. To quickly recover from erroneous deletions of objects in this database, keep a script or a backup of an empty `ReportServerTempDB` handy.

In a scale-out deployment, the SSRS catalog is shared across all of the report servers in the deployment.

## Scheduling and Delivery Processor

The Scheduling and Delivery Processor is hosted in SSRS Windows service and monitors for events. When the Scheduling and Delivery Processor receives an event, the Scheduling and Delivery Processor collaborates with the Report Processor to render a report. After a report is rendered, the Scheduling and Delivery Processor uses delivery extensions to deliver a report.

The Scheduling and Delivery Processor leverages the SQL Server Agent as a scheduling engine. The schedule is based on the local time of the Report Server that owns the schedule. When an administrator creates a new schedule, the SSRS creates a SQL Server Agent job to run on the requested schedule. Then SSRS adds a row in the `Schedule` table of the `ReportServer` database. The row's `ScheduleId` field is the job's identifier. Administrators can schedule subscriptions, report history, and snapshot execution.

When the scheduled time comes, the SQL Server Agent generates an event by executing the scheduled job. The job inserts a row in the `Event` table of the `ReportServer` database. This row serves as an event for the Scheduling and Delivery Processor.



The Scheduling and Delivery Processor checks the Event table every `PollingInterval` seconds and initiates appropriate actions as a response to an event.

---

**NOTE**

The `PollingInterval` is specified in the `rsreportserver.config` configuration file and is set to 10 seconds by default.

The scheduling and delivery process “breaks” when either (or both) the SSRS Windows service is not running (the Scheduling and Delivery Processor is not processing events) or the SQL Server Agent is not running (the agent is not generating events).

---

**NOTE**

When the SSRS Windows service is not running and the SQL Server Agent is running, the job history for SQL Server Agent would indicate that the scheduled request (“insert event”) ran successfully. The job will be successful despite the fact that the scheduled operation could not complete because the Scheduling and Delivery Processor is not running to process the event.

## Report Builder NEW in 2005

One of the most requested features in the previous version of SSRS was an ability to develop end-user reports. Microsoft delivered this functionality in SSRS 2005.

Report Builder is a ClickOnce, ad hoc, end-user report authoring and publishing tool, which provides drag-and-drop, easy-to-use report design functionality.

---

**NOTE**

You can find more information about ClickOnce applications by searching <http://www.microsoft.com> and reading <http://msdn.microsoft.com/msdnmag/issues/04/05/clickonce/default.aspx>.

As a typical ClickOnce application, Report Builder is deployed from a browser and executes on a client’s computer. Report Builder does not require administrative permissions during installation and runs in a secure sandbox provided by .NET code access security.

To deploy Report Builder, click on the Report Builder button on the Report Manager’s toolbar. Alternatively, you can use `http://<server>/ReportServer/ReportBuilder/ReportBuilder.application` URL to launch Report Builder. Report Builder is deployed to `C:\Documents and Settings\<UserName>\Local Settings\Apps\2.0\<obfuscated directory>`.

Before you can use Report Builder,

- You must have appropriate permissions, and be a member of the Report Consumer role or a custom role that includes the Consume Reports task.
- At least one report model has to be published.
- An Internet browser must allow you to download files.

Reports developed by Report Builder can be interpreted by Report Designer.

## Report Model Designer *NEW in 2005*

The Report Model Designer generates report models for use by Report Builder. A model abstracts complexities of an underlying data; for example, a model allows mapping names of tables and columns to business terms that an end user can easily understand.

The Report Model Designer is hosted in Business Intelligence Development Studio (BIDS for short) or Visual Studio and is intended for use by developers. Actually, BIDS is a Visual Studio shell with only Business Intelligence projects and no language projects. One of the Business Intelligence projects is the Report Model Project, which launches the Report Model Designer and allows developers to create models.

In this release, models and, therefore, ad hoc reports can only work with SQL Server data sources: SQL Server database engine and SQL Server Analysis Services. However, developers can work around this limitation and access other data sources by using link servers or Analysis Services Unified Data Model. Both provide a thin layer of abstraction and allow access to any OLE DB- or ODBC-compliant data source including Oracle.

## Report Designer

Report Designer is a developer-oriented, comprehensive report authoring, previewing, and publishing tool hosted in Business Intelligence Development Studio or Visual Studio.

To organize the report development process, Report Designer provides three tab views of a report: Data, Layout, and Preview.

The Data tab helps developers to define data sources and design data set queries. Report Designer provides three drag-and-drop graphical query designers to assist with SQL queries, Analysis Services Multidimensional Expressions (MDX query designer is a new feature of SSRS 2005), and Analysis Services Data Mining Expressions (DMX query designer is a new feature of SSRS 2005) .

The Layout tab helps developers to design graphical presentations of a report and associate graphical presentation with data. Report Designer provides a drag-and-drop layout designer and toolbox with reporting controls. Layout design is very similar to a user interface design that Visual Studio provides for Windows and web applications: You can drag and drop reporting controls to a report, arrange them as needed, set properties, and associate with data sets that were designed through the Data tab.

The Preview tab provides a preview for a report so developers can test and adjust the report as needed.

Report Designer provides the Report Wizard that takes developers through the guided steps to create a report. As such, the wizard provides a limited number of layouts to choose from, but a report developer can modify the layout as needed by using the Layout tab after the wizard's steps are completed.

Finally, Report Designer allows developers to build and deploy reports to SSRS.

---

**NOTE**

Reports developed by Report Designer cannot be interpreted or edited by Report Builder.

## Report Manager

Report Manager is a web-based report access and management tool providing access to a single instance of a Report Server. Some of the operations that users can perform using the Report Manager include view, search and subscribe to reports, manage security (report access and roles), create folders and move reports around folders, manage data sources, set report parameters, and more. Actions that a user can perform using Report Manager depend on the user's security permissions. The default URL that invokes Report Manager is <http://<server>/reports>. The default directory that contains the Report Manager's binaries, pages, and so on is `C:\Program Files\Microsoft SQL Server\MSSQL.3\Reporting Services\ReportManager`.

Although Report Manager provides for a limited customization, it is neither designed nor supplied with sufficient documentation to support customization. This leaves companies with several customization options, which can be combined:

- Accept limited customization capabilities of Report Manager, such as modification of style sheets it uses (by default located at `C:\Program Files\Microsoft SQL Server\MSSQL.3\Reporting Services\ReportManager\Styles`) and adjusting the name the Report Manager displays through the site settings (<http://<server>/Reports/Pages/Settings.aspx>).
- Understand how Report Manager functions internally through the use of classes in the `ReportingServicesWebUserInterface` assembly and leverage its undocumented functionality.
- Write custom management pages to replace one or more management pages in Report Manager (by default located at `C:\Program Files\Microsoft SQL Server\MSSQL.3\Reporting Services\ReportManager\Pages`).
- Write a custom façade that would display a company's information and eventually take a user to the Report Manager pages.
- Write a custom report management application to replace Report Manager.

## SQL Server Management Studio *NEW in 2005*

SQL Server Management Studio provides a Windows form-based, integrated environment that can manage various SQL Server components. From the SSRS perspective, the Management Studio has similar functionality to Report Manager when used to manage a single instance of SSRS.

The advantages of using the SQL Server Management Studio include consolidated content view for SSRS web farm (scale-out) deployment, slightly better performance, an ability to script and replay administrative tasks, and a finer granularity for role-based security settings.

---

### TIP

Use the SQL Server Management Studio for a consolidated view of a SSRS web farm.

## Reporting Services Configuration Tool *NEW in 2005*

The Reporting Services Configuration Tool is a Windows form application that can be used to start and stop the Report Server Windows Service and reconfigure report servers. For example, administrators can change the Report Server's database and SQL Server names, change SSRS' Windows service identity, and change the virtual directories used to access the Report Server and Report Manager. Administrators can start the Reporting Services Configuration Tool from SQL Server 2005, Configuration Tools, Reporting Services Configuration menu or from the SQL Server Configuration Manager using the Configure button in the SQL Server Reporting Services Properties dialog box.

## RSPrintClient Control *NEW in 2005*

The RSPrintClient ActiveX control provides client-side printing for reports viewed in Report Manager. The control presents the Print dialog box for a user to initiate a print job, preview a report, specify pages to print, and change the margins. Developers can access this control programmatically in the code to enable report-printing functionality in their applications.

## WMI Provider

SSRS includes a WMI provider that maps SSRS' XML configuration files to a set of classes to simplify configuration management of the Report Server and Report Manager, and to minimize configuration errors. WMI provider also supplies a class that provides basic properties and status information for a SSRS instance, and thus assists with discovery of SSRS instances on a network.

Both the Reporting Services Configuration Tool and the `rsconfig.exe` utility use the SSRS WMI provider.

## Performance Monitoring Objects

SSRS Windows and Web Service include performance objects that supply performance counters that provide information about report processing and resource consumption. The objects are called RS Windows Service and RS Web Service, respectively.

---

### NOTE

In SSRS 2000, the web service performance object was called Reporting Services and the Windows service performance object was called Delivery Processor.

To have a more complete picture and to gather more information, an administrator can also monitor SQL Server, ASP.NET, Processor, Memory, and Physical or Logical Disk counters.

## Summary

This chapter discussed the SQL Server Reporting Services (SSRS) architecture. Table 3.3 presents the SSRS components summary.

**TABLE 3.3 Reporting Services Components Summary**

Component	Brief Description
Programmatic interfaces	Provides access to SSRS functionality through SOAP and HTTP requests.
Report Processor	Facilitates a set of report generation operations from data retrieval to rendering. Report Processor invokes other components, such as data extensions to assist with report generation.
Data-processing extensions	Retrieves report's data from a data source. Developers can develop additional custom data-processing extensions.
Command-line utilities	Three utilities, designed to assist with scripting of administrative tasks, installed automatically during the Reporting Services install.
Rendering extensions	Transform the report's intermediate format (a combination of report's layout and data) into a device-specific format, such as HTML. Developers can create new rendering extensions.
Report Server database	Stores report definitions, report metadata, report history, cached reports, snapshots, resources, security settings, encrypted data, scheduling and delivery data, and more.
Report Server temporary database	Stores intermediate processing products, cache, and data.
Scheduling and Delivery Processor	Monitors for events (such as timed subscription), collaborates with Report Processor to render a report, and delivery extensions to deliver scheduled reports to a location specified in the subscription.

TABLE 3.3 Continued

Component	Brief Description
Delivery extensions	Deliver reports to specific devices, such as email or a file system. Developers can create new delivery extensions.
Report Manager	Provides web-based report access and management capabilities. Default URL that invokes Report Manager is <code>http://&lt;server&gt;/reports</code> .
Report Model Designer	Generates report models for use in Report Builder.
Report Builder	Provides drag-and-drop, easy-to-use report design functionality. Report Builder is an ad hoc end-user report authoring and publishing tool executed on a client computer.
Report Designer	Allows developers to develop complex reports. Report Designer is a comprehensive report authoring and publishing tool, hosted in Business Intelligence Development Studio or Visual Studio.
Security extensions	Enable authentication and authorization of users and groups. Developers can (excluding SQL Server Express edition) create new security extensions.
SQL Server Management Studio	Provides administrators with Windows form-based, integrated environment to manage SQL Server components including SSRS. From the report management perspective, Management Studio has similar functionality to Report Manager, but provides additional capabilities, such as consolidated web-farm management.
Reporting Services Configuration Tool	Provide administrators with functionality to start and stop Report Server Windows service and reconfigure report servers. This is a Windows forms application.
WMI provider	Provides a set of WMI interfaces to manage settings of a Report Server and assists with SSRS instance discovery on a network.
Performance monitoring objects	Provide a view of SSRS Windows and web service performance.

In the next chapter, authors discuss various SSRS deployment scenarios and features of SSRS editions.