

CHAPTER 5

Reporting Services Deployment Scenarios

This chapter provides an overview of Reporting Services deployment scenarios (including Internet deployment), including SSRS hardware and software requirements, licensing, and security. More technical information about security is covered in Chapter 20, “Securing Report Server Items.”

NOTE

Although the test (staging) environment might not be as “powerful” as production, it is best to have a total match for the most effective and realistic scalability testing.

In a SQL Server Reporting Services enterprise production environment, support for web farms and scale-up capabilities of Enterprise Edition come in handy for high-volume reporting. Web farm deployment is flexible and enables administrators to add capacity to a Report Server web farm as demand grows. In addition, if one of the servers in the web farm fails, the remaining servers pick up the load. Thus, a web farm provides high availability for a report-processing layer, but not the SSRS catalog (database).

To achieve complete high availability for a reporting solution, a company can install a Reporting Services catalog on a SQL Server 2008 cluster.

For an environment that does not have high-performance or -availability requirements, you can simplify deployment and use a single Report Server instance with a catalog placed in a nonclustered instance of SQL Server 2008.

IN THIS CHAPTER

- ▶ High-Availability Deployment Considerations
- ▶ Internet Deployment Considerations
- ▶ Minimum Hardware Requirements
- ▶ Software Requirements
- ▶ Key Features of SSRS 2008 Editions
- ▶ Licensing

You can further simplify deployment in a development environment, install all the Reporting Services components on a single server, and install development tools on a set of workstations.

If a developer or a user needs to be completely mobile, that user can install all the necessary components and a subset of data sources on a laptop, as depicted in the Single Server Deployment in Figure 5.1.

NOTE

There is no separate Books Online for SSRS. Books Online covers all the SQL Server 2008 components: Reporting Services, SQL Server engine, T-SQL, and so on.

SSRS is a fairly memory- and CPU-intensive application. It is hard to be precise with the exact hardware configuration that an administrator might need for installation. Table 5.1 presents approximate CPU needs that depend on the number of concurrent users.

TABLE 5.1 Estimates of Reporting Server CPUs Needs

Concurrent Users	Approximate Number of CPUs
< 150	1
< 700	2
700 > < 2,000	4–8
2,000 > < 4,000	8–16
4,000 >	16+

Table 5.1 provides estimates for a 3GHz 32-bit Intel Xeon CPU server and is based on SSRS performance for rendering a report of an average layout complexity, which retrieves approximately 5,000 rows of data from a data source and provides users with HTML output and reasonable completion times of no more than 25 to 30 seconds. The data source used in this analysis is well tuned and available without significant latency.

Keep in mind that your results will likely differ from the result in the table. A test is the best way to determine precise configuration needs best suited for your deployment scenario.

Configuration tips that you might want to consider when deploying SSRS (or specifically a Report Server) include the following:

- ▶ A 32-bit instance of a Report Server can use memory up to 3GB (requires the /3GB switch in boot .ini). Because of this, efficient hardware use would be at 4GB per instance (3GB for a Report Server and 1GB for the OS). To effectively use servers with larger amounts of memory, consider installing multiple instances of SSRS per server.
- ▶ For performance, start with scaling up (fastest CPU available, 4GB of RAM, and capable I/O subsystem), then move to scale out, and add capacity as necessary (add

Report Servers to a web farm). Host the Report Server catalog in a SQL Server instance on a separate box from your data sources (transactional, data warehouse, or line-of-business database) or at least make sure that a SQL Server instance can handle additional workload.

- ▶ For scale-up scenarios, SSRS 2008 supports a 64-bit platform for both x64 (Opteron, Athlon64, and Xeon EMT64T CPUs) and IA64 (Itanium CPU). A 64-bit platform overcomes the 4GB memory limitation of the 32-bit platform and should be considered for reporting applications with high memory demand. A reporting application that renders a fair amount of or large Microsoft Excel or PDF reports is an example of a high-memory-demand application.
- ▶ For reliability, use redundant components: at least two SSRS web servers and a database cluster for the Reporting Services catalog database, redundant disk arrays, and network pathways. Although high availability requires at least two servers, three is better. With three servers, you can do maintenance on one of the servers and still have a high-availability configuration running in your environment.
- ▶ For cost evaluation when deciding whether to buy more servers with a smaller number of CPUs versus fewer servers with a larger number of CPUs in each, consider the price of the hardware, the additional costs associated with extra servers, and the cost of a reporting-solution failure. As the number of servers grows, so do the server management overhead and other costs, such as the cost of additional space, cooling, and energy.

High-Availability Deployment Considerations

To create a highly available Reporting Services installation, an administrator can deploy Reporting Services on a web farm and use clustering for the Reporting Services catalog database. Enterprise Edition of Reporting Services is the only edition that supports web farm deployment in the production environment. Developer Edition and Evaluation Edition can be deployed on a web farm, but only in a testing environment. No other editions support the web farm feature.

Although the Enterprise Edition of SSRS supports a web farm, it does not include a functionality to create and manage a web farm. This is why a company would have to use separate software (or hardware) to create and manage a web farm. An example of web farm management software is the Network Load Balancing (NLB) feature of Windows Server. The steps to install Reporting Services on a web farm (scale-out configuration) are covered in Chapter 6, "Installing Reporting Services."

To protect the catalog database, companies can deploy a SQL Server 2008 cluster. If Windows authentication is being used between the Report Server and the SQL Server 2008, both Report Server and the SQL Server 2008 cluster have to be in either the same or in the trusted domains. Both nodes of the SQL Server 2008 cluster must have an exact match and all hardware and software installed on a cluster must be supported.

Alternative high-availability options can be used to protect from a database server failure: hardware-based data replication or peer-to-peer replication in SQL Server 2008.

NOTE

The database mirroring functionality of SQL Server 2008 is another high-availability option.

Overview of Deployment Scenarios

SSRS has two main deployment scenarios. The first is possibly the simplest: the single-server deployment. In this scenario, a single machine is responsible for hosting both major components of SSRS: the database and the Report Server.

The second major scenario is the scale-out deployment, in which the database is on one machine, possibly a clustered virtual machine, and the Report Server is on another machine or on a web farm.

Figure 5.1 shows a sample SSRS deployment. When administrators install SSRS, they have a choice to install one or more client- and server-side components, as outlined in Table 5.2.

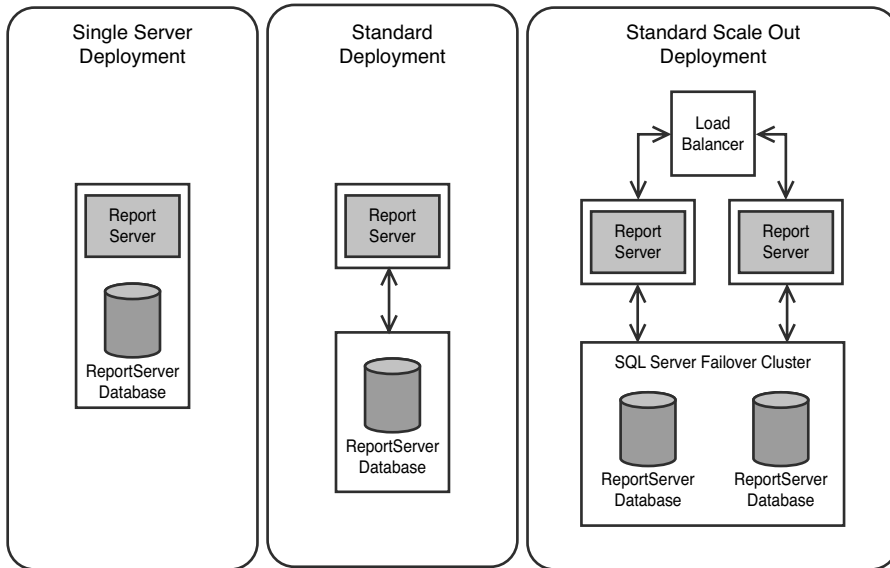


FIGURE 5.1 Deployment scenarios.

TABLE 5.2 Reporting Services Deployable Elements

Component	Approximate Size	Typical Install Location
Reporting Services	230MB	Deployed on the server

TABLE 5.2 Continued

Component	Approximate Size	Typical Install Location
Books Online	160MB	Developer's or administrator's workstation
Basic management tools - command-line tools	880MB	Developer's or administrator's workstation
SQL Server Management Studio (includes basic management tools)	900MB	Developer's or administrator's workstation, .NET Framework
Business Intelligence Development Studio	1GB	Developer's workstation

SSRS 2008 added the ability to separate out servers to do simply scheduled batch or subscription processing. Figure 5.2 shows an advanced scale-out scenario where servers are isolated for doing simply on-demand or batch processing.

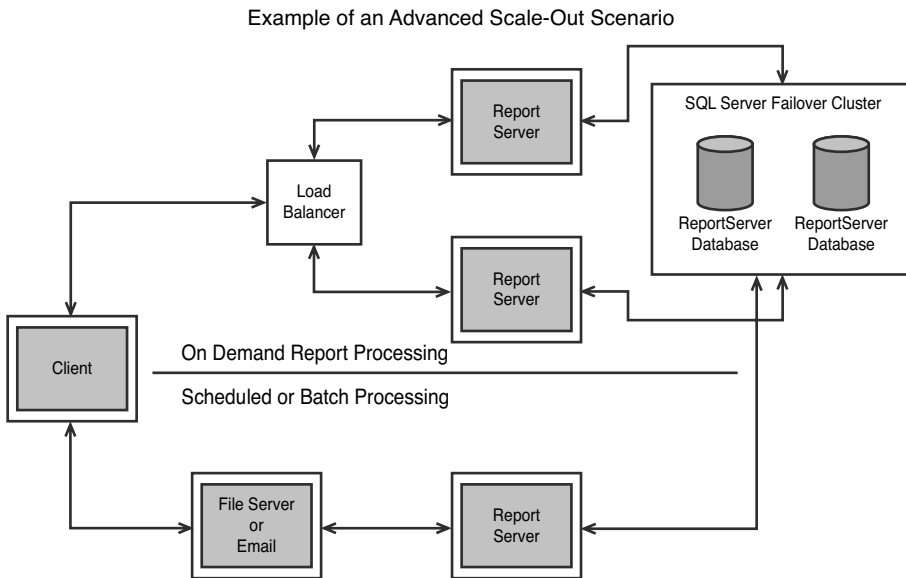


FIGURE 5.2 Advanced deployment scenario.

Advantages/Disadvantages of the Standard Model

The standard model, or single-server deployment model, might sound simple and easy to do at first, and it is certainly the way to do it for a development workstation, or a simple trial or proof of concept. However, you should consider a couple of things when debating whether to use this model in a production environment.

Performance Impact of the Standard Model

The primary consideration for most administrators after cost is performance. Having both the database and the Report Server on the same machine might sound tempting on the financial front because SSRS is included with the SQL Server relational engine. However, both the relational engine and Report Server love RAM and CPU cycles. Although SSRS 2008 has made huge strides in rendering efficiency, SSRS is still going to use all the RAM it can get or whatever it needs (the lower of the two numbers) to render a report. Rendering reports, and especially rendering large reports, also chews up lots of CPU cycles. Adding this overhead to an older machine that is already struggling with the database server is not advisable.

Disk Space Requirements for SSRS

Anyone who has known a DBA, or who has been one, knows there is one thing all DBAs love: storage. They just can't seem to get enough of it. Even in today's environments with large storage area networks (SANs) and hundreds of spindles, the DBA always wants more. This is for good reason.

SSRS, like most databases, installs with a very small footprint. It's almost, and possibly is, negligible. However, depending on how SSRS is used, the disk space requirements can grow pretty large. To understand how space is used inside the SSRS database, an overview of the different types of objects and how they are stored is required.

By now, it should be understood that the SSRS database holds the Report Definition Language (RDL) files, data sources, models, and all metadata, such as folders and access control lists (ACLs). This might seem like a lot to store, but in reality this is rather small, and only in the most extreme cases should this cause issues. Session state information for SSRS is stored in the Report Server temporary database. Because only one row is generated per user session, this should not get very large, and grows at a predictable rate.

Other things stored in the database can, however, grow to be very large. Resources for reports are stored in the catalog as a binary large object (BLOB). It's a sure bet that your friendly neighborhood DBA hates BLOBs. When a BLOB is stored initially with the report RDL, it might not be such a big deal. However, if a resource is stored as part of a report in an archive solution, this can get very large very quickly. Cached reports or temporary snapshots are stored in the Report Server temporary database as a BLOB in intermediate format. Because cached reports include raw query results, the BLOB can get pretty large. Another disk space consideration when using cached reports with parameterized reports is that a separate copy of the cached report is generated for each combination of report parameters. The bottom line is that if you are using temporary snapshots, prepare to use disk space. In addition, you must consider report history snapshots, too. The only difference between them and temporary snapshots is that the report history is saved inside the Report Server database and not inside the Report Server temporary database.

Availability Impact of Standalone Deployment

If the performance impact of the single-server deployment can be shrugged off, the availability impact of it can't be. Having one machine be the central data store and Report Server creates a single point of failure in an enterprise environment. This makes having a backup essential to save the system from some unforeseen calamity. Not much more can

be said about it. It is up to the administrator to decide how critical the functionality SSRS provides is. If it can be down for as much time as needed to restore from tape, or if SSRS is not yet important enough to be deployed in a redundant manner, a standalone deployment should suffice.

Advantages/Disadvantages of the Scale-Out Model

The scale-out model of deployment has two main advantages over the standalone model: performance and availability. However, it has one major downside: cost. Because in the scale-out model the database server is separate from the web server, the performance penalty of combining the database engine with the Report Server's rendering engine gets nullified. In addition, the database can be clustered in a virtual server to provide high availability.

With modern SAN technologies, the database can even be replicated to a remote site. The SSRS application server lives on a separate server. The server is simply the first node in what could become an NLB cluster. The cluster makes it possible to scale out for performance/availability or both. Scaling out also helps with dispersing the workload generated by scheduled subscriptions, because each machine on the cluster looks for events that trigger a subscription to process. The cluster also allows one node to be removed for upgrades/maintenance and then be placed back online when the maintenance is complete.

NOTE

NLB clusters are not a function of SSRS. Instead, they are a function of the OS or hardware. SSRS is just an application that can be placed on an existing NLB cluster.

All of this flexibility comes at a price (literally). The only editions to support a scale-out deployment are Developer and Enterprise. Microsoft does not offer support for the Developer Edition, and does not license it for use in a production environment. In addition, every machine in a scale-out deployment has to be licensed separately for Enterprise Edition. More than anything, the cost of a scale out is what keeps most shops from adopting it.

Requirements for a Standard Deployment

In a standard deployment, the web server/application server and the database server are installed on the same machine. For this reason, it is important that the minimum hardware requirements be met or exceeded. It is also helpful to have the NetBIOS name or IP address of the Simple Mail Transfer Protocol (SMTP) server handy and the service account used to execute the reports in unattended mode and the credentials with which to log in to the database.

After collecting all the necessary information, you just need to run setup and configure the Report Server. Sounds easy, doesn't it? While running, the installation program offers two main options. The first option is the default installation. This is the option used for running the standard deployment. This option sets up the database server and the Report

Server on the same machine. The second option is called the Files Only option. This option is used primarily in scale-out deployments. For the brave or simply curious, this option can be used to set up SSRS locally; however, the administrator must run the Report Services Configuration tool after the install completes and configure the options herself.

Requirements for a Scale-Out Deployment

As discussed earlier in this chapter, SSRS can be deployed in a scale out on a web farm. Each machine in the web farm runs SQL Server Reporting Services Windows service, which contains the Report Server web services, and the scheduling and delivery processor. As anyone who has managed a web farm knows, in theory any machine on the farm should be easily replaceable with another in the same configuration, and ideally state should not be stored on any box on the farm. SSRS accomplishes this task by using data source configuration information and reports inside the Report Server database. The application servers just need to register themselves with the database server. This might sound simple, but it is not trivial. SSRS 2008 has given administrators much better tools to aid in this configuration process.

Overview of Report Server Initialization

Because SSRS uses potentially sensitive information, it is important to secure it appropriately. In addition, in a scale-out situation, multiple Report Servers need to encrypt and decrypt the data stored in the database. To understand how SSRS accomplishes this, you need a bit of knowledge about encryption and decryption techniques.

In general, there are two kinds of encryption: symmetric and asymmetric. Symmetric is very fast because it uses only one possible key to encrypt and decrypt the data. However, this form of encryption has its drawbacks. How can you share information that has been encrypted with the symmetric key without compromising the key? The answer is to use asymmetric encryption. Asymmetric encryption uses a combination of keys, one public and one private. The public key can be shared with another host and can be used to decrypt messages encrypted with the private key. The same can be said for the private key. Asymmetric encryption is relatively slow, so it should not often be used to encrypt/decrypt.

SSRS uses both types of encryption in a simple, yet intelligent way. For every Report Server database, SSRS generates a unique symmetric key that can then be used to encrypt the data. At this point, every Report Server that needs access to the data must publish its public asymmetric key along with its unique installation ID and client ID to the Report Server database. The Report Server database then uses the public to encrypt the internal symmetric key and share it with the client. After being encrypted with the client's public asymmetric key, the symmetric key cannot be decrypted by anyone else without the private key. Administrators can actually watch this process unfold by watching the changes in the Keys table during the activation process. The process of exchanging public keys and symmetric keys is called activation.

Activation is a two-phase process. The first phase is the Announce Self phase, and the second phase is the Activated phase. The Announce Self phase covers the reading of the

keys from the Keys tables and, if needed, the writing of the client's public key to the Keys table. The Activated phase is the time the Report Server gets the symmetric key in encrypted form.

NOTE

Because the private keys are stored under the user's profile in SSRS, changing the user the service runs under could force a reactivation.

The process of adding and removing machines in the scale-out deployment model is simply the process of running activation over again. The same is true for taking an SSRS installation and pointing it to a different database.

NOTE

To use ASPNET with a web farm, the `validationKey` and `decryptionKey` should be the same on every machine in the web farm. You can find information about how to accomplish this in the Microsoft Knowledge Base article at <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q312906>.

To remove a server, just uninitialized it by opening the Reporting Services Configuration tool from any node on the cluster, select the node to be removed, and click the Remove button. To move a node, remove the node from its existing setup and follow the steps to add it to the new cluster.

Internet Deployment Considerations

Reporting Services is not specifically designed for Internet-facing scenarios. This is, partially, because the default authentication mechanism of Reporting Services is Windows integrated security. For security reasons, SQL Server setup does not provide options to deploy SSRS with anonymous access to reports.

Several deployment options are available to an SSRS administrator to make reports accessible over the Internet:

- ▶ Keep only public data in the SSRS catalog and enable Report Server for anonymous access.
- ▶ Deploy SSRS with Windows authentication and leverage Kerberos delegation to authenticate users.
- ▶ Use programmatic options (such as custom security extensions) to authenticate and authorize users.

Internet Deployment Option 1: Enable Report Server for Anonymous Access

This scenario is designed to distribute public information. In this scenario, none of the reports are secured, and all the users would get the same information. When accessing Reporting Services deployed in this fashion, Internet users will not be prompted for login credentials. Best practice for this scenario is to place the SSRS catalog database on the same server with an instance of the Report Server. Because the Report Server has web components, this option means that the SQL Server 2008 instance that hosts catalog data will also be running on the web server and there are no queries that cross boundaries of the web server.

To reduce data exposure in this scenario, the catalog must contain only a limited subset of public data. To further reduce data exposure, reports can be configured to be rendered from an execution snapshot; in this latter case, the SSRS catalog would contain only the snapshot data.

NOTE

To configure a report's rendering from a report-execution snapshot, an administrator can use the Report Manager, navigate to a report that needs to be configured, then navigate to the Properties tab, Execution screen, and select the Render This Report from a Report Execution Snapshot option.

Because this scenario does not protect data from unauthorized access, it might only be used when a company intends to publish public data, such as a product catalog. Secure Sockets Layer (SSL) configuration is not required for this scenario.

To provide public data (or snapshots with public data) to the SSRS catalog in this configuration, an administrator can use replication or SQL Server Integration Services to “copy” public data (or snapshots) from an internal data source to the SSRS catalog placed on a web server.

Internet Deployment Option 2: Deploy Report Server with Windows Authentication

This scenario leverages a default authentication mechanism of SSRS and uses a corresponding security extension.

In this scenario

1. A company would have a domain associated with web-facing servers and use Kerberos delegation to validate a user by interacting with a corporate domain inside the firewall.
2. Customers can configure Reporting Services virtual directories with either Windows integrated or basic authentication.

3. When accessing Reporting Services deployed in this fashion, Internet users are prompted for credentials. After users are validated, they have the level of access to a report corresponding to their credentials.

If this option is chosen, an administrator must configure SSL for proper security, especially for basic authentication.

Internet Deployment Option 3: Use the Programmatic Approach

Situations in which a programmatic approach can be used include the following:

- ▶ Users do not have Windows accounts.
- ▶ User IDs and passwords are stored in a third-party security provider, which, in turn, is used for user authentication.
- ▶ Single sign-on technology (such as Microsoft Passport) is used in place of Windows authentication.

To programmatically handle security, a company can develop a custom security extension, handle security within a .NET application, or use the new ReportViewer control.

NOTE

Remember that security breaches can have far-reaching financial consequences for a business. Therefore, use custom security solutions with caution, especially when a reporting solution is exposed on the Internet.

This book discusses some aspects of security extensions in Chapter 29, “Extending Reporting Services.” An example of a security extension is provided with SQL Server 2008.

On a high level, to handle security within an application, a developer could

- ▶ Authenticate a user in the code by either collaborating authentication processing with a third-party security provider or perhaps simply comparing the user’s identifier and password to the values stored in a database.
- ▶ After the user has been successfully authenticated, the code would either query a third-party security provider or a database for the user’s security access options.
- ▶ The code needs to control access to a report, based on the user’s security access options.

You have several options to control a user’s access to a report. Depending on the need of the reporting application, a code can impersonate a Windows user who mapped to the SSRS Content Manager role (an administrative access). In turn, the code itself would control which reports can be accessed by a user.

Alternatively, depending on the actions that the code must take, the code may impersonate different Windows users who have finer granularity of permissions. In this case, there could be a Windows user who has access to just a single report.

After a user is impersonated, the code can, for example, use the function `Render` to access the report's data stream or use the `ReportViewer` control.

The `ReportViewer` control can process remote server and local reports. When the `ReportViewer` control processes local reports, it does it internally and does not need access to a Report Server.

Most data sources (like SQL Server) that a `ReportViewer` control uses require user identification and a password to access data. In this case, an application can collect, for example, a user's SQL Server credentials and pass those credentials to a data source, thereby restricting the user's access to data.

Enabling a Report Manager for Internet Access

As previously stated, Report Manager was never specifically designed to be an Internet-facing application. But in case it is, a few tips can help make it more secure when exposed to the Internet. Figure 5.3 shows a possible Internet deployment scenario.

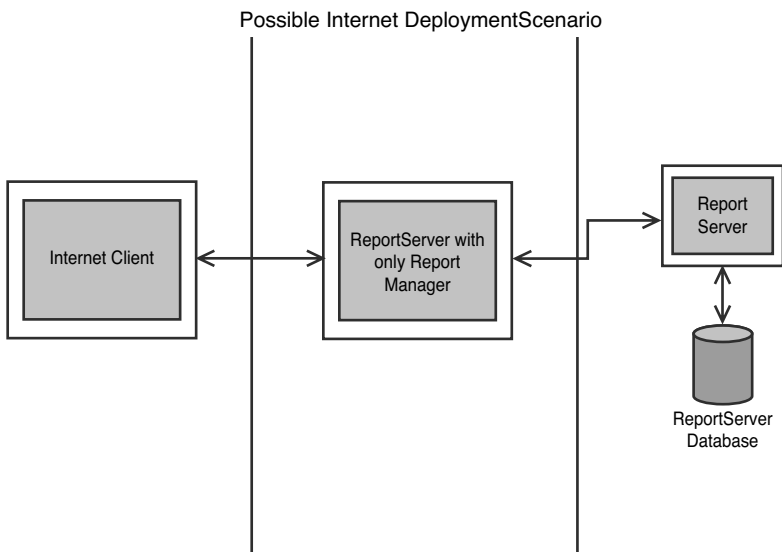


FIGURE 5.3 Internet deployment scenario.

The first of these is to see whether you can run Report Manager on its own server, separate from the Report Server web service, scheduling and delivery processor, and the database server. The key is to remember that SSRS 2008 consolidates all these services into a single Windows service. It is possible to turn off every feature of SSRS except for Report Manager and add the server to a scale-out deployment. This way, the server with Report Manager reaches out to another machine to render and process reports.

Another thing to consider is security. First, build a custom security extension that uses Forms authentication or another kind of technology. After authenticating your users,

minimize their permissions on the Report Server. Two roles are required for viewing reports: Browser and System User.

In addition, minimize the footprint of the exposed server. Make sure Report Manager uses another Report Server to process reports by setting the `ReportServerURL` and `ReportServerVirtualDirectory` setting in the `RSReportServer.config` file. Also turn off any features you are not using. This may include My Reports, client-side printing, Report Builder, subscriptions, and so on.

If all of this fails, and you still end up running Report Manager on the same computer as the Report Server, go ahead and disable the `defaultProxy`. By default, this should be set to false, but go ahead and verify it. An example is shown here:

```
<configuration>
...
<system.net>
    <defaultProxy enabled="false" />
  </system.net>
...
</configuration>
```

Minimum Hardware Requirements

Table 5.3 outlines hardware requirements for SQL Server 2008 installations.

TABLE 5.3 Minimum Hardware Requirements

Hardware	Minimum Requirements 32-Bit	Minimum Requirements x64	Minimum Requirements IA64
CPU	Pentium III-compatible processor or faster. 1GHz minimum. Recommended 2GHz or faster.	Any Intel EMT64 or AMD x64 chip. Minimum 1.4GHz. Recommended 2GHz or faster.	Itanium processor. Recommended 1GHz or faster.
Memory (RAM)	512MB minimum, 2GB or more recommended. Report Server will use a maximum of 3GB (with /3GB switch in <code>boot.ini</code>).	512MB minimum, 2GB or more recommended. Maximum is the OS-specified maximum.	512MB minimum, 2GB or more recommended. Maximum is the OS-specified maximum.
Hard disk space	Total will vary depending on selected components. See Table 5.2.	Total will vary depending on selected components. See Table 5.2.	Total will vary depending on selected components. See Table 5.2.

TABLE 5.3 Continued

Hardware	Minimum Requirements 32-Bit	Minimum Requirements x64	Minimum Requirements IA64
Monitor	VGA or higher resolution. 1024x768 recommended for SQL Server graphical tools.	VGA or higher resolution. 1024x768 recommended for SQL Server graphical tools.	VGA or higher resolution. 1024x768 recommended for SQL Server graphical tools.
Pointing device	Microsoft mouse or compatible pointing device.	Microsoft mouse or compatible pointing device.	Microsoft mouse or compatible pointing device.
CD/DVD-ROM	CD or DVD drive as needed for given installation media.	CD or DVD drive as needed for given installation media.	CD or DVD Drive as needed for given installation media.

The following is the terminology used in relation to the 64-bit platform:

- ▶ IA64 refers to Itanium-compatible hardware architecture. This architecture can run IA64 software and 32-bit software using the Windows-On-Windows (WOW64) software emulator. The Itanium CPU cannot natively run 32-bit x86-compatible instructions and uses instruction emulation as a part of WOW64 processing.
- ▶ x64 refers to Extended Memory Technology support-compatible architecture and includes systems based on Opteron, Athlon 64, Intel Xeon EM64T, and Intel Pentium EM64T. x64 architecture can run classic 32-bit x86-compatible instructions natively on the CPU. One of the advantages of this architecture is an ability to support both 32- and 64-bit code. To ease an adoption of the 64-bit platform and optimize a hardware purchase, some companies might first deploy a 32-bit operating system and software on x64 hardware and then upgrade to 64-bit software on the same hardware requirements.

NOTE

System Configuration Check blocks setup from running if the CPU type (Pentium III or higher) requirement is not met. Setup issues a warning, but allows you to proceed, if the CPU speed or minimum memory requirement is not met.

Software Requirements

We recommend installing Reporting Services on Windows 2008. Although Windows 2003 SP2 is a fully supported platform, Windows 2008 reflects the latest technological advances, including enhanced coverage in the areas of security and high availability.

Windows Server 2008 also provides the Hyper-V virtualization systems. SQL Server 2008 and all of its components, including SSRS, are supported in virtual environments created using Hyper-V, provided of course sufficient CPU and RAM resources are allocated to the virtual machine and that the virtual machine runs an operating system supported by SSRS.

Tables 5.4, 5.5, and 5.6 list operating system requirements and additional software requirements for installation of Reporting Services on 32- and 64-bit platforms.

TABLE 5.4 Operating Systems That Can Run 32-Bit Versions of Report Server

	Enterprise Edition	Enterprise Evaluation Edition	Developer Edition	Standard Edition	Workgroup Edition
Windows XP Professional SP2	No	Yes	Yes	Yes	Yes
Windows XP SP2 Media Center Edition	No	Yes	Yes	Yes	Yes
Windows Vista Ultimate	No	Yes	Yes	Yes	Yes
Windows Vista Business	No	Yes	Yes	Yes	Yes
Windows Vista Enterprise	No	Yes	Yes	Yes	Yes
Windows Vista Home Premium	No	Yes	Yes	No	No
Windows 2003 SP2 Standard	Yes	Yes	Yes	Yes	Yes
Windows 2003 SP2 Enterprise	Yes	Yes	Yes	Yes	Yes
Windows 2003 SP2 Data Center	Yes	Yes	Yes	Yes	Yes
Windows 2008 Standard	Yes	Yes	Yes	Yes	Yes
Windows 2008 Enterprise	Yes	Yes	Yes	Yes	Yes
Windows 2008 Data Center	Yes	Yes	Yes	Yes	Yes

NOTE

Systems that are not explicitly listed in Table 5.4 are not supported by Reporting Services. For example, Reporting Services 32-bit is not supported on Windows 2003 64-bit Itanium.

For situations with heavy memory or I/O requirements, such as heavy graphics and PDF rendering, customers can benefit from deploying SSRS on a 64-bit platform. Table 5.5 outlines SSRS support on a 64-bit platform.

TABLE 5.5 Operating System Requirements, 64-Bit

	Enterprise x64	Standard x64	Workgroup x64	Web x64	Express x64
Windows XP Pro x64	No	Yes	Yes	Yes	No
Windows Server 2003 Standard x64	Yes	Yes	Yes	Yes	Yes
Windows Server 2003 Data Center x64	Yes	Yes	Yes	Yes	Yes
Windows Server 2003 Enterprise x64	Yes	Yes	Yes	Yes	Yes
Windows Vista x64 Ultimate	No	Yes	Yes	Yes	Yes
Windows Vista x64 Home Premium	No	No	Yes	No	Yes
Windows Vista x64 Home Basic	No	No	Yes	No	Yes
Windows Vista x64 Enterprise	No	Yes	Yes	Yes	Yes
Windows Vista x64 Business	No	Yes	Yes	Yes	Yes
Windows Server 2008 Standard x64	Yes	Yes	Yes	Yes	Yes
Windows Server 2008 Data Center x64	Yes	Yes	Yes	Yes	Yes
Windows Server 2008 Enterprise x64	Yes	Yes	Yes	Yes	Yes

The following operating systems are supported by SQL Server Enterprise/Developer Edition IA64:

- ▶ Windows Server 2008 64-bit Itanium
- ▶ Windows Server 2003 SP2 64-bit Itanium Data Center
- ▶ Windows Server 2003 SP2 64-bit Itanium Enterprise

Note that with any 64-bit operating system, management tools may be supported in WOW64. WOW64 allows native 32-bit code to execute natively on non-32-bit systems.

NOTE

Development tools such as Business Intelligence Development Studio (BIDS) are neither installed nor supported on the IA64 platform. For IA64 deployments, use development tools installed on a separate 32-bit or x64 workstation.

Table 5.6 outlines additional software requirements for both 32- and 64-bit platforms and optional software that can be installed to benefit Reporting Services.

TABLE 5.6 Additional Software Requirements, 32- and 64-Bit

Software	Requirement Notes
.NET Framework	Windows 2003 IA63 requires .NET Framework 2.0 SP1. Every other version of requires the .NET Framework 3.5.
Microsoft Data Access Components (MDAC)	All versions require MDAC 2.8 SP1 or higher.
Windows Installer	All versions require Windows Installer 4.5 or later.

Key Features of SSRS 2008 Editions

At least some components of SSRS are available in almost all editions of SQL Server 2008: Workgroup, Standard, Enterprise, Developer, and Evaluation.

Whether a customer is a large enterprise or a small company, the key features of Reporting Services that are always available include the following:

- ▶ **Manageability:** Reporting Services is easy to deploy and manage. In addition to having a convenient web-based management interface, both deployment and management of Reporting Services can be scripted.
- ▶ **Security:** Reporting Services keeps corporate data secure. Reports and information are not accessible, unless sufficient privilege is granted to a user.
- ▶ **Programmability:** Reporting Services allows developing of a custom functionality that can be embedded in a report, called from a report, or scripted.

- ▶ **Reporting controls and wizard:** Windows and web-based ReportViewer controls are supplied with Visual Studio 2008. Report controls simplify adding reporting functionality to Windows and web-based applications.

Additional features available in the Standard Edition of Reporting Services include the following:

- ▶ **Extensibility:** Reporting Services allows adding new server functionality. RDL is an XML-based language and is designed to be extensible. SSRS also allows for extending data-processing, data-rendering, and data-delivery extensions with your own custom implementations.

Additional features available in the Enterprise Edition of Reporting Services include the following:

- ▶ **Scalability:** Reporting Services Enterprise Edition supports large workloads and high-volume reporting. Support for web farms in Enterprise Edition allows easy scale out, providing an ability to add extra capacity as needed. In addition, Enterprise Edition scales up, supporting more than two CPUs.
- ▶ **Availability:** Web farm support of Reporting Services Enterprise Edition paired with the Reporting Services catalog installed on a SQL Server 2008 cluster enables high-availability reporting solutions.
- ▶ **Data-driven subscriptions:** Reporting Services Enterprise Edition allows customers to dynamically change the recipient list, report parameters, and processing options. In contrast, Standard Subscription, available in Standard Edition of Reporting Services, is for a single predefined user and single predefined parameter set.

To help determine the most appropriate version, refer to Table 5.7 to review key features of SSRS editions.

TABLE 5.7 Key Features by Reporting Services Editions

	Express	Workgroup	Standard	Enterprise
Data sources	Local SQL Server instance only	SQL Server and Analysis Services	Supports all data sources (relational and OLAP)	
Rendering formats	Excel, PDF, Image (RGDI, Print), HTML, Word	Excel, PDF, Image (RGDI, Print), HTML, Word	Supports all output formats	
Management	Report Manager	Supports SQL Server Management Studio and Report Manager		
Caching	No	No	Supported	
History	No	No	Supported	
Delivery	No	No	Supported	
Scheduling	No	No	Supported	

TABLE 5.7 Continued

	Express	Workgroup	Standard	Enterprise
Extensibility	No	No	Can add/remove data sources, renderers, and delivery	
Custom authentication	No	Supported		
Scale-out Report Servers	No	No	No	Supported
Subscriptions	No	No	Supported	
Data-driven subscriptions	No	No	No	Supported
Role-based security	Cannot modify roles	Cannot modify roles	Can add roles	
Report Builder	No	Supported		
Report models	No	Supported		
Model-level security	No	Supported		
Infinite clickthrough	No	No	No	Supported

NOTE

Developer and Evaluation editions have the same capabilities as the Enterprise Edition of SSRS. However, the Developer Edition is licensed and supported only in the development environment, and the Evaluation Edition expires after 180 days.

Licensing

In a “nutshell,” a server license (for Workgroup, Standard, or Enterprise editions) is required for every operating system environment on which that edition of SQL Server software or any of its components (for example, Reporting Services) is running.

This means that a company does not have to buy a separate license if SSRS is installed with SQL Server 2005 together on a single computer. For scale-out (web farm) deployments, each web server that runs Report Server must have a SQL Server license.

Summary

In this chapter, you learned about various SSRS deployment choices. Deployment choices for SSRS components range from a developer’s workstation, in which all SSRS components are installed on a single computer, to an enterprise high-availability and high-performance multiserver web-farm deployment.

This chapter also discussed SSRS deployment options for Internet access, and examined the hardware and software requirements, licensing, and key features of the various SSRS editions.

The next chapter delves into the SSRS installation process.