

CHAPTER 2

Installing and Configuring Reporting Services

As we discussed in Chapter 1, Reporting Services is dependent on ASP.NET version 1.1 hosted on IIS, and on SQL Server 2000 SP3a Standard Edition and later. Reporting Services won't work, and is not supported, on SQLExpress or MSDE Editions of SQL Server, and there's a different version specifically for Yukon (SQL Server 2005). Sure, you can use SQLExpress, MSDE or a non-SQL Server database as a data source—just not as the Reporting Services Catalog.

In this chapter, we discuss various installation setups you can use to install and configure Reporting Services. For the most part, this process is managed by the Setup.exe installation wizard, so expect to be prompted for a number of configuration options that determine how, where, and whether each segment of the Reporting Services package will be installed. We know that there are a variety of ways to install Reporting Services, so we've tried not only to address the common case, but also provide hints and techniques to be used for some of the more sophisticated installation scenarios. To make this process as painless as possible, we've broken this chapter down into several sections:

- **Installation Pathways and Preparation:** This section discusses things you need to consider for each of the configuration scenarios—typical or not. Here we show additional details and considerations regarding SSL configuration and other security issues, separate IIS and SQL Server systems and instances, and using the command-line installation option.
- **Step by Step: Installing Reporting Services on a Server with IIS:** This section assumes a typical development machine running IIS, SQL Server, and a Visual Studio .NET tool. It walks

through the Setup wizard step by step and discusses how to respond to each dialog.

- **After Installation—Tuning and Reconfiguring:** This section discusses the various configuration files and the command-line tools used to manipulate the files and perform other security and maintenance tasks. If you choose to use Secure Sockets Layer (SSL) (as we recommend) and you also decide to force SSL for all communications with the Report Manager and Report Server, you should revisit the Tuning and Reconfiguring section after the installation wizard completes, which allows you to tune the config files and change the `SecureConnectionLevel`.
- **Testing the Installation:** This section shows step by step how to verify that your Reporting Services installation is ready to produce reports. Testing helps you install the sample reports using Visual Studio .NET and verify that you can successfully deploy and retrieve reports with a browser.
- **Installing Reporting Services on a Web Farm:** This section walks through the process of setting up Reporting Services on a web farm. This configuration uses several linked IIS servers to share the same Report Server database.
- **Removing Reporting Services:** In case you need to uninstall Reporting Services, this section discusses the mechanisms and side effects of the tear-down process.

Installation Pathways and Preparation

Before we get started, we need to mention the most important prerequisite: the website that hosts the Reporting Services ought to support SSL and therefore needs an SSL web server certificate. In Appendixes A and B, we've included a detailed walkthrough and an article written for MSDN that should prove useful to you if you need further guidance in installing an SSL web server certificate and (if necessary) setting up your own certificate services. Be sure to check out the DVD content as well because we've provided plenty of short media clips on different aspects and issues associated with installation and configuration.

There are two major parts of installation: Server components and client components. The client components are needed by developers to design reports, provide some report programming and database samples, as well as provide a suite of command-line tools to administer the Report Server. These client components are not needed for your users to *run* reports—all they need is a web browser (HTML 3.2 and later).

Operating System Choices

Reporting Services can be installed on the recent breeds of Microsoft operating systems as shown below. We categorize these in two groups: “server” and “workstation” operating systems. Server operating systems include Windows Server 2003, Windows Small Business Server 2003, and Windows 2000 Server. Workstation operating systems include Windows XP Professional and Windows 2000 Workstation.

For test and development situations, we feel that workstation operating systems are suitable platforms, and Windows XP Professional can be an ideal platform for a single machine where you can safely install all of the Reporting Services components and expect reasonable performance and up-to-date functionality. However, take note that Windows XP has a default maximum limit of 10 concurrent connections to its IIS server. This is usually adequate when you’re testing as a single user but can prove to be very limiting and frustrating in some development situations. For example, as we’ll see in Chapter 4, you may hit this barrier of 10 connections when you want to test accessing the Report Server and Report Manager concurrently under the context of different users. Consider that each web browser session consumes at least two IIS connections and often as many as four. All too frequently, we’ve seen that connections can hang and block the IIS server, which reports this stoppage by returning a 403.9 error code message (Too many users connected). Sometimes, even after the performance monitor (Perfmon) shows that all connections to IIS have been released, new connections still generate strange errors.

To address this issue, the first and most important thing we recommend for production environments is to use an operating system from the group of server OSs listed above. These server-class operating systems are designed for “serving” and don’t have a limit on

42 Chapter 2 Installing and Configuring Reporting Services

the possible number of connections to IIS. We recommend that you use either Windows Server 2003 or Windows 2000 Server for your production server. However, Windows Server 2003 is more secure.

If you must use Windows XP and want to address 403.9 errors that have blocked the IIS server, you can take either of two approaches, depending whether or not you want a treatment or cure. The “treatment” approach is to restart IIS, either at the command prompt with the `IISRESET` command, through the Services MMC snap-in or through the IIS MMC snap-in. Restarting IIS usually alleviates the problem—albeit temporarily. The “cure” is to delve into the Windows XP IIS metabase and increase the maximum connection limit to no greater than 40 connections. No, you can’t go beyond 40 connections because IIS is hard-coded to explicitly slam you directly back down to 10 if you try to go above 40. In our development situations, we found it frustrating to work with the 10-connection governor for any extended period of time, but once raised to 40 in normal development scenarios we found it adequate to resolve IIS 403.9 (Too many users connected) errors. See Appendix D for the steps to take to update the IIS metabase. Microsoft also suggests that you make the IIS connection timeout shorter, and we show you how to do that in Appendix D.

NOTE *Installing Reporting Services on Domain Controllers* When installing Reporting Services on a Windows 2003 server that is also a domain controller, no manual configuration is necessary for Reporting Services to install and run properly. However, on a domain controller on Windows 2000 Server, while Reporting Services installs properly it is not automatically activated. In this case, you’ll need to perform the following steps, either before or after running Setup, in order to properly configure Reporting Services to run on a domain controller:

1. Grant Impersonate Privilege to the `IWAM_<machine>` account. For more information, see the Knowledge Base Article “IWAM Account Is Not Granted the Impersonate Privilege for ASP.NET 1.1 on a Windows 2000 Domain Controller with SP4” (KB 824308).
 2. Remove the `IWAM_<machine>` account from the Guest group. Guest users cannot store or maintain encrypted content. For more information, see the Knowledge Base Article “Roaming Profiles Cannot Create Key Containers” (KB 265357). Then reboot the computer.
-

All on One Machine (Typical Development Scenario)



Guide me!

One approach to installation is to simply execute¹ the Reporting Services Setup.exe without using command-line arguments and install everything on a single machine, accepting all the wizard's defaults. (In this case, you need to do this with Local Administrator privileges.) The Setup.exe installation wizard inspects the local machine and performs an initial check, looking for any “missing” components, some of which the wizard might be prepared to update for you—for example, the .NET Framework, although it will only report other missing dependencies in a dialog when it cannot find a qualifying version of SQL Server 2000. These components are “missing” based on the bold assumption that your intention is to install the Reporting Services, Report Designer add-in utilities, documentation, and sample reports all on the same machine. This is the simplest installation approach, assuming you are targeting a development machine that already has the following Setup:

- A Visual Studio .NET 2003 Development Tool installed (e.g., Visual Basic .NET 2003)
- SQL Server 2000 Developer Edition² or better with Service Pack 3a
- Internet Information Server (IIS) with ASP.NET 1.1³
- An enabled Default website *with an SSL web server certificate installed*

NOTE It's best to make sure the IIS service is running on your system before installing Reporting Services. (IIS is called the “World Wide Web Publishing” service in the Services management console (MMC)). If you already installed Reporting Services (before IIS), you'll need to use a command prompt. Navigate to `C:\Windows\Microsoft.NET\Framework\v1.1.4322` and execute `aspnet_regiis -i` to register ASP.NET in the IIS metabase.

¹ Why do programs start working when you “execute” them, but people stop working?

² The SQLEXPRESS, MSDE or Personal Editions of SQL Server can't host Reporting Services.

³ The RTM readme says Reporting Services Setup requires that version 1.1.4322 of ASP.NET be registered with IIS.

44 Chapter 2 Installing and Configuring Reporting Services

Setting up a machine with these prerequisites⁴ should be straightforward enough. The only thing that might trip you up is the installation of an SSL web server certificate that we casually slipped in there—especially if you’ve not had much exposure to IIS and Certificate Services. Please, don’t install Reporting Services without an SSL certificate. Yes, the wizard will let you uncheck Require an SSL Certificate, but be prepared to tell your manager why you needlessly (and carelessly) exposed your data and potentially your systems to the world. Keep in mind that there are situations in which user credentials (logon name and password) need to be passed to and from the Report Server and Report Manager—even if you’re simply updating data sources with the Report Manager. If you don’t have SSL installed, those credentials can be harvested by evil people. Think about how careful you should be when entering credit card details into a web form. The same degree of care should be applied to any user credentials that you enter into web forms such as the Report Manager. Microsoft gives you all the tools and the Setup wizard defaults to remain secure, so there can be no excuse if you choose to be reckless. Sure, Reporting Services works without SSL, and perhaps in a development environment you might be tempted to go without this degree of protection—but we still don’t recommend this approach. If you’ve not used SSL in production before, it would be a good idea to experiment with it in a development environment first.

It’s possible to separate the installation into several phases and place the Report Designer on machines where Visual Studio .NET 2003 is installed (and licensed), and install the Reporting Services (Report Manager and Report Server) on any machines running IIS. In a web farm environment, you can have several machines hosting the Reporting Services Server components, all accessing the same Report Server database catalog. The present license arrangements (as we understand them) require you to have a SQL Server license (either per seat or per CPU⁵) for each server that hosts the Reporting Services Server components (Report Manager and Report Server),

⁴ The installation wizards (should) take care of other prerequisites such as installing the 1.1 version of the .NET Framework—however, the readme.htm puts a shadow on this assertion.

⁵ Machine or CPU licensing is selected during SQL Server Setup.

⁶ EULA (noun): End-User License Agreement.



I know Bill Vaughn has worn out three pairs of spectacles going over the minutiae of the United States EULA⁶ paragraph 12, rereading and rereading it. Beats counting sheep, doesn't it, Bill?

—Peter

Well, did you notice in paragraph 54 that the EULA grants rights to your first-born child in servitude to Microsoft?

—Bill



Guide me!

whether or not you have SQL Server installed on that machine. In the case of per seat licensing, that means each user that accesses the Reporting Services or the Design Tools will need a CAL (Client Access License). No, we're not licensing experts (or lawyers), and of course licensing is a complex subject, so read your licensing conditions very carefully.

If installation doesn't go smoothly, you may need to use some of the command-line administration tools and delicately futz⁷ with some configuration files. We explain those tools in brief later in this chapter, but we can't cater to every single caveat here, especially because new caveats can be created by Service Packs. For the latest information, take a look at our website (www.SQLReportingServices.NET) and join our community, as we'll be doing our best to help folks out there stay current on evolving Reporting Services technology.

Installing Only the Report Designer Add-In

The Report Designer add-in for Visual Studio .NET is used to interactively design reports (as we discussed in Chapter 1). If you want to install just the Report Designer add-in to a target machine, run the Reporting Services Setup.exe file, but ensure beforehand that you already have a Visual Studio .NET 2003 Development Tool like Visual Basic .NET installed. Don't worry when the wizard reports that ASP.NET, IIS, or the Default website are not installed if you deliberately don't have them on the target machine—they aren't needed to install and use the Report Designer add-in. On the wizard's Feature Selection dialog, just choose that you only want the Report Designer. It's probably a good idea to install the Books Online (BOL) help file as well because it can be hosted in the Visual Studio .NET designer. In Chapter 4, we'll show you how to customize your Visual Studio .NET Help search interface to find topics in the Reporting Services help files.

You may be asked for a 25-character CD key if the wizard can't detect a qualifying version of SQL Server on the target machine—so make sure you have the CD key from your SQL Server handy.

⁷ futz (verb): Manipulate, fix, experiment with, tune, play with.

46 Chapter 2 Installing and Configuring Reporting Services

(If you've mislaid the hard copy of the CD key—you should be able to locate it in the registry of the SQL Server machine under *HKLM\SOFTWARE\Microsoft\Microsoft SQL Server\80\Registration:CD_Key*). We'll first use the Design Tools in Chapter 3 when we show how to use Design wizards, and we really go to town with the Report Designer in Chapter 6. After installation, if you have issues deploying reports, and you're getting messages like: "The underlying connection was closed: Could not establish trust relationship with remote server." chances are you need to deploy using HTTPS and include the same URL embedded in the SSL web server certificate.

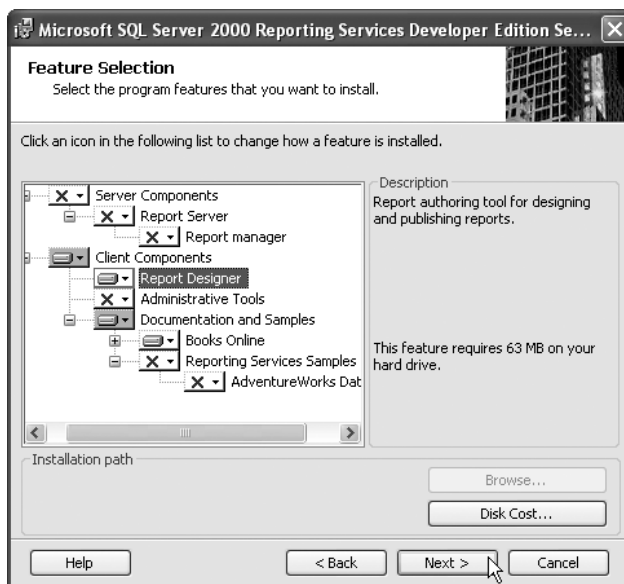


Figure 2.1 Select the Portions of Reporting Services That You Wish to Install

Installing Only the Sample AdventureWorks2000 Database

You can install the Sample AdventureWorks2000 database on a target SQL Server by running Setup on the server and selecting just the AdventureWorks2000 database from the Feature Selection dialog (see Figure 2.1). After the AdventureWorks2000 database is

installed, only administrators will be able to get at the data to produce reports from it, so don't forget to add domain groups to the database and at least assign them to the SQL Server db_datareader role.

Installing Reporting Services on a Server with IIS

In a production environment you probably won't want to install the Sample Databases or the Design Tools onto a production SQL Server or IIS server. However, before you run Setup (the installation wizard), step through this checklist:

1. Decide which IIS server is to host the Reporting Services. Initially, the IIS server must have the Default website enabled.
2. Check that the IIS server has an SSL web server certificate. If `http://<Server>/Postinfo.html` returns a web page, but `https://<Server>/Postinfo.html` does not, you need to install an SSL certificate. For detailed information on installing SSL, see Appendixes A and B and watch the video clips.
3. Decide where the Reporting Services' own catalog database is going to be installed. This needs to be SQL Server 2000 with at least SP3a—not Yukon, MSDE, or Personal Editions. This SQL Server need not be on the same server as the IIS components, but if it is, you economize on the SQL Server licences that you'll need.
4. Do you have sufficient privileges on your database? Make sure you have an account that is a member of the sysadmins role on the SQL Server to use during installation.
5. If you need to run Setup with command-line options, consider our security caveats, and especially consider encrypting any Setup ini file.
6. Launch the installation wizard from a non-network path.

Which Account Is Running the Install Wizard?

During installation of Reporting Services, the bootstrapper installation wizard logs on to the SQL Server used to host the Report Server catalog databases with the wizard's account credentials, using

48 Chapter 2 Installing and Configuring Reporting Services

“Trusted” connection SSPI security. This means the target SQL Server must expose a login account that corresponds to the rights granted to the user running the wizard. As we said earlier, it’s easiest if the wizard is run with Administrator credentials as SQL Server automatically creates a login account for all system administrators. If the domain administrator won’t let you have access to the Administrator account and you choose to use an ordinary Domain User account, it must belong to the System Administrators (sysadmin) SQL Server Security “Server Role” (at least during the install). The wizard will check up on you to ensure that it is and won’t continue if it isn’t.⁸ (If you’re a trainee SQL Server guru, you might think that the Database Creators (dbcreator) SQL Server Security “Server Role” would be sufficient. However, the bootstrapper installation wizard also needs to make a couple of calls to the `sp_addrole` stored procedure, in addition to creating the Reporting Services databases. Accounts that are only members of the Database Creators (dbcreator) role can’t do that.)

Of course, using a trusted connection is by far the most sensible route from a security standpoint, but there may be situations where you have the SQL Server in one Active Directory Domain, the Reporting Services in another Active Directory Domain, and no formal Active Directory Trust between the two Domains. Perhaps you have no Active Directories at all. In this case, it means that you won’t be able to add the Domain User Account under which you are running the installation wizard to the SQL Server System Administrators SQL Server Security “Server Role.” Thus, the wizard won’t be able to create the database catalogs. We told you it would be easier just to steal the Administrator’s password. Don’t worry, you are not pooched—there is a solution. You can instruct the wizard to use a SQL account that belongs to the sysadmin role instead. To do this, launch the Setup installation wizard and provide a few command-line options: `RSSETUPACCOUNT` and `RSSETUPPASSWORD`. We discuss these options in the next section.

⁸ We’ve found that when the wizard catches you, things work best if you cancel the install, make the changes to the account on the SQL Server, and then start the install again.

Security Bulletin 1

If you put sensitive information into an ini file—things like the credentials of a SQL Server sysadmin account, as when setting the `RSSETUPACCOUNT` and `RSSETUPPASSWORD` options, practice safe computing. Setting ACLs on files should be your first line of defense, but you should also be concerned about what happens to those files when they are deleted.

Before you edit and save any values in your copy of the template file, encrypt your copy of the file (right-click it, select Properties, on the General tab click the “Advanced” button, and then select Encrypt contents to secure data). In addition, consider the permissions on this option file. Why? Well, if you don’t follow our advice you might as well just write the SA password on a Post-it note and stick it on the monitor for all to see. Consider that when you delete an unencrypted ini file, any credentials or other information it contained could be easily “harvested” from the disk. A wealth of utilities floating around the Internet (for free) are for doing just that, as anyone watching one of the crime shows on TV would know.

Security Bulletin 2

We also hope that if you are going to be using a SQL Server account for the Setup (i.e., setting the `RSSETUPACCOUNT` and `RSSETUPPASSWORD` options) that you have assigned an SSL certificate to the SQL Server and have configured the SQL Server to force encryption. If you haven’t, you should be aware that SQL Authentication, which is what will by default be used over an unencrypted connection, is fairly insecure. One final thought for the paranoid:

After the installation, you can always change the Password for the SQL Server account that you used during installation—just don’t rely on this approach. It only takes someone listening to grab the credentials while on the wire and immediately create their own backdoor sysadmin account. Yes, folks, we’ve seen this kind of thing happen in some environments when the door was only open for a few seconds. (Sleep well tonight.)

Installing Reporting Services on Web Farms

It’s possible to scale out and configure a number of IIS servers to all use the same Report Server database. Such a configuration is called a *web farm*. Web farm configurations are only available to the Developer/Enterprise/Evaluation versions of SQL Server, but there

50 Chapter 2 Installing and Configuring Reporting Services

are licensing restrictions on use of the Developer or Evaluation versions in production. If during the installation you instruct the wizard to use a pre-existing Report Server database, the wizard asks if you want to setup a web farm and leads you through the installation steps. There may be occasions when you want to take a stand-alone Reporting Services installation and create or join it to an existing web farm. With some configuration file editing and some command-line utilities (`rsconfig.exe`, `rsactivate.exe`, and possibly `rskeymgmt.exe`), you'll be able to get it to work. We briefly discuss some these utilities to provide an overview of what they do in "Installing Reporting Services on a Web Farm," later in this chapter.

Licensing Your Reporting Services Installation

As with the installation of the Report Designer, if the wizard cannot find a qualifying⁹ installed SQL Server 2000, it asks you for the 25-character product key of a qualifying version. To comply with the license agreement, you need a SQL Server 2000 server license for every machine on which the Reporting Services *Server* components are installed. If you decide to split the installation so that the Reporting Services SQL Server Catalog is on one machine and the IIS Report Manager and Report Server are on another machine, you'll need two SQL Server 2000 server licenses—but only one if you run them on the same box. As we said earlier, we're not licensing experts, and licensing arrangements change and are different from locale to locale, so please check your license arrangements carefully. For development purposes only, our solution is that we subscribe to MSDN Universal, which includes SQL Server Developer Edition,¹⁰ which has sufficient rights and licenses for our development machines—but the MSDN Universal is not licensed for any production purposes.

⁹ Qualifying versions do not include SQL Server Express, MSDE or Personal Editions. Yes, these versions can be reported from and used as data sources, but they are not licensed to host the Reporting Services Catalogs, and the installation wizard is hard-coded not to install onto them.

¹⁰ SQL Server Developer Edition can be purchased for \$49.

Command-Line Options for the Installation Wizard

It's possible to run the installation wizard from a Command prompt window and provide some or all of the options to the wizard—it's even possible to do a completely silent install. The command-line options are typically supplied as parameters to Setup.exe or through an ini file. You'll find a template ini file in your Reporting Services distribution media. The template is well documented and explains all of the options. It's convenient (and a good idea) to copy this template and edit the values you need. Next, launch the Setup.exe wizard from a command prompt with `setup.exe /settings myoptions.ini`—assuming myoptions.ini is your file based on the template.ini.

Preparing Your System to Run the Setup Wizard

If you've already received an SSL web server certificate from a public Certificate Authority or from an Enterprise Certificate Services Certificate Authority within your Active Directory Domain and created and installed an SSL web server certificate for the web server hosting Reporting Services, you'll have little to do but answer a few simple questions in the following Reporting Services Setup dialogs. If not, then you might need to take a couple of side trips to make sure this SSL infrastructure is installed. We've provided a detailed explanation of the steps you'll need to take to enable SSL security for your website in Appendix A, "Using SSL to Protect Your Data," and Appendix B, "Using Secure Sockets Layer for Reporting Services." Better yet, watch the Guide me! narrated screen capture demonstration that shows how to do this.

Step by Step: Installing Reporting Services on a Server with IIS

With SSL in place, you're ready to proceed with the Reporting Services Setup. As we discussed earlier, the default prerequisites for the Report Services installation specify the target machine has IIS, ASP.NET 1.1, and a Default website (with an SSL web server certificate). So, we are working with either Windows 2000 Server, Windows Server 2003 configured as an Application Server, or

52 Chapter 2 Installing and Configuring Reporting Services

Windows XP with IIS enabled. You'll also need connection credentials to an instance of SQL Server 2000, which has at least Service Pack 3a applied. Make sure you setup a properly permissioned login ID. To do all of this, you're going to need to use the sysadmin role¹¹ as described below, but only for the installation. This means that if you don't have sufficient permission, you need to get your SQL Server SA to do this for you.

If you are installing Reporting Services on a machine that:

- Belongs to an Active Directory Domain
- The SQL Server hosting the Reporting Services catalog database is in the same Domain
- You haven't messed around with your SQL Server security settings
- You are installing under an account that has Admin privileges

If all of the above are true, the major roadblocks to a successful installation have been addressed. However, if you stepped off that line, encountered some setup issues, or want to perform unattended installs, you might have to return to the earlier section that discusses the issues you're likely to encounter.

Stepping Through the Setup Wizard

The Component Update dialog appears when you first start the Reporting Services Setup wizard. It automatically checks to make sure that the "Microsoft SQL Server Reporting Services Setup Support Files" and other dependencies are installed. This dialog also runs when you restart Setup to uninstall or reconfigure Reporting Services. According to `readme_en.htm`, you'll have to have ASP.NET 1.1.4322 registered with IIS. You'll also need to ensure that MDAC 2.6 or later has been installed, which should not be a problem as this version of MDAC (or later) is included with all of the recent versions of Windows and is installed with Visual Studio .NET.

This dialog is followed by a Registration Information dialog asking you to personalize your installation. There's no clear use for this information, but we expect it's salted away in the registry somewhere.

¹¹ Use the SA login or a login that's a member of the sysadmin fixed role.

Next, you'll see a Feature Selection dialog, which permits you to select how much of Reporting Services should be installed and where to install it. This dialog determines whether the Server components, the client components, or both should be installed. If some prerequisites are not met, the features that depend on them are missing from the component tree. For example, no IIS means no Server features, and no Visual Studio means no Report Designer. If you're installing Reporting Services on a development system, we strongly suggest including the Reporting Services Samples and AdventureWorks database, as we will be using these to illustrate how Reporting Services works. You don't have to install Reporting Services onto the machine hosting the SQL Server Report Server database catalog (but the machine will need a SQL Server license).

Remember, all you need for Reporting Services is a machine running IIS server, ASP.NET 1.1, and a Default website with an SSL web server certificate. This system can be Windows 2000 Server or a Windows Server 2003 configured as an Application Server. It can even be a Windows XP system if you install IIS. The Reporting Services Setup program will check to make sure most of these requirements are in place when first started—all except the SSL certificate. Because you might not have an SSL web server certificate, we'd better tell you what you need to know about Secure Sockets Layer (SSL).

Reporting Services Windows and Web Service Account Installation Options

The next dialog (shown in Figure 2.2) specifies the accounts under which the ReportServer Services will run. This dialog can sometimes seem a little tautologically¹² confusing, and it's easy to lose track of which service uses which account, so let's take a deep breath and make a start.

¹² tau-tol-o-gy (noun): A redundant repetition of a meaning in a sentence or idea using different words.

54 Chapter 2 Installing and Configuring Reporting Services



Figure 2.2 Selecting the Account for the ReportServer Windows Service

- **The ReportServer Windows service**, a Windows service, can execute under a number of accounts, depending on which operating system it's running on. On Windows 2000 and Windows XP, you can configure this to be either a Domain Account or the Local System Account (NT AUTHORITY\SYSTEM). On Windows Server 2003, there is an additional choice of running this with the Network Service Account (NT AUTHORITY\NETWORK SERVICE). We recommend that you use the Local System Account on Windows 2000 and Windows XP and that you choose Network Service Account on Windows Server 2003. This choice will make more sense in a minute.
- **The Report Server web service** is presently an ASP.NET program, so by default it's run under the account configured for the ASP.NET Worker Process (aspnet_wp.exe). On Windows XP and Windows 2000, this (by default) is a local user account

on the machine called ASPNET. On Windows Server 2003, the ASP.NET Worker Process is run under the Network Service Account (NT AUTHORITY\NETWORK SERVICE). If you have changed this account in machine.config, chances are Setup will not be able to run correctly.

There is also an account that we need to concern ourselves with—this is the account whose credentials the Report Server web service and the ReportServer Windows service both use to access the Report Server database. If you followed our recommendation and are running the Windows Report Service under a Local System or Network Service, you have several choices for the credentials for the Report Server web service:

1. The same account that the Windows Report Service is running under
2. A Domain User Account
3. A SQL Login account

If you are running Windows 2000 but didn't accept the recommendation, and you decided you knew better and used a Domain Account for the ReportServer Windows service, your only option is to access the report database using SQL Login credentials or use the same Domain Account that you specified for the Windows service. In addition, your reports will not be able to use prompted or stored Windows credentials to access remote data sources; they would have to use SQL credentials. If you use a SQL account, Setup creates it if it does not exist and assigns the correct SQL permissions to it. It also assigns the required SQL permissions to any account you use. However, using a SQL Login can be considered a security vulnerability unless you also have enabled Encryption on your SQL Server. Why? We knew you were going to ask that (again). This security vulnerability is mostly due to the fact that SQL Login credentials are not as secure as trusted connections or Domain credentials, which have more security features built in—things like making sure passwords are unique and are replaced regularly, and SQL Authentication is much easier to crack than Windows Domain Credentials from a hacking point of view. Another factor here is the cavalier attitude some folks

have when passing around and using the SQL Server SA account password. Ah! You mean, why can't you use a Domain Account? Well, the answer is that it is a limitation of Windows 2000, and also the infrastructure of Reporting Services—so, it is because it is...unfortunately.

If you are on Windows Server 2003, then party on, but use the Network Service Account (NT AUTHORITY\NETWORK SERVICE) rather than the Local System Account (NT AUTHORITY\SYSTEM) for the Reporting Services Windows service, as it has fewer privileges but is sufficient for Reporting Services and is therefore somewhat more secure.

Auto-Starting the Reporting Services Service

At the bottom of the Service Account dialog, you'll find a checkbox that specifies whether or not to auto-start the service. If you uncheck this box, the Reporting Services service will have to be started on first use. For development systems, this might be a good idea to reduce the amount of overhead the service consumes.

Choosing the Reporting Services Virtual Directories

The next dialog you'll encounter in the Reporting Services Setup wizard asks you to specify the virtual directories on which Reporting Services and Report Manager are accessible (as shown in Figure 2.3). Throughout the book and DVD examples, we'll use the defaults of "ReportServer" for the Reporting Server virtual directory and "Reports" for the Report Manager virtual directory.

You may have already gathered that the Report Manager and the Report Server are installed in virtual directories on the Default website of an IIS server. By default, Reporting Services Setup assumes that the website that you want to install on has a web server certificate issued to it and supports SSL. This is great if your website supports SSL, but it may possibly send you off in a diversionary spin if it doesn't, and on a substantial time-consuming learning curve if IIS configuration minutiae are outside of your hitherto core experience. This is especially true if you don't have an experienced IIS administrator on hand to bully or cajole into setting up the SSL for you.

Unless you uncheck the Use SSL checkbox in the installation wizard (which is shown checked in Figure 2.3), you won't be able to install with the wizard—that is, unless you have SSL already installed. Take heart, because Appendix A will give you a *boost* with instructions and a Guide me! video, and if that still isn't enough to get you going we offer correspondence and onsite consultancy services for a modest fee.

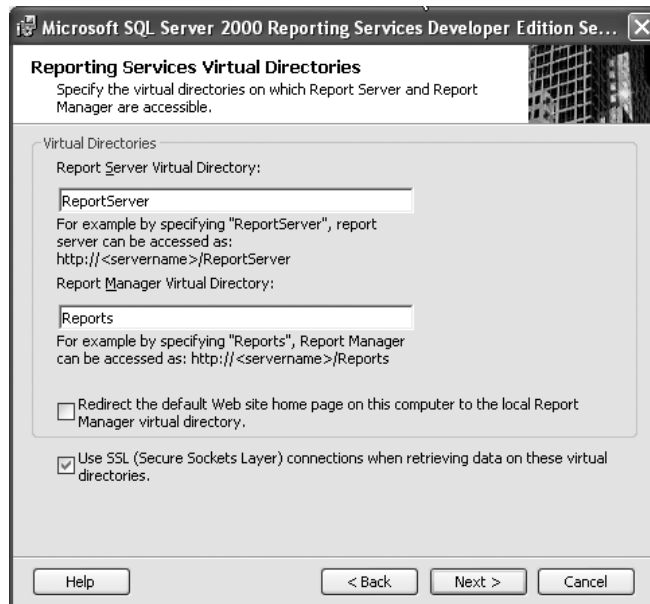


Figure 2.3 If You Don't Have SSL Installed, You Won't Get Past This Installation Dialog Without Removing the Check from "Use SSL"

Why Is Secure Sockets Layer (SSL) Important?

SSL provides the ability for an IIS server to encrypt network traffic, and makes it possible for a browser to interact with a website (securely) using *HTTPS*. Web sites *should* be using HTTPS and SSL when you provide your credit card details. The theory is that if the network traffic is encrypted it makes it more difficult for anyone who

intercepts that traffic to decrypt it—assuming that they don't possess the keys. The cost of having SSL (apart from the certificate mentioned in Appendixes A and B) is a slight dent in performance.

In the Reporting Services context, one important role for SSL is effective encryption of any user credentials that might need to be passed to Reporting Services to enable it to gain access to a managed report's underlying data. While there may be other approaches and better practices, which we'll talk about elsewhere, it is possible to pass these credentials in the URL or in an HTTP POST. If you don't have SSL and therefore are not using HTTPS, those credentials will traverse the network in plaintext and be a trivial exercise for evil folks to harvest.

As we said in Chapter 1, you'll discover that in order to access many of the more important SOAP interface web methods, you'll have to have SSL enabled on the IIS server. If you choose to write your own custom applications using the SOAP methods, you'll find you have secure and complete access to the whole SOAP interface, but only if you have SSL enabled on the Reporting Services website.

Appendix A walks you through the process of setting up SSL on your website and provides a great deal of detailed information on the minutiae you might need to consider if you haven't done this before. Appendix B is an MSDE article we wrote on installing SSL and a certificate service that might also prove beneficial. We're going to assume from this point forward that you've created and installed an SSL certificate for your website.

Selecting the Report Server Database and Credentials

The next dialog exposed by the Reporting Services Setup wizard (as shown in Figure 2.4) asks you to specify the SQL Server to host the Reporting Services database. Remember to address the correct instance using the <instance>\<server name> notation. Next, supply (or accept) the name of the Reporting Services database—the default is "ReportServer." If the wizard finds this database on the Report Server, it stops and demands that you change the name or remove the existing database. Otherwise, it creates a new ReportServer (2.31 MB) and a ReportServerTempDB (1.24 MB)

database. As we'll discuss later, if you uninstall Reporting Services, the database is not removed (among other things) so you'll have to clean up your system after the uninstall wizard runs.



Figure 2.4 Select the Report Server Database and the Domain Account That Accesses It

Which Accounts—Best Practice Summary

Windows 2000 and Windows XP

- Run the Windows service under NT AUTHORITY\SYSTEM.
- Use a Domain Account for the services to access the SQL Reporting Services database.

Windows Server 2003

- Run the Windows service under NT AUTHORITY\NETWORK SERVICE.
- Use a Domain Account for the services to access the SQL Reporting Services database.

SMTP Mail Configuration

The next dialog you'll encounter (shown in Figure 2.5) asks you to specify how Reporting Services should contact you via e-mail. It's important to establish these linkages so that Reporting Services can pass reports via mail and notify you if there are problems with the server. This section discusses how to specify the SMTP and "From" address as well as how to edit elements in the `RSReportServer.config` file to ensure mail is delivered as expected.

As discussed in Chapter 1, Reporting Services supports a number of delivery extensions, including SMTP,¹³ that can be used to send reports via e-mail. During the Setup process, you'll be asked to configure two parts of the SMTP configuration: the SMTP server and the e-mail address that the Report Server will use as a "From" address. Note that the SMTP server address is either an IP Address or a resolvable network name to an SMTP service.

If you want to enhance your *From* address, you can provide a free-text name and specify the e-mail address in < > brackets, as shown in the Setup dialog in Figure 2.5. We've provided "smtp.boost.net" as our SMTP server address. We also used "Boost Report Server <Reports@boost.net>" as the From address. However, we suggest that you *don't* use these addresses for your server as the boost mail server won't guarantee that you'll get your e-mail—use your own SMTP server address.

When e-mail arrives in the destination mailbox, it shows (in this case) as being from "Boost Report Server" and masks the e-mail address (as shown in Figure 2.6). If you have a web farm installation, you could use different names with the same e-mail address in the From address—but if that's the case, be sure to see "Installing Reporting Services on a Web Server" later in this chapter.

No, you don't really need to create an Exchange mailbox for the From address—that is, unless you want to monitor the "non-delivery" reports sent back to this e-mail address, or read the replies your users might send the Report Server.

¹³ SMTP: Simple Mail Transfer Protocol, a protocol for servers to send electronic mail among themselves.

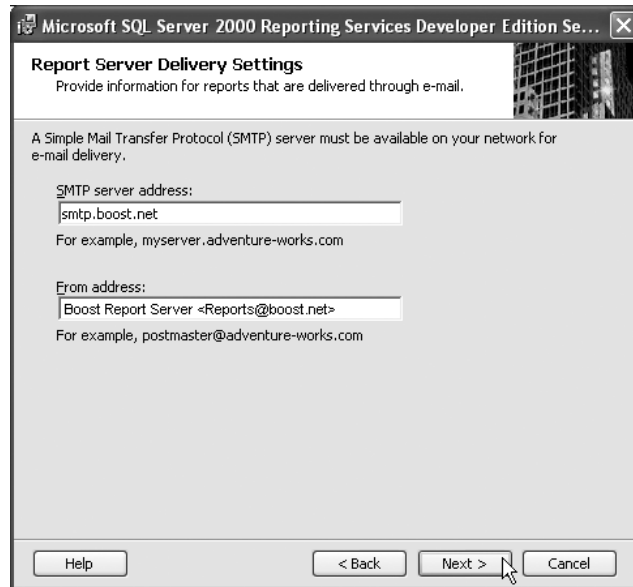


Figure 2.5 SMTP Setup Dialog

	From	Subject	Sent	Size
☑	Boost Report Server	Product Catalog was executed at 24/12/2003 01:20:06	Wed 24/12/2003 01:20	2 KB
☑	Boost Report Server	Report2 was executed at 24/12/2003 01:20:06	Wed 24/12/2003 01:20	4 KB
☑	Boost Report Server	Product Catalog was executed at 24/12/2003 01:19:13	Wed 24/12/2003 01:19	2 KB

Figure 2.6 Reports in an E-Mail Inbox with Enhanced “From” Names

Once the SMTP installation phase is over, the only way to change the SMTP server and the e-mail address is to edit elements in the `RSReportServer.config`¹⁴ file under the Report Server Email extension node. Shown in Listing 2.1, on the next page, is the raw default configuration resulting from the SMTP installation wizard. If you want to enhance the e-mail name after having already run the wizard, notice that “<” is represented as `<`; and “>” as `>`, just like in HTML!

¹⁴ You’ll find the `RSReportServer.config` file in `\Program Files\Microsoft SQL Server\MSSQL\Reporting Services\ReportServer`.

62 Chapter 2 Installing and Configuring Reporting Services

Listing 2.1 The RSReportServer.config File

```
<Extension Name="Report Server Email" Type="Microsoft.ReportingServices.
EmailDeliveryProvider.EmailProvider,ReportingServicesEmailDeliveryProvider">
  <MaxRetries>3</MaxRetries>
  <SecondsBeforeRetry>900</SecondsBeforeRetry>
  <Configuration>
    <RSEmailDPConfiguration>
      <SMTPServer>smtp.boost.net</SMTPServer>
      <SMTPServerPort></SMTPServerPort>
      <SMTPAccountName></SMTPAccountName>
      <SMTPConnectionTimeout></SMTPConnectionTimeout>
      <SMTPServerPickupDirectory></SMTPServerPickupDirectory>
      <SMTPUseSSL></SMTPUseSSL>
      <SendUsing></SendUsing>
      <SMTPAuthenticate></SMTPAuthenticate>
      <From>Boost Report Server<&lt;Reports@boost.net&gt;&gt;</From>
      <EmbeddedRenderFormats>
        <RenderingExtension>MHTML</RenderingExtension>
      </EmbeddedRenderFormats>
      <PrivilegedUserRenderFormats></PrivilegedUserRenderFormats>
      <ExcludedRenderFormats>
        <RenderingExtension>HTMLLOWC</RenderingExtension>
        <RenderingExtension>NULL</RenderingExtension>
      </ExcludedRenderFormats>
      <SendEmailToUserAlias>True</SendEmailToUserAlias>
      <DefaultHostName></DefaultHostName>
      <PermittedHosts></PermittedHosts>
    </RSEmailDPConfiguration>
  </Configuration>
</Extension>
```

We don't actually like the default installation for e-mail because after <MaxRetries> at set intervals the mail can fail. We prefer using an alternative that makes use of the SMTP Windows service and its pick-up directory, as this can be more robust and takes the pressure off Reporting Services to make delivery. We tell you how you can configure that in Appendix C, "Configuring SMTP."

Report Server Samples Setup

If you checked the samples option in the Feature Selection dialog, the Setup wizard asks you where the 72.63 MB AdventureWorks2000 database should be installed. Normally, this is installed on the Report Server host SQL Server, but it can be sited anywhere there's room and connectivity. As with the earlier steps, the user running Setup must have permission to create databases on the specified SQL Server instance. If this database exists on the target instance, you'll have to delete it before proceeding or point to another target server.

Licensing Mode

The next dialog should be familiar to anyone who has installed SQL Server before—it asks you to specify how you intend to license Reporting Services.

Ready to Install

Up to this point, the wizard has done nothing as far as actually installing anything on the target servers. When you click Install on the Ready to Install dialog, the wizard completes the Setup process. Once the installation wizard completes, you might want to take a few more steps to further improve the security of your Reporting Services installation. If you installed and accepted SSL (by default), any credentials passed back and forth between the server and the browser are protected by SSL and are secure. However, if the reports themselves are of a sensitive nature, you might want to force all communications to be over SSL, including managed report delivery. To achieve this degree of security, there are a number of additional steps you need to take:

- The RSWebApplication.config file <ReportServerURL> element contains the correct URL for your Report Server.
- The RSReportServer.config file has the SecureConnectionLevel set to 3.
- The RSReportServer.config file has the correct URL for your Report Server.

The location of these files and details on how to change them is located in the section “After Installation—Tuning and Reconfiguring,” later in this chapter.

As we’ve said, Reporting Services installation can be a fairly complex operation (especially if you want to protect your data), so there are a few things that could go wrong. We have seen many of these issues as we’ve worked extensively with the Setup wizard and configuration files, so we should be able to help you build a safe server. To help validate your server installation, we provide an outline of steps to verify that you’re ready to start creating, deploying, and publishing reports. See “Testing the Installation” near the end of this chapter.

Installing Reporting Services in a Non-Default Website

The ordinary installation wizard installs Reporting Services as two virtual directories to the Default website. If you’re installing to a development system on Windows XP, you don’t have anything other than the Default website anyway. However, if you’re installing to a server operating system such as Windows 2000 Server or Windows Server 2003, you can manually create a completely separate website on the server just for Reporting Services. This certainly makes it easier to manage, as it is separated from the Default website, and it enables you to do things like stop and start such a website independently of any others your server may be hosting. It also permits you to install an SSL certificate specifically for the Reporting Services website.

If you are using Windows 2000 Server, you’ll have to perform a normal install to the Default website, and subsequently create a completely new and empty website (not virtual directory), sign it with an SSL certificate (not mandatory but it is if you want to be one of our friends), and then create two virtual directories: one called “Reports” and the other “ReportServer.” At this point, you can compare all the properties of the Reports and ReportServer virtual directories and copy them over—one property at a time—to the new

directories. You haven't quite finished by this stage as you must also install the ASP.NET 1.1 client-side scripts to your website by executing `aspnet_regiis.exe` from the correct .NET framework and with the correct parameter. At the command prompt, navigate to `%windir%\Microsoft.NET\Framework\v1.1.4322` and execute `aspnet_regiis -c`. Once all of this is done, you can now remove the Reports and ReportServer virtual directories on the Default website. And finally, you'll need to change the URLs in the `RSWebApplication.config` (`<ReportServerUrl>`) and `RSReportServer.config` (`<UrlRoot>`) files to reflect the URL where you are putting the virtual directories. This is why we told you about those elements earlier!

Windows Server 2003 has a nice "back door" route that can simplify some of the steps. Perform an ordinary installation of Reporting Services to the Default website. Next, take backups of the Reports and ReportServer virtual directories by right-clicking on each of them, and then under All Tasks select Save configuration to a file. You can then go to your new website, and under New select New virtual directory from file, and use each of the backups to create new virtual directories. You can then remove the virtual directories from the Default website. Yes, you still need to execute `aspnet_regiis -c` as described above, and you still need to edit the `RSWebApplication.config` and `RSReportServer.config` files.

Installation Log Files

During installation, the install wizard records all of its actions and checks to log files it maintains in a file called `RSStp_cab` in `\Microsoft SQL Server\80\RS Setup Bootstrap\Log`. In case there were any confusing setup issues or failure points that baffled you, these log files will be a lot more helpful and humanly readable to the average administrator than BSOD¹⁵ dumps are.

¹⁵ BSOD: Blue Screen of Death. This used to result from catastrophic failures of drivers and so on in the operating system, and huge dumps of the memory core would be moved to a file. Fortunately, we've not been plagued by any BSODs for a while.

Runtime Trace Log Files

There are three runtime trace log files that are created by default, and they all live in `\Microsoft SQL Server\<SQL Server Instance>\Reporting Services\LogFiles`. They all take the name form of `<logfile name><awkward US timestamp>.log`. Okay, why is this so awkward? Well, it seems to be `<month number>_<day of Month number>_<Year>_<Hours>_<Minutes>_<Seconds>`.

Here's an example: `ReportServer__01_02_2004_18_50_47.log`. It would have been ever so convenient if instead an ISO date format had been used, such as `yyyymmddhhnssReportServer.log`. Such a format would not have confused Europeans and Antipodeans and also would have more easily permitted sorting in the explorer and scripting programs.... Ah well, that aside you need to know what these log files trace, how you can configure how much detail they record, and what automated maintenance is performed on them.

The logs are:

- Report Server web service—`ReportServer<datetime>.log`
See the <RSTrace> element in the web.config file in \Microsoft SQL Server\MSSQL\Reporting Services\ReportServer
- Report Manager—`ReportServerWebApp<datetime>.log`
See the <RSTrace> element in the web.config file in \Microsoft SQL Server\<instance>\Reporting Services\ReportManager
- Report Server Windows Service—`<datetime>.log`
See <RSTrace> in ReportingServicesService.exe.config in \Microsoft SQL Server\<instance>\Reporting Services\Bin

Within the various `web.config` files, you'll notice that you can configure the log filename, file size, and retention times. You'll also find a peer element to `<RSTrace>` called `<system.diagnostics>` and within that an element `<switches>` with an attribute of `DefaultTraceSwitch` set to a default of 3. While the file itself documents in a comment possible values of `<!-- 1 = error, 2 = warning, 3 = info, 4 = verbose -->`, you can switch off logging entirely by setting this to 0.

If you are troubleshooting, it can be quite handy to attach a debugger to the `aspnet_wp.exe` process as new entries are echoed to the debug output window as well as written to the file. If you've not done this before, we have a video demo for you to watch.

After Installation—Tuning and Reconfiguring

After you've completed the Setup wizard, you might find it necessary to go back and change some of the settings. One approach you might consider is to uninstall and reinstall Reporting Services to make some of these changes. However, we don't encourage that as there are a number of splinters left behind that make this problematic. See the section later in this chapter on uninstalling before taking that route. You can change the configuration files as described here—if you're careful.

In any case, if you're using SSL security as we suggested, we recommend that you change both the `RSReportServer.config` and `RSWebApplication.config` files to include the correct URL and `SecureConnectionLevel` settings. These changes are described in detail below.

The Reporting Services Configuration Files

RSReportServer.config

We've already talked quite a bit about the configuration items in the `RSReportServer.config` file—its default location is `\Program Files\Microsoft SQL Server\MSSQL\Reporting Services\ReportServer`. We should mention a few more elements before discussing the other configuration files.

You can control which instance of SQL Server the Report Server web and Windows services uses through the `<InstanceId>MSSQL.1</InstanceId>` element—`MSSQL.1` is, of course, the default instance. When the Report Services file is created, a GUID `<InstallationID>` is created to uniquely identify an instance of the installation. This same GUID is used to index the encryption keys table we spoke of earlier. You may need to adjust other entries in the `RSReportServer.config` file—for example, when you want to create custom extension assemblies. However, we'll cover those elsewhere in the book.

SSL Configuration After Install

Ah! So you didn't listen to us about how to install SSL, or you installed oblivious to the importance of SSL and now want to be able to reconfigure to use SSL without having to take the pain of uninstall followed by reinstall. Well, you are in luck because this is controlled in the `RSReportServer.config` file by the entry: `<Add Key="SecureConnectionLevel" Value="3"/>` where the Value can be set as follows:

- 3 is the most secure—use SSL for absolutely everything
- 2, secure—use SSL for rendering but don't insist on it for all SOAP calls
- 1, basic security—accept HTTP but reject any calls that might be involved in the passing of credentials
- 0, the least secure—don't use SSL at all

The Report Server redirects the browser to HTTPS as needed, depending on the value of `SecureConnectionLevel`.

RSWebApplication.config

The Report Manager's main configuration file is `RSWebApplication.config`. By default, you'll find this file in the `\Program Files\Microsoft SQL Server\MSSQL_Reporting Services\ReportManager` folder. The most important thing this file contains is the URL configuration used by the Report Manager to contact the Report Server. If you change the address name of the Report Server, you'll have to come here to hook up the Report Manager. The entry you'll be looking for is

```
<ReportServerUrl>http://rs1.boost.net/ReportServer</ReportServerUrl>
```

(Replace `rs1.boost.net` with your own server name!)

Reporting Services BOL¹⁶ version 1 tells you not to modify this setting.

¹⁶ BOL: The Books Online master help file included with Reporting Services.

Another entry in this file determines the maximum number of requests that can come from each user at a given point in time. With the `<MaxActiveReqForOneUser>` element the default is 20 concurrent requests per user. There is a similar entry in the `RSReportServer.config` file too.

When the Report Manager encounters an error, it can display an error message with a link off to the Microsoft website for further updated information. You can control whether your users are presented with the link to this Microsoft site by adjusting the `<DisplayErrorLink>` element. The default is “true.”

RSReportDesigner.config

The Report Designer also has a configuration file: `RSReportDesigner.config`. This is located in the `\Program Files\Microsoft SQL Server\80\Tools\Report Designer` folder. For the most part, this file lists the custom data, designer, and render extensions. The entries that it contains are generally related to any custom extensions that you would like to run.

The Policy Configuration Files

In the Report Manager, Report Server, and Report Designer main configuration files, we studiously avoided mentioning the “policy level” keys so that we could deal with them here. There are three XML-like policy files: `rspreviewpolicy.config` for the Designer, `rsmgrpolicy.config` for the Report Manager, and `rssrvpolicy.config` for the Report Server. These configuration files determine how Reporting Services manages Code Access Security (CAS). For the most part, this is how Reporting Services implements security and trust in the CLR code assemblies that it’s prepared to load and execute. We’ll provide more detail on CAS when we deal with writing a custom extension and how you can declare what permission features your assembly needs—such as registry access, file system access, or network access. CAS also determines how someone installing your assembly can determine how much trust they want to give your assembly, and prevent it from doing unapproved things. This is all governed by the contents of the aforementioned policy files.

Policy Files Are *Not* XML Files

Policy files might look like hierarchical XML files, but they don't contain XML headers and, indeed, they certainly don't support XML comments `<!-- -->`

If when you end up editing them (as we will) you enter comments or (heaven forbid) change the case on something like `Url="$CodeGen$/*"` to `Url="$CODEGEN$/*"` you can expect trouble. The problem is that when things come tumbling down, you can get messages telling you that there are fatal errors in `Machine.config`, which you probably haven't touched. So if you get there, you won't have to suffer the pain we did, and boy did it hurt—two days of ripping ASP.NET to bits with `Machine.config`, and all because of a measly XML comment we'd put in a policy file to remind us what the previous value was.

The Reporting Services Command-Line Utilities

In some cases, you might need to tune or reconfigure the Reporting Services installation using one of the command-line tools. Each of these are provided to address one or more issues that can crop up in "off-the-beaten-path" (non-typical) installations.

Activation...rsactivate.exe

No, this is not some new-fangled license-related activation spy-ware scheme; it is a clever way to protect credentials. You don't normally need to be concerned about activation unless you are manually configuring a web farm, but you might need it to complete a problematic installation if you're using a non-default IIS configuration or if other required components are configured in a way that Setup does not expect. (This is a known issue with the first release and is high on the agenda to be addressed in future releases.) In most single-system installations, the installer takes care of activation for you, but we're still going to tell you what it is all about here so that you'll be aware of what goes on.

Reporting Services encrypts (both Windows and SQL) credentials that it uses in Reports and Data Sources. The technique is a reversible encryption using a symmetric key. Yes, it needs to be reversible—because how else is it going to be able to use those credentials if it can't work out what they are? During the installation, the Reporting Services Windows service co-operates with the

Reporting Services web service and generates the symmetric keys, which can be inspected in the Report Server Database Keys table. You should be able to see at least two entries there; one for the web service and one for the Windows service. If you create a web farm, you'll see further entries in the Keys table for each Windows service and web service in the farm. Until the web service and Windows service have been activated, neither can make use of content that has encrypted credentials. Reporting Services activates (which means that it shares between all the web and Windows services involved in the web farm the symmetric key) on first use of the web service. That said, sometimes things don't quite work out for some reason, and you may have to do surgery with the Reporting Services command-line tool `rsactivate`.

Key Management...`rskeymgmt.exe`

Microsoft provides an administrator's command-line utility, `rskeymgmt.exe`, to manage the symmetric key. For example, it's possible to use the `rskeymgmt` tool to back up the symmetric key to an encrypted file. To achieve this, `rskeymgmt` enlists the help of the Windows Reporting Service—so you have to make sure that the service and the SQL Server are both running. You can store the 52-byte backup key file on floppy or other removable media. Don't forget to ensure you also securely retain the password you use to encrypt the file—otherwise it's all a bit pointless.

You can also use `rskeymgmt.exe` to rip out *all* encrypted content from the Report Server by calling `rskeymgmt.exe -d`. Just before you do, be aware that this not only removes each encrypted copy of the symmetric key from the Keys table, but it also removes all encrypted credentials that shared Data Sources and reports use, and you are going to have to put them all back in. Note that subscriptions retain a pointer to encrypted credentials specified for the managed report's Data Source. After you run `-d` (and respecify your stored credentials), you have to manually open each subscription and resave it.

To “restore” a symmetric key using `rskeymgmt.exe`, you call it with the “apply” parameter, `-a`, and supply the backup file and password you used. If you removed encrypted content using `rskeymgmt.exe -d`, reapplying the symmetric key from a backup will restore only the key—it can't restore the deleted credentials; they're

72 Chapter 2 Installing and Configuring Reporting Services

gone forever. However, if it is your desire to only temporarily remove access to encrypted credentials, you could directly delete all rows in the SQL Server Keys table where the Client column is greater than or equal to 0. This will effectively remove the key, and as such the encrypted credentials will not be able to be unencrypted until you reapply the backup copy of the key.

Tips for `rskeygmt.exe` and `rsactivate.exe`

We experimented quite extensively with the Reporting Service command-line utilities. One thing we found helpful (when things were a little more challenging than usual) was to restart IIS (which can be done with the command-line `IISRESET`) and also to restart the Report Server Windows service.

What Is `rsconfig.exe`?

Well, we know that the Report Server web and Windows services must access the Report Server database, and to do that these services must use some form of Windows or SQL Login credentials. Those credentials are stored in an encrypted format in an XML-like configuration file called `RSReportServer.config`, which you'll find in the `\Program Files\Microsoft SQL Server\MSSQL\Reporting Services\ReportServer` folder.

The `rsconfig.exe` command-line tool provides a one-way mechanism to encrypt and store any updated credentials in the configuration file. It's also used to specify credentials used for unattended processing for reports that do not use credentials. As you know, you can choose "No credentials" for a report definition or managed report's Data Source. This creates a security hole because the Report Server had to log on to the remote database server using its own credentials. This has been fixed by the introduction of a requirement for credential information—users must now run `rsconfig` with the `-e` option to specify a `-u` user name and `-p` password that the Report Server uses to log on to a remote data server.

Report Server Windows Service Polling

Scheduled Jobs, be they subscriptions or snapshots, are managed by the SQL Server Agent. However, it's the Report Server Windows

service that processes these reports. On the off chance that there is some processing to do, the Report Server Windows service polls the database every 10 seconds (by default). Okay, so you might think that the Microsoft development team has taken a shortcut here, as polling instinctively feels like the wrong approach. You could be forgiven for thinking that all the Microsoft developers needed to do was to put the Windows service to sleep and wake it up with a signal. For example, by creating a signaling procedure with the COM invoking the `sp_OACreate` stored procedure, or with an extended procedure. That might seem simple enough, but what if the Report Server's web components are on a server that does not host the Reporting Services database? In this case, you not only have to cross process boundaries, you've got to get onto another machine. Ah, DCOM? .NET Remoting? A bespoke IP message? (What about firewalls?) Notification Services? Or MSMQ? This is more apparent if you consider the web farm scenario, in which none of the machines know about other machines in the farm. Any machine in the farm can pull an event from the database queue. Who decides which machine gets the event? What if a machine goes down? Does the controller ping them to see if they want the event? So this seemingly simple situation becomes more complicated to program and more complicated to install and troubleshoot, and the polling solution that we have for now is probably the best compromise. We agree that a pull model works best in this scenario.

However, if you don't like the idea of polling every 10 seconds, then there is something you can do about it. In the `RSReportServer.config` file, you can change the polling element by editing `<Polling-Interval>10</PollingInterval>`.

Testing the Installation

If you installed with SSL enabled, you'll probably find that there is a change you'll need to make to the sample reports Microsoft provides. In most cases, you'll need to correct the project's `TargetServerURL` to reflect the `https://` address of your Report Server.

74 Chapter 2 Installing and Configuring Reporting Services

Let's quickly show you how to use Visual Studio .NET to load, build, and deploy the sample reports. These reports will help verify several aspects of the install and give you a degree of confidence that your installation has succeeded. Let's step through the process:

1. Start Visual Studio .NET on your development system.
2. Open the SampleReports.sln solution file (File|Open|Project...). The installation wizard gave you the option of installing the sample reports—we assume you chose to install the report samples. If you changed the path where these were installed, you're on your own, but if you used the default location, you'll find these samples at *C:\Program Files\Microsoft SQL Server\MSSQL\Reporting Services\Samples\Reports*.
3. Open the Solution Explorer for the SampleReports project, as shown in Figure 2.7.

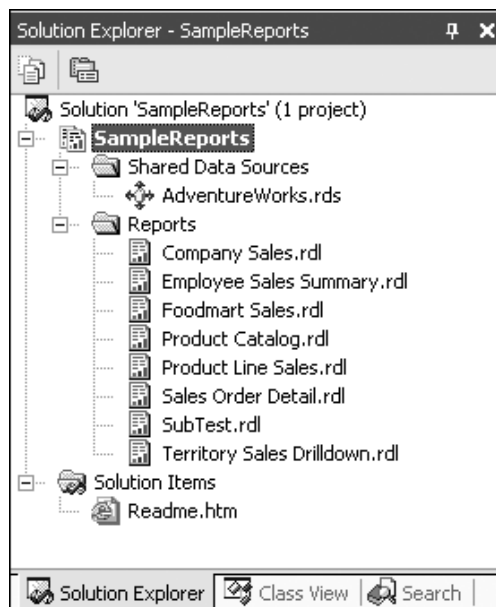


Figure 2.7 The Solution Explorer Window for SampleReports

4. Change the SampleReports TargetServerURL to the URL of your IIS ReportServer virtual directory. To access this property, right-click the SampleReports project in the Solution Explorer and choose Properties. If you're using SSL security, the URL needs to specify https:// instead of http://. Because the IIS ReportServer might not be on the developer's local system—and it's likely not to be—you also need to set the correct server name instead of "localhost."

Figure 2.8 shows how we changed the TargetServerURL to https://betav1/reportserver so that it targets the Betav1 IIS ReportServer virtual directory. This name must match the Common Name you specified in the IIS SSL certificate. Click OK to close the Property page and apply the changes.

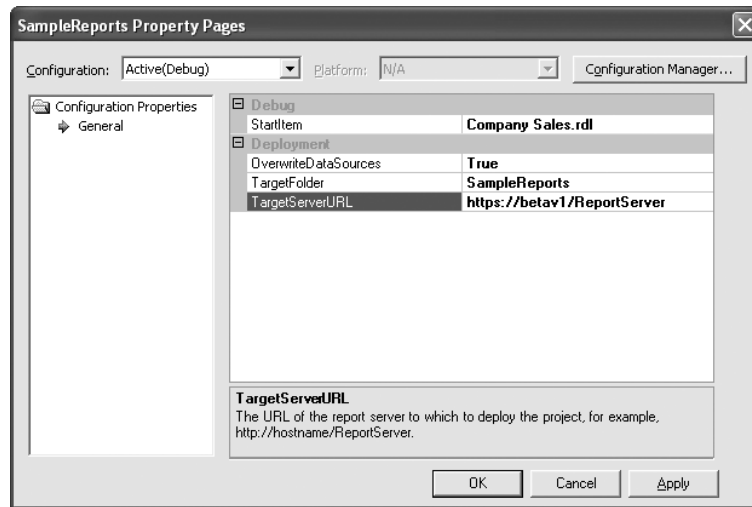
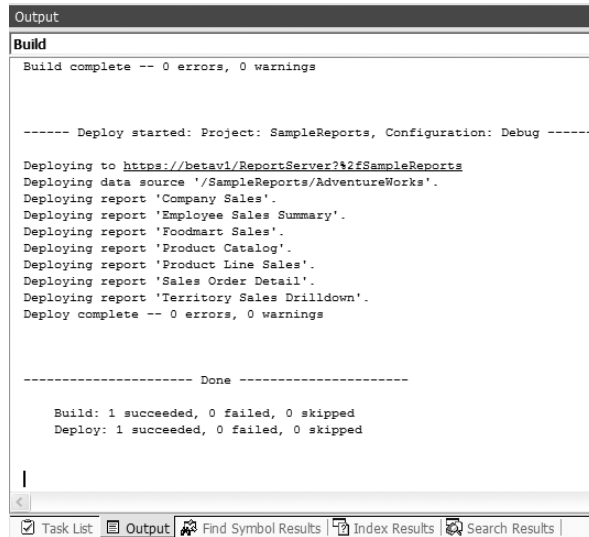


Figure 2.8 Change the TargetServerURL to Point to the Correct IIS ReportServer Virtual Directory

5. Deploy the SampleReports project, paying close attention to the Output window. To start deployment, right-click on the SampleReports project and choose Deploy. After the SampleReports project is built, each report definition should deploy one at a time, as shown in Figure 2.9.

76 Chapter 2 Installing and Configuring Reporting Services



```

Output
Build
Build complete -- 0 errors, 0 warnings

----- Deploy started: Project: SampleReports, Configuration: Debug -----

Deploying to https://betav1/ReportServer?%2fSampleReports
Deploying data source '/SampleReports/AdventureWorks'.
Deploying report 'Company Sales'.
Deploying report 'Employee Sales Summary'.
Deploying report 'Foodmart Sales'.
Deploying report 'Product Catalog'.
Deploying report 'Product Line Sales'.
Deploying report 'Sales Order Detail'.
Deploying report 'Territory Sales Drilldown'.
Deploy complete -- 0 errors, 0 warnings

----- Done -----

Build: 1 succeeded, 0 failed, 0 skipped
Deploy: 1 succeeded, 0 failed, 0 skipped

```

Figure 2.9 The Visual Studio .NET Output Window Shows the Deployment Was Successful

- Use a browser to address the TargetServerURL you entered above. Remember, this is the IIS ReportServer virtual directory. For our test, we'll use `https://betav1/reportserver`. If this works, you should get a page that looks something like Figure 2.10. If you get a Page Not Found error or a permissions error, you have missed a step somewhere or you don't have sufficient rights to see this server.

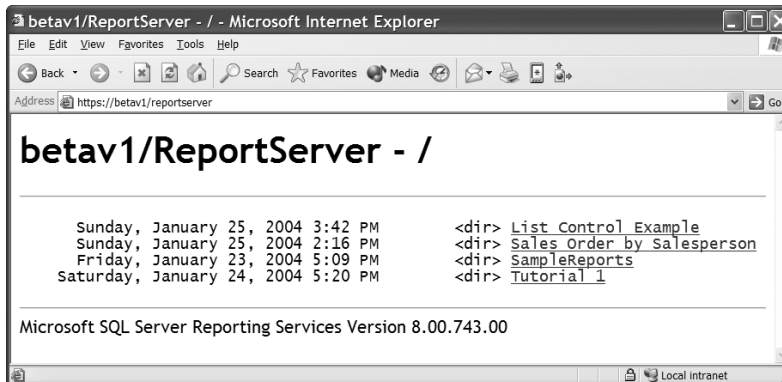


Figure 2.10 Using a Browser to Address the ReportServer Virtual Directory

7. To view one or more of the sample reports, click on the SampleReports link. This should show a list of the sample reports you just deployed, as shown in Figure 2.11.

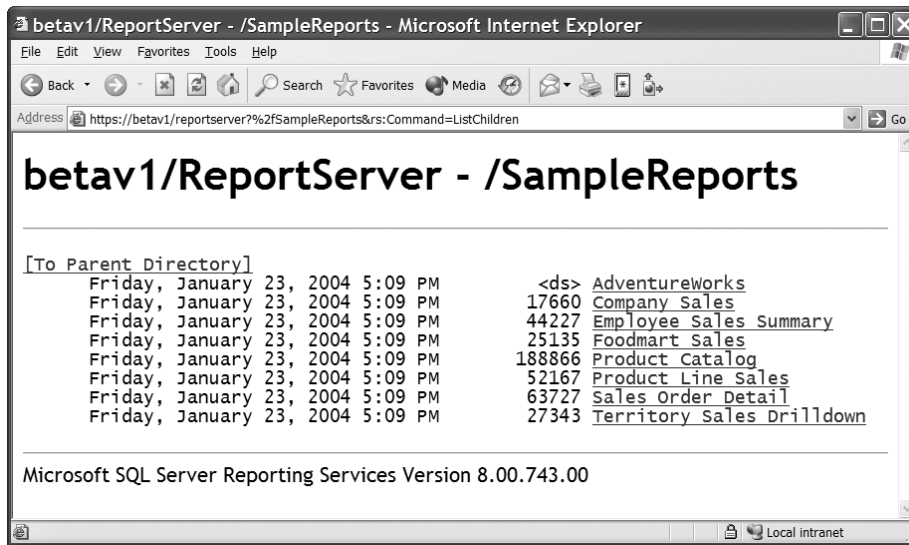
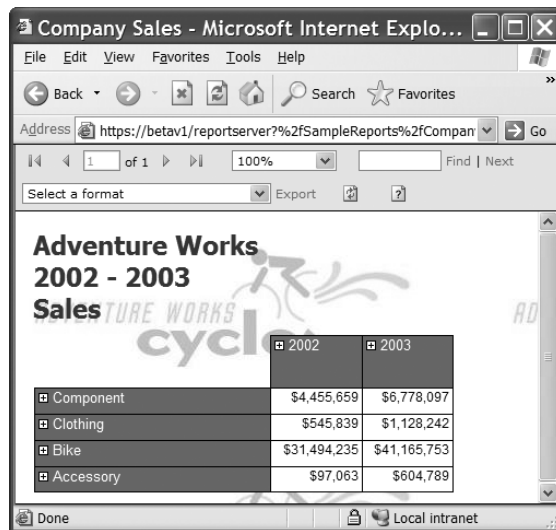


Figure 2.11 The betav1/ReportSever SampleReports Directory

8. Click on one of the reports in the list to get Reporting Services to render the report. After a window appears that tells you the report is being generated, the selected report should be rendered in the browser. We selected Company Sales, and Figure 2.12 shows how this is rendered in the browser.
9. As a final step, you might like to check that the SQL Server Agent is running, in addition to the Report Services Windows service—as they both are used with Report Subscription. We'll tell you more about that in Chapter 4, however, so don't get overly worried about it here.

Once you've completed these steps, you should be ready to continue with the rest of the book with a degree of confidence that you can create, deploy, and view reports in a browser. We'll get into more detail on developing, managing, publishing, and fine-tuning reports in subsequent chapters.

78 Chapter 2 Installing and Configuring Reporting Services



Component	2002	2003
Component	\$4,455,659	\$6,778,097
Clothing	\$545,839	\$1,128,242
Bike	\$31,494,235	\$41,165,753
Accessory	\$97,063	\$604,789

Figure 2.12 The Company Sales Report as Rendered in a Browser

Installing Reporting Services on a Web Farm

This section is for people running the SQL Server Enterprise, Developer, or Evaluation versions who want to share the Report Server database with a number of IIS servers and so create a web farm. Before you follow any of these steps, you first need to perform one normal installation of Reporting Services because the steps show you how to add additional IIS servers to an already working installation.

Web Farm Installations

You can take two approaches when installing Reporting Services technology on a web farm.

1. During the installation wizard, specify that the SQL Server instance to contain the Report Server database is already located on another machine.

2. Take a preconfigured Reporting Services installation and change the DSN Connection to the Report Server databases and the account credentials that should be used to connect to that database—in the RSReportServer.config file.

We'll walk you through both of these configuration options.

Specify a Different SQL Server Instance

This first approach is fairly straightforward, but it has a restriction: both systems have to be in the same domain or in trusted domains. To start with, prior to running Reporting Services Setup, use Enterprise Manager on the target SQL Server to create a new Domain user named “WebFarmUser” (as shown in Figure 2.13). Next, give this user permission to log in to the Reporting Services database and assign that user to the RSExec role in those catalogs. During Reporting Services installation, specify that the SQL Server instance that you want to be used is located on the SQL Server machine with the WebFarmUser user account.



Figure 2.13 “D2” Is a Secondary Development Server Where Reporting Services Was Already Installed. This Screen Was Captured While Executing Setup on Server “D1”

80 Chapter 2 Installing and Configuring Reporting Services

Once you specify a different server to host the SQL Server instance, the installer recognizes this is not a local Report Server database, and assumes that this is a web farm installation. At this point, you'll be presented with an additional dialog (as shown in Figure 2.14) to capture details of an already working installation in the web farm and login details to that installation. These details are used by the wizard to ensure that the symmetric key is correctly configured for the new member of the farm.

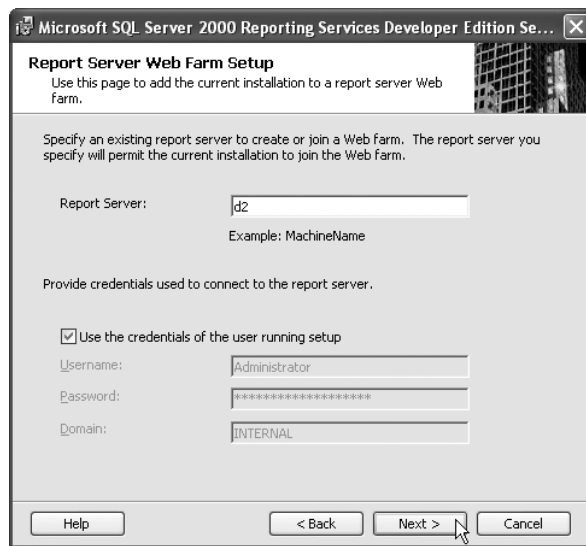


Figure 2.14 Specifying a Pre-Existing Report Server

Reconfigure an Existing Reporting Services Configuration

The other route you can take to setup a Reporting Services web farm configuration is to take an existing Reporting Services installation and change the DSN Connection to the Report Server databases and the account credentials that should be used to

connect to those databases. These changes are made to the RSReportServer.config file. The problem is that the details you need to change are encrypted in the file, so you have to use the rsconfig.exe command-line tool and specify the information. For example, the following command-line statement shows how to make the needed changes:

```
rsconfig.exe -c -s <Server Hosting RS DB> -d <RS Database Name> -l  
-a Windows -u "<Domain Name>\<User>" -p "<password>"
```

- Having done this, next you need to use Enterprise Manager against the SQL Server hosting the ReportServer database. Here you enable the account you've just specified to log in to the Server with access to the ReportServer and ReportServerTempDB databases. You also need to make this account a member of the RSExec role in both the ReportServer and ReportServerTempDB databases.
- Restart the Report Server Windows service, and restart IIS (you can restart IIS at the command line with IISRESET)
- If you are changing the database that a Report Server used to use, you might like to disable the old database at this point.
- Try to call up the Report Server or the Report Manager in the browser. At this point, it should fail—telling you that the Report Server installation is “not initialized.” Don't let that worry you unduly. This is a good sign, as it confirms that you are no longer pointing at any previous installation. However, if you get a message about bad passwords, you need to revisit the rsconfig command line utility and review the account name and password you used. You also need to double-check that you made that account a member of the RSExec role in the two databases as described.
- As a further check, look in your RSReportServer.config file and find the GUID in the <InstallationID> node. Note the GUID and examine the Keys table in the ReportServer

82 Chapter 2 Installing and Configuring Reporting Services

database. You should be able to find two entries with this GUID. You'll find only one if you haven't tried to access the Report Server or Report Manager via the browser yet.

- Assuming that you have found the two entries in that Keys table (in addition to the other entries that will be in there from the other members of the farm), you should now be able to “initialize” or “activate” this installation. The easiest way to do this is to log on to any other machine that is already working in the web farm and execute `rsactivate`. No, you can't do this on the machine you are trying to get to join the farm.

```
rsactivate -m <machine to be added> -u <Administrator User>
-p <Password>
```

- Retry accessing the Report Server in the browser on the newly added machine.

At this point, you should be able to have a working Reporting Services configuration on your web farm.

Removing Reporting Services

To uninstall or reconfigure Reporting Services, you can simply start the Setup wizard again using Windows Control Panel Add/Remove Programs. This launches a new dialog (as shown in Figure 2.15) that permits you to change the installed components or remove Reporting Services. Be aware that if you try to uninstall, the wizard doesn't do a complete tear down. It seems that after running the uninstall program you're going to have to get down on your hands and knees and pick a few splinters out of the system by hand.

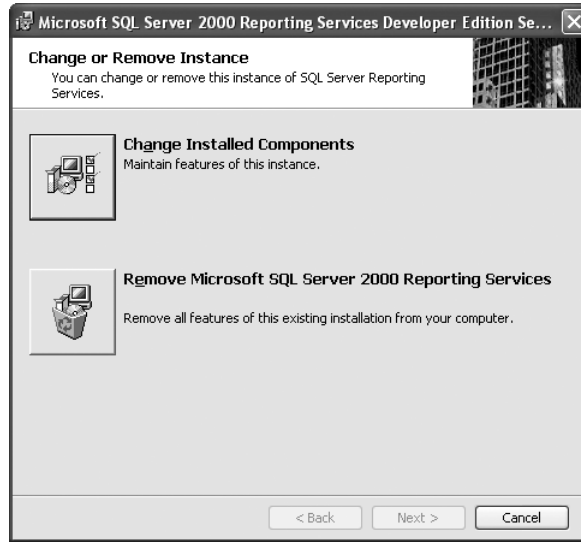


Figure 2.15 Relaunching the Setup Wizard from Add/Remove Programs

Removing the Splinters Left Behind After Uninstall

Let's start with the databases created to support Reporting Services and host your reports. One of the problems is that the Reporting Services database can contain user data that you really want to keep. Just because the application is being uninstalled doesn't necessarily mean that you want to remove your user data too. In a web farm configuration, the uninstall routines can't tell if this is the last machine being uninstalled. Just imagine the chaos if, when uninstalling Microsoft Word, it also removed all your Word documents. In the same way, the uninstall program leaves the ReportServer and the ReportServerTempDB (or whatever you renamed them during installation) databases behind, so you (might) need to delete them by hand.

Deleting all of the things Reporting Services has installed and configured on your SQL Server can be problematic and tedious. This is especially true if you had any Scheduled Report jobs or

Subscription jobs. That's because these jobs have entries for the SQL Server Agent squirreled away in the MSDB database. Here's a warning: don't go digging into MSDB to remove any Report Services SQL Server Agent jobs. It's much easier to make these changes using the Enterprise Manager by navigating to Management | SQL Server Agent | Jobs and deleting those dead jobs manually. You will, however, need to go digging into the Master and the MSDB databases to remove the RSExec roles that were created during the install, but you'll have to delete the members of the role first.

But wait, there is a bit more to tear down. If you launch the Component Services MMC snap-in from the Administrative Tools, and navigate down to Component Services | Computers | <My Computer> | COM+ Applications, you'll find at least a pair of IIS applications for ReportServer and Reports. We say at least a pair because if you've been installing and uninstalling you'll see a pair of applications for each time you installed! You can safely remove these from the system, although in order to do so you'll need to select each application's Properties dialog and uncheck Disable deletion on the Protection tab. This will permit you to delete them.

Reporting Services Documentation Refresh

Shortly after releasing Reporting Services, Microsoft released a refresh to the documentation. To achieve consistency in the formal documentation among the nine language translations in which SQL Server is released, the documentation team had to freeze the documentation while the product was still under active development. So if you want to have documentation that more accurately reflects the released product, take a hike over to www.microsoft.com and search for RSDocRefresh.msi. The article is titled "Reporting Services Books Online January 2004 Update" and (currently) links from <http://go.microsoft.com/fwlink/?LinkId=29077>. You can simply download the RSDocRefresh.msi and double-click on it to install. After installation, expect Visual Studio .NET 2003 to take some time when you next open it to update its help file indexes.

Reporting Services Service Pack 1

To keep the download as small as possible, Reporting Services Service Pack 1 (SP1) doesn't install the Documentation Refresh. So be aware that you'll need to install the Documentation Refresh separately.

SP1 is supplied as a single executable, and it upgrades all versions of Reporting Services (Enterprise, Developer, Standard, and Evaluation). It also updates all the installation configurations. To install it, simply execute it on the development machines and servers. While it's independent of Service Pack 4 for SQL Server 2000, Microsoft recommends that SQL Server Service Pack 4 be applied first—once it's released. We guess this means that you may need to re-apply the Reporting Services Service Pack after you've installed SQL Server Service Pack 4. <Sigh> However, *do not* postpone installing Reporting Services Service Pack 1—it contains certain security enhancements that we strongly advise be applied.

Summary

In this chapter, we discussed some of the more common setup scenarios—from installing everything on one development system to installing the various Reporting Services components on different machines. We also walked you through a number of configuration and testing steps that should help you get started with Reporting Services more quickly and with fewer hassles.

While we can't give advice for every conceivable configuration, we invite you to participate in our community at www.SQLReportingServices.NET. Here you can let us know any issues you encounter, and we'll try to help you solve them. You might also share with us any solutions you find for particular problems. As Reporting Services evolves, we envisage that the most up-to-date best practice suggestions will be posted on our website.

TIP A Final Suggestion Whatever Reporting Services installation configuration you have built, we strongly recommend that you use the `rskeymgmt.exe` utility, back up the symmetric key to offline storage, and secure it along with the password you used to encrypt the file.

