Beginning SQL Server 2008 for Developers

From Novice to Professional

Robin Dewson

Apress®

Beginning SQL Server 2008 for Developers: From Novice to Professional

ISBN-13 (pbk): 978-1-59059-958-7

ISBN-10 (pbk): 1-59059-958-6

ISBN-13 (electronic): 978-1-4302-0584-5

ISBN-10 (electronic): 1-4302-0584-9

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Jonathan Gennick
Technical Reviewer: Jasper Smith
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Kylie Johnston
Copy Editor: Nicole Abramowitz
Associate Production Director: Kari Brooks-Copony
Production Editor: Ellie Fountain
Compositor: Susan Glinert
Proofreader: Nancy Sixsmith, ConText Editorial Services, Inc.
Indexer: Broccoli Information Management
Artist: Kinetic Publishing Services, LLC
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at http://www.apress.com/info/bulksales.

The source code for this book is available to readers at http://www.apress.com.

This book, as many of my books are, is dedicated to my family. First, to my mum and dad whom I love very much and who made me what I am today. Without their help, understanding, and patience when it came to my use of the television for the Sinclair ZX80 and the Sinclair ZX81, and without helping me find and organize my further education, I probably would have wasted a great opportunity. To my three kids—Ellen, Cameron, and Scott—who have been brilliant and wonderful and whom I am very, very *proud of in many ways. And they are such great kids because they have who can only be termed the best mother kids can have, right there helping, loving, and, yes, screaming at them when needed. Julie, I love you more than I love my Irn-Bru, rugby, and lemon meringue pie . . . and you know how much those mean to me!*

Contents at a Glance

Contents

About the Author

ROBIN DEWSON has been hooked on programming ever since he bought his first computer, a Sinclair ZX80, in 1980. His first major application was a Visual FoxPro program that could be used to run a fantasy league system. It was at this point that he met someone who would become a great help in the development of his PC life, Jon Silver at Step One Technologies. In return for training, Robin helped Jon with some other Visual FoxPro applications. From there, realizing that the marketplace for Visual FoxPro was limited, Robin decided to learn Visual Basic and SQL Server.

Starting out with SQL Server 6.5, Robin soon moved to SQL Server 7, accessing the database via Visual Basic 5. He became involved in developing several applications for clients both in the UK and in the United States. From there, he moved through SQL Server 2000 and 2005 and through Visual Basic 6 and C#. Robin now specializes in using Visual Studio .NET to write C# applications against SQL Server 2008.

Robin has several Apress books on SQL Server available. You can contact him at `robin@fat-belly.com` or at `http://www.fat-belly.com`.

About the Technical Reviewer

JASPER SMITH is an independent SQL Server consultant specializing in scalability, availability, and manageability, and he has worked with SQL Server for the past eight years. He is a Microsoft SQL Server Most Valued Professional (MVP) and is a frequent speaker at Professional Association for SQL Server (PASS) conferences. He runs and authors content for his web site, `http://www.sqldbatips.com`, which specializes in free SQL Server tools such as Reporting Services Scripter, ExpressMaint, and SQL 2005 Service Manager.

Acknowledgments

Once again, there are so many people to thank, from the great Damian Fisher, for teaching me how to play the drums, to Andrew and everyone at host-it Internet Solutions, my ISP, for putting up with my incessant hassling over SQL Server and DotNetNuke. Thanks to my bosses Bill Cotton and Aubrey Lomas and coworker Andrew O'Donnell at Lehman Brothers, as well as Andrew Harding, a great DBA; and thanks to Robert McMillan (Toad), a great mate from college with whom I got back in touch after many years. Cheers also to Simon Collier for whipping me at table tennis week in and week out. Thanks to all at Bedford Blues Rugby Club who make my Saturdays, well, exciting.

There are several people at Red Gate Software whom I have to thank for many reasons, including Tony Davis, who has been brilliant for many years with my SQL Server work, and Richard Collins, who organized the Apress and Red Gate collaborations. Also, thanks to Salar Golestanian at SalarO for producing excellent skins that I use on my web site creations.

Thanks also to my mother-in-law, Jean, for being so brilliant when things needed doing and keeping my wife sane. And to my late father-in-law, David, who was a brilliant person with all in my family.

Thanks, of course, to all at Apress, especially Kylie Johnston and Jonathan Gennick for help with this book, as well as Paul Carlstroem and Gary Cornell.

And finally, I have to acknowledge Mr. and Mrs. Barr for making Scotland's other national drink.

Introduction

Beginning SQL Server 2008 for Developers is for those who see themselves as becoming developers, database administrators, or a mixture of both but have yet to tread that path with SQL Server 2008. Whether you have no knowledge of databases, have knowledge of desktop databases such as Microsoft Access, or even come from a server-based background such as Oracle, this book will provide you with the insight to get up and running with SQL Server 2008.

Right from the start, this book will expand your basic knowledge, and you will soon find yourself moving from a beginner toward a competent and professional developer. This book aims to cater to a wide range of developers, from those who prefer to use a graphical interface for as much work as possible, to those who want to become more adept at using SQL Server 2008's programming language, Transact SQL (T-SQL). Where practical, this book demonstrates, explains, and expands upon each method of using SQL Server 2008 so that you can evaluate which works best in your situation.

This book contains plenty of examples that let you see how areas of SQL Server work and how you can apply the technology in your own work. You will learn the best way to complete a task, and you'll even learn how to make the correct decision when presented with two or more choices. Once you reach the end of this book, you will be able to design and create solid and reliable database solutions competently and proficiently.

Who This Book Is For

This book is ideal for developers starting out with SQL Server 2008 or for those looking at becoming database administrators. The book is laid out so that it works well with both of these camps.

How This Book Is Structured

The book helps you decide which version of SQL Server 2008 to buy, shows you how to install and configure SQL Server 2008, and explains how to use the graphical user interface (GUI) tool, SQL Server Management Studio. You will use this tool to work through a fully functional database example that is built from a design covered within the book, using graphical and code-based try-it-out exercises. You'll then learn about the security of your database and how to enforce a secure and reliable database setup. Once you back up your database, you'll learn how to work with the data. You'll start with simple coding techniques and move on to more complex ones. Your final task will allow you to build and produce reports on your database. Throughout the book, I explain each item, informing you of what is happening and ensuring that as you progress through the book, you're building on knowledge gained from previous chapters. You'll be building your expertise in a clear and structured manner.

Prerequisites

You will need to have either an evaluation or full version of SQL Server 2008 Developer Edition. In addition, it's ideal, but not compulsory, to have either the Windows Vista Ultimate or Business version if you want to alter security settings for specific Windows logins.

Downloading the Code

Example code from this book is available via the Apress web site (`http://www.apress.com`). Navigate to the book's home page, then look in the left sidebar for the section called "Book Extras." You'll find the link to the code examples there.

Contacting the Author

You can contact Robin Dewson via e-mail at `robin@fat-belly.com` or via his web site at `http://www.fat-belly.com`.

CHAPTER 1

SQL Server 2008 Overview and Installation

Welcome to *Beginning SQL Server 2008 for Developers*. This book has been written for those who are interested in learning how to create solutions with Microsoft SQL Server 2008, but have no prior knowledge of SQL Server 2008. You may well have had exposure to other databases, such as MySQL, Oracle, or Microsoft Access, but SQL Server uses different interfaces and has a different way of working compared to much of the competition. The aim of this book is to bring you quickly up to a level at which you are developing competently with SQL Server 2008. This book is specifically dedicated to beginners and to those who at this stage wish to use only SQL Server 2008. It is also for those developers who have SQL Server 2005 experience and want a quick method to get up to speed on SQL Server 2008. You may find this book useful for understanding the basics of other databases in the marketplace, especially when working with T-SQL. Many databases use an ANSI-standard SQL, so moving from SQL Server to Oracle, Sybase, etc., after reading this book will be a great deal easier.

This chapter covers the following topics:

- Why SQL Server 2008?
- How do I know if my hardware meets the requirements?
- Can I just confirm that I have the right operating system?
- What can I do with SQL Server 2008?

We will also then look at installing our chosen edition and cover the following:

- Installing SQL Server 2008 on a Windows XP platform
- Options not installed by default
- Where to install SQL Server physically
- Multiple installations on one computer
- How SQL Server runs on a machine
- How security is implemented
- Logon IDs for SQL Server, especially the sa (system administrator) logon

Why SQL Server 2008?

The following discussion is my point of view, and although it no doubt differs from that of others, the basis of the discussion holds true. SQL Server faces competition from other databases, not only from other Microsoft products such as Microsoft Access and Microsoft Visual FoxPro, but also from competitors such as Oracle, Sybase, DB2, and Informix, to name a few.

Microsoft Access is found on a large number of PCs. The fact that it is packaged with some editions of Office and has been around for a number of years in different versions of Office has helped make this database ubiquitous; in fact, a great number of people actually do use the software. Unfortunately, it does have its limitations when it comes to scalability, speed, and flexibility, but for many small, in-house systems, these areas of concern are not an issue, as such systems do not require major database functionality.

Now we come to the serious competition: Oracle and Sybase. Oracle is seen as perhaps the market leader in the database community, and it has an extremely large user base. There is no denying it is a great product to work with, if somewhat more complex to install and administer than SQL Server; it fits well with large companies that require large solutions. There are many parts to Oracle, which make it a powerful tool, including scalability and performance. It also provides flexibility in that you can add on tools as you need them, making Oracle more accommodating in that area than SQL Server. For example, SQL Server 2008 forces you to install the .NET Framework on your server whether you use the new .NET functionality or not. However, Oracle isn't as user friendly from a developer's point of view in areas like its ad hoc SQL Query tool and its XML and web technology tools, as well as in how you build up a complete database solution; other drawbacks include its cost and the complexity involved in installing and running it effectively. However, you will find that it is used extensively by web search engines, although SQL Server could work just as effectively. With the new functionality in SQL Server 2008, Oracle will be under pressure to expand its existing functionality to meet this challenge. SQL Server has always been a one-purchase solution, such that (providing you buy the correct version) tools that allow you to analyze your data or copy data from one data source such as Excel into SQL Server will all be "in the box." With Oracle, on the other hand, for every additional feature you want, you have to purchase more options.

Then there is Sybase. Yes, it is very much like SQL Server with one major exception: it has no GUI front end. Sybase iAnywhere, which is mainly used for small installations, does have a front end, but the top-of-the-range Sybase does not. To purists, there is no need for one, as GUI front ends are for those who don't know how to code in the first place—well, that's their argument, of course, but why use 60+ keystrokes when a point, click, and drag is all that is required?

Sybase is also mainly found on Unix, although there is a Windows version around. You can get to Sybase on a Unix machine via a Windows machine using tools to connect to it, but you still need to use code purely to build your database solution. It is very fast and very robust, and it is only rebooted about once, maybe twice, a year. Another thing about Sybase is that it isn't as command- and feature-rich as SQL Server. SQL Server has a more powerful programming language and functionality that is more powerful than Sybase.

Each database has its own SQL syntax, although they all will have the same basic SQL syntax, known as the ANSI-92 standard. This means that the syntax for retrieving data, and so on, is the same from one database to another. However, each database has its own special syntax to maintain it, and trying to use a feature from this SQL syntax in one database may not work, or may work differently, in another.

So SQL Server seems to be the best choice in the database marketplace, and in many scenarios it is. It can be small enough for a handful of users, or large enough for the largest corporations. It doesn't need to cost as much as Oracle or Sybase, but it does have the ability to scale up and deal with terabytes of data without many concerns. As you will see, it is easy to install, as it comes as one complete package for most of its functionality, with a simple install to be performed for the remaining areas if required.

Now that you know the reasons behind choosing SQL Server, you need to know which versions of SQL Server are out there to purchase, what market each version is aimed at, and which version will be best for you, including which version can run on your machine.

Evolution of SQL Server

SQL Server has evolved over the years into the product it is today. Table 1-1 gives a summary of this process.

Table 1-1. *The Stages in the Evolution of SQL Server*

Year	Version	Description
1988	SQL Server	Joint application built with Sybase for use on OS/2.
1993	SQL Server 4.2, a desktop database	A low-functionality, **desktop** database, capable of meeting the data storage and handling needs of a small department. The concept of a database that was integrated with Windows and had an easy-to-use interface proved popular.
1994		Microsoft splits from Sybase.
1995	SQL Server 6.05, a small business database	Major rewrite of the core database engine. First "significant" release. Improved performance and significant feature enhancements. Still a long way behind in terms of the performance and feature set of later versions, but with this version, SQL Server became capable of handling **small e-commerce and intranet** applications, and was a fraction of the cost of its competitors.
1996	SQL Server 6.5	SQL Server was gaining prominence such that Oracle brought out version 7.1 on the NT platform as direct competition.
1998	SQL Server 7.0, a web database	Another significant rewrite to the core database engine. A defining release, providing a reasonably powerful and feature-rich database that was a truly viable (and still cheap) alternative for **small-to-medium businesses**, between a true desktop database such as MS Access and the high-end enterprise capabilities (and price) of Oracle and DB2. Gained a good reputation for ease of use and for providing crucial business tools (e.g., analysis services, data transformation services) out of the box, which were expensive add-ons with competing databases.
2000	SQL Server 2000, an enterprise database	Vastly improved performance scalability and reliability sees SQL Server become a major player in the **enterprise database** market (now supporting the online operations of businesses such as NASDAQ, Dell, and Barnes & Noble). A big increase in price (although still reckoned to be about half the cost of Oracle) slowed initial uptake, but the excellent range of management, development, and analysis tools won new customers. In 2001, Oracle (with 34% of the market) finally ceded its No. 1 position in the Windows database market (worth $2.55 billion in 2001) to SQL Server (with 40% of the market). In 2002, the gap had grown, with SQL Server at 45% and Oracle slipping to 27%.[a]

Table 1-1. *The Stages in the Evolution of SQL Server (Continued)*

Year	Version	Description
2005	SQL Server 2005	Many areas of SQL Server have been rewritten, such as the ability to load data via a utility called Integration Services, but the greatest leap forward was the introduction of the .NET Framework. This allowed .NET SQL Server–specific objects to be built, giving SQL Server the flexible functionality that Oracle had with its inclusion of Java.
2008	SQL Server 2008	The aim of SQL Server 2008 is to deal with the many different forms that data can now take. It builds on the infrastructure of SQL Server 2005 by offering new data types and the use of Language Integrated Query (LINQ). It also deals with data, such as XML, compact devices, and massive database installations, that reside in many different places. Also, it offers the ability to set rules within a framework to ensure databases and objects meet defined criteria, and it offers the ability to report when these objects do not meet this criteria.

[a] *Gartner Report, May 21, 2003*

Hardware Requirements

Now that you know a bit about SQL Server, the next big question on your list may well be, "Do I have a powerful enough computer to run my chosen SQL Server edition on? Will this help me refine my decision?"

Judging by today's standards of minimum-specification hardware that can be bought—even the low-cost solutions—the answer will in most cases be "Yes" to most editions. However, you may have older hardware (things move so fast that even hardware bought a couple of months ago can quickly be deemed below minimum specification), so let's take a look at what the minimum recommendations are and how you can check your own computer to ensure that you have sufficient resources.

CPU

The minimum recommended CPU that SQL Server will run on is a 1GHz processor for the 32-bit edition and a 1.6GHz for the 64-bit version, or a compatible processor, or similar processing power; however, 2GHz is recommended. However, as with most minimums listed here, Microsoft wholly recommends a faster processor. The faster the processor, the better your SQL Server will perform, and from this the fewer bottlenecks that could surface. Many of today's computers start at 2GHz or above, but the faster the processer the better. You will find your development time reduced for it.

However, it is not processor alone that speeds up SQL Server. A large part is the amount of memory that your computer has.

Memory

Now that you know you have a fast enough processor, it is time to check whether you have enough memory in the system. SQL Server requires a minimum of 512MB of RAM onboard your computer, although you shouldn't have too many more applications open and running, as they could easily not leave enough memory for SQL Server to run fast enough. Microsoft recommends 1GB or above, and really double that at least for when you start using your SQL Server in earnest.

If you wanted to run the Enterprise Edition, then a minimum, and I mean a bare minimum, of 1GB really should be installed, especially if you want to use any of the more advanced features.

The more memory the better: I really would recommend a minimum of 1GB on any computer that a developer is using, with 2GB ideal and sufficient to give good all-around performance. If a process can be held in memory, rather than swapped out to hard drive or another area while you are running another process, then you are not waiting on SQL Server being loaded back into memory to start off where it left off. This is called **swapping**, and the more memory, the less swapping could, and should, take place.

Taking CPU speed and memory together as a whole, it is these two items that are crucial to the speed that the computer will run, and having sufficient speed will let you develop as fast as possible.

When it comes to installing SQL Server, insufficient memory won't stop the install, but you will be warned that you need more.

Hard Disk Space

You will need lots! But name a major application these days that doesn't need lots! For SQL Server alone, ignoring any data files that you are then going to add on top, you will need over 1GB of space. Certainly, the installation options that will be used later in the chapter will mean you need this amount of space. You can reduce this by opting not to install certain options—for example, Books Online; however, even most notebooks these days come with a minimum 40GB, and 80GB is not uncommon either. Hard disk space is cheap as well, and it is better to buy one disk too large for your needs than have one hard drive that suits now, and then have to buy another later, with all the attendant problems of moving information to clear up space on the original drive.

Again, you will need spare space on the drive for the expansion of SQL Server and the databases, as well as room for temporary files that you will also need in your development process. So think big—big is beautiful!

Operating System Requirements

You will find that SQL Server 2008 will run on Windows Vista Home Basic Edition and above, as well as Windows XP. From the server side, it will work on Windows Server 2003 with Service Pack 2 and Windows Server 2008. It will also work on the 64-bit operating systems for Windows XP Professional, as well as the 64-bit editions of Windows Server 2003 and 2008. So there is plenty of scope for running SQL Server on many operating systems.

The Example

In order to demonstrate SQL Server 2008 fully, together we will develop a system for a financial company that will have features such as banking, share purchase, and regular buying, including a unit trust savings plan and so on. This is an application that could fit into a large organization, or with very minor modifications could be used by a single person to record banking transactions.

The book builds on this idea and develops the example, demonstrating how to take an idea and formulate it into a design with the correct architecture. It should be said, though, that the example will be the bare minimum to make it run, as I don't want to detract from SQL Server. The book will give you the power and the knowledge to take this example, expand it to suit your financial application needs, and give it the specifics and intricacies that are required to make it fully useful for yourself.

But before we can get to this point, we need to install SQL Server.

Installation

The remainder of this chapter will guide you through the installation process of the Developer Edition, although virtually all that you see will be in every edition. Some of the differences will be due to the

functionality of each edition. Microsoft offers a 120-day trial version at `http://www.microsoft.com/sql/evaluation/trial/`, which you can use to follow along with the examples in this book if you don't already have SQL Server 2008.

This book will cover many of the options and combinations of features that can be included within an installation. A number of different tools are supplied with SQL Server to be included with the installation. We will look at these tools so that a basic understanding of what they are will allow us to decide which to install.

Installation covers a great many different areas:

- Security issues
- Different types of installation—whether this is the first installation and instance of SQL Server or a subsequent instance, for development, test, or production
- Custom installations
- Installing only some of the products available

Most of these areas will be covered so that by the end of the chapter, you can feel confident and knowledgeable to complete any subsequent installations that suit your needs.

This book uses the Developer Edition because it is most suited to our needs, as developers, for it doesn't have all the operating system requirements of the Enterprise Edition. Insert the CD for the Microsoft SQL Server 2008 edition of your choice in your CD-ROM drive and start the installation. What the upcoming text covers is a standard installation.

Beginning the Install

First of all, ensure that you have logged on to your machine with administrative rights so that you are allowed to create files and folders on your machine, which is obviously required for installation to be successful.

If you are using a CD-ROM and the installation process does not automatically start, open up Windows Explorer and double-click autorun.exe, found at the root level of the CD-ROM. If you are not using a CD-ROM, double-click the installer executable that you downloaded.

You may now be presented with the installation screen for Microsoft .NET 3.5 Framework if it is not already installed. .NET is a framework that Microsoft created that allows programs written in VB .NET, C#, and other programming languages to have a common compile set for computers. SQL Server 2008 uses .NET for some of its own internal work, but also, as a developer, you can write .NET code in any of Microsoft's .NET languages and include this within SQL Server. With SQL Server 2008, there is also the ability to query the database using .NET and LINQ rather than T-SQL.

■ **Note** Including .NET code is an advanced topic and outside the scope of this book. For more information, try *Pro SQL Server 2005 Assemblies* by Robin Dewson and Julian Skinner (Apress, 2005).

Once this is installed, you are presented with the SQL Server Installation Center. This screen, shown in Figure 1-1,deals with planning an installation, setup processes, including new installations, upgrades from previous versions of SQL Server, and many other options for maintaining SQL Server installations.

When you click on the Installation item on the left of the Installation Center, you can then select from the first item of the list of Installation options, New SQL Server stand-alone installation or add features to an existing installation, and the SQL Server 2008 installation starts.

A quick system check is performed before you enter your product key and accept the license terms of SQL Server. There are a number of support files that SQL Server uses as part of the installation process, as well as to ensure that a clean and valid installation occurs. In Figure 1-2, you will see that there is one warning in the process, but it is still possible to proceed. Provided no errors are listed in your process, click Next.

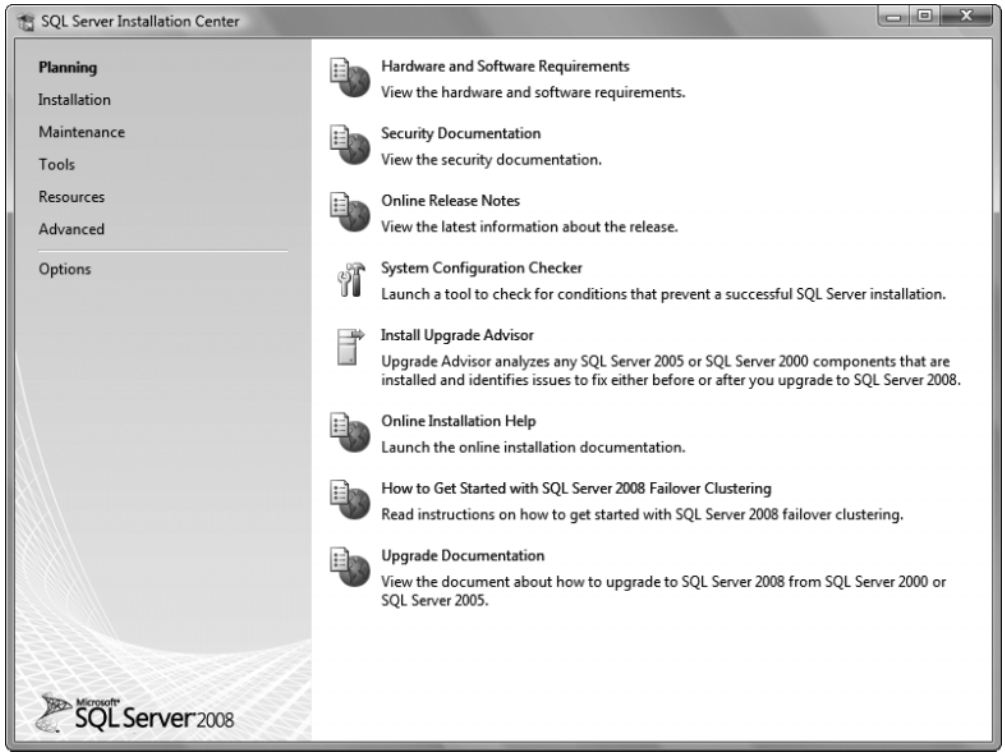

Figure 1-1. *Beginning the install with the Installation Center*

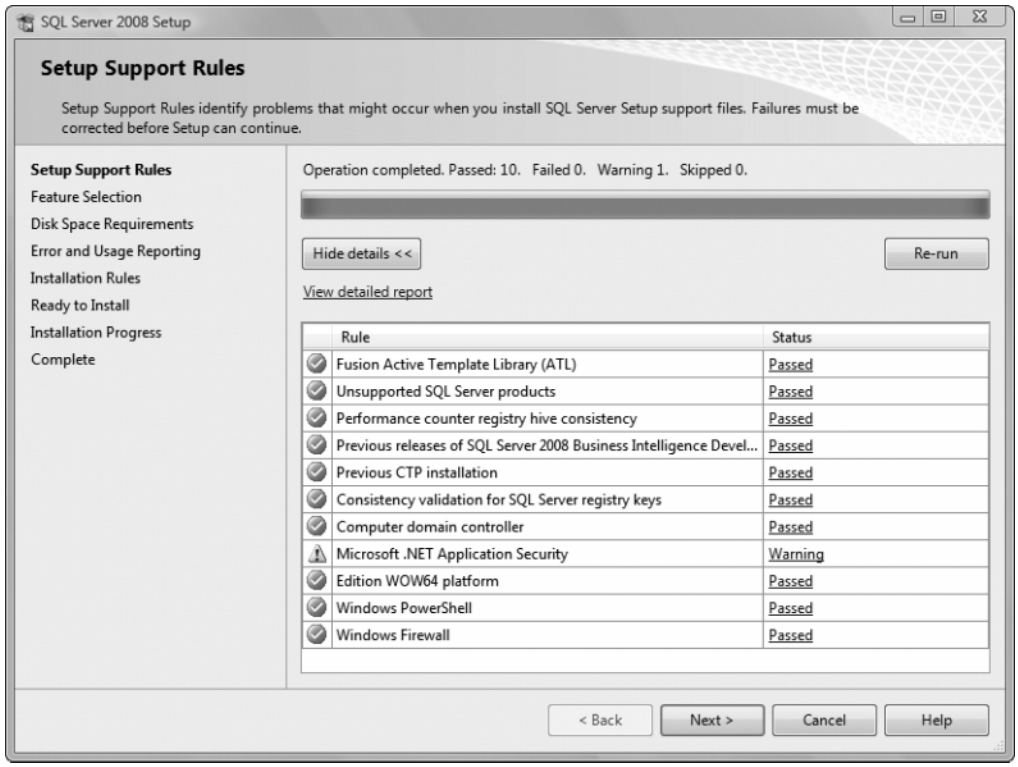

Figure 1-2. *System configuration checks*

Choosing the Features to Install

We now come to the Feature Selection screen, where we have to make some decisions. As you see in Figure 1-3, this installation will have everything installed, because this will be your development instance where you'll be testing every aspect of SQL Server away from any development of projects taking place. This is therefore going to be more of a training environment. However, you can be selective regarding what parts of the components you want to install. For this book, you will need Database Engine Services, Reporting Services, Client Tools, and Business Intelligence Development Studio for building reports, so ensure at least that these are checked.

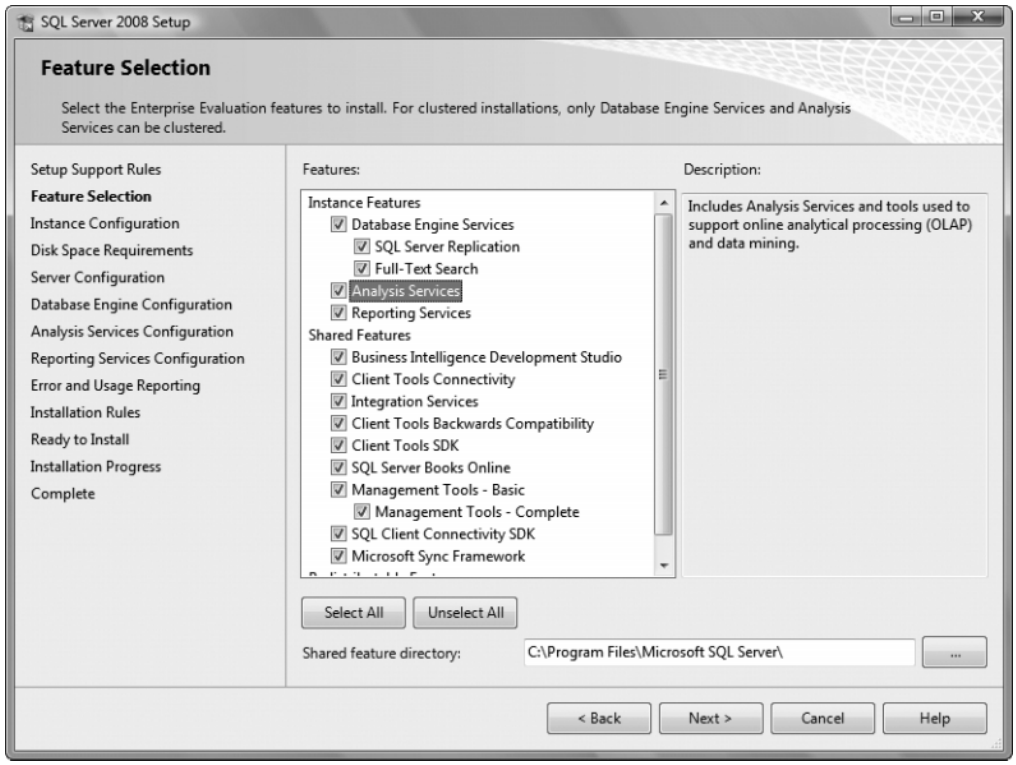

Figure 1-3. *Selecting every component to install*

Let's briefly take a look at most of these components:

- *Database Engine Services*: This is the main core for SQL Server 2008 and installs the main engine, data files, etc., to make SQL Server run.
  - *SQL Server Replication:* When you want to send data changes not only on the database it is being executed on, but also on a similar database that has been built to duplicate those changes, then you can use this option to replicate the changes to that database.
  - *Full text search:* This option lets you allow searching of the text within your database.
- *Analysis Services*: Using this tool, you can take a set of data and dice and slice and analyze the information contained.

- *Reporting Services*: This allows reports to be produced from SQL Server instead of using third-party tools such as Crystal Reports. We look at this component in more detail in Chapter 14.
- *Client Tools:* Some of these tools sit on the client machine and provide the graphical interface to SQL Server, while others sit on the client and work with SQL Server. This is the option you would package up for rollout to developers to sit on their machines.
- *Microsoft Sync Framework:* When working with offline applications such as those on mobile devices, there has to be some sort of synchronization mechanism in place. This option allows these interactions to occur.
- *SQL Server Books online:* This is the help system. If you need more information, clarification, or extra detail on any area of SQL Server, then this is the area to turn to.
- *Business Intelligence Development Studio:* When you want to analyze data using analysis-based services, then you can use this GUI to interact with your database. This book doesn't cover this option.
- *Integration Services:* This final option allows you to build processes to complete actions, such as importing data from other data sources and manipulating the data. You will see Integration Services in action in Chapter 7 when we look at building a backup maintenance plan.

Of these components, Analysis Services and Business Intelligence Development Studio fall outside the scope of this book, and we only touch upon Integration Services as mentioned.

■ **Note** At this point, SQL Server no longer has the option to install the sample databases. Microsoft is also altering the way sample databases and samples are delivered, so you may find newer versions on the SQL Server web site, `http://www.microsoft.com/sql`, or at `http://www.codeplex.com/SqlServerSamples`.

Naming the Instance

As you know, SQL Server is installed on a computer. It is possible to install SQL Server more than once on one computer. This could happen when you have a powerful server and it has enough resources such as memory and processor to cope with two or three different applications running. These different applications want to have their own SQL Server databases. Each install is called an **instance**. We are now at the stage that we can name the instance of the install. Each instance must have a unique name attached to it, although "no name," known as a **default instance**, is also classified as a unique name.

Naming instances is important as the first step to organizing your environments. For example, you may have an instance for development, another instance for system testing, and finally one for user testing. It is bad practice to share production server hardware with anything but production databases. If you don't, and if you have a rogue development action that occurs and crashes the server, you will stop production from running. Although you would have to make this decision at the start of the install process you are currently going through, it is a useful second reminder here when naming the instance.

The default instance is available, which is what is selected when you are not giving the install a specific name; once you install SQL Server outside of a learning environment, you should avoid this, as it gives you an installation with no name and therefore no hint as to its use. As you are still learning, the easiest option to understand is to use the default instance, so select Default Instance as shown in Figure 1-4 and then click Next.Once you have instances installed on the server, they will be listed here. You can also see the path detailed for each of the directories for the three services selected in the previous step.

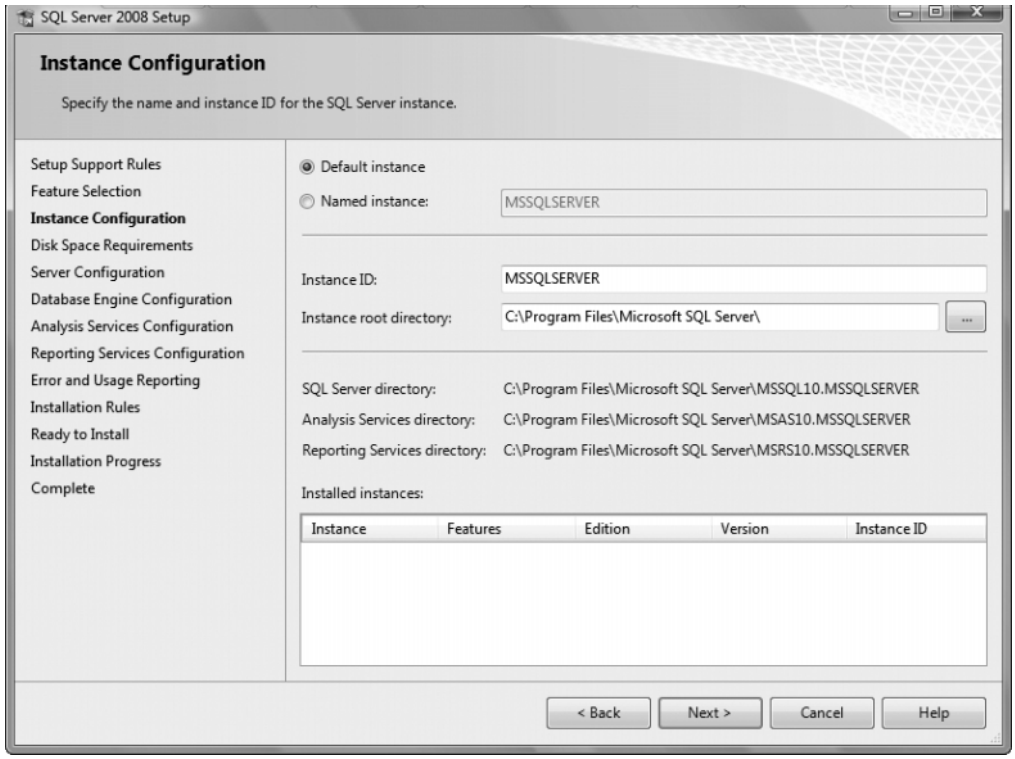

Figure 1-4. *Naming the install instance*

Choosing Service Accounts

SQL Server and other services, as defined in the Feature Selection screen (shown earlier in Figure 1-3), require you to log on to Windows before starting, just as you need to log on to Windows before using the system. SQL Server, Reporting Services, and so on can run without you or anyone being logged in to the computer that the install took place on. They can run just so long as the computer has fired up successfully. This is normal when SQL Server is installed on a server that is held in a remote location like a server room.

We will look at these options in more detail toward the end of the chapter. The options you see in Figure 1-5 will install SQL Server with a low-level set of privileges.

You can always change these later via the Services icon within the Control Panel. However, it would be much better to use the SQL Server Configuration Manager, found in Configuration Tools. By using the Configuration Manager, the account will be added to the correct group and be given the right permissions. Click Next.

If you look at the SQL Server Browser, which is another name for SQL Server Management Studio, it will be disabled by default. Many SQL Server installations will be on servers, quite often on remote servers; therefore, there should be no need to have the SQL Server Browser running. Normally, you will connect to SQL Server from a client machine. However, I am making the assumption that this installation is not on a server and is on a local computer, so change this option to start automatically.

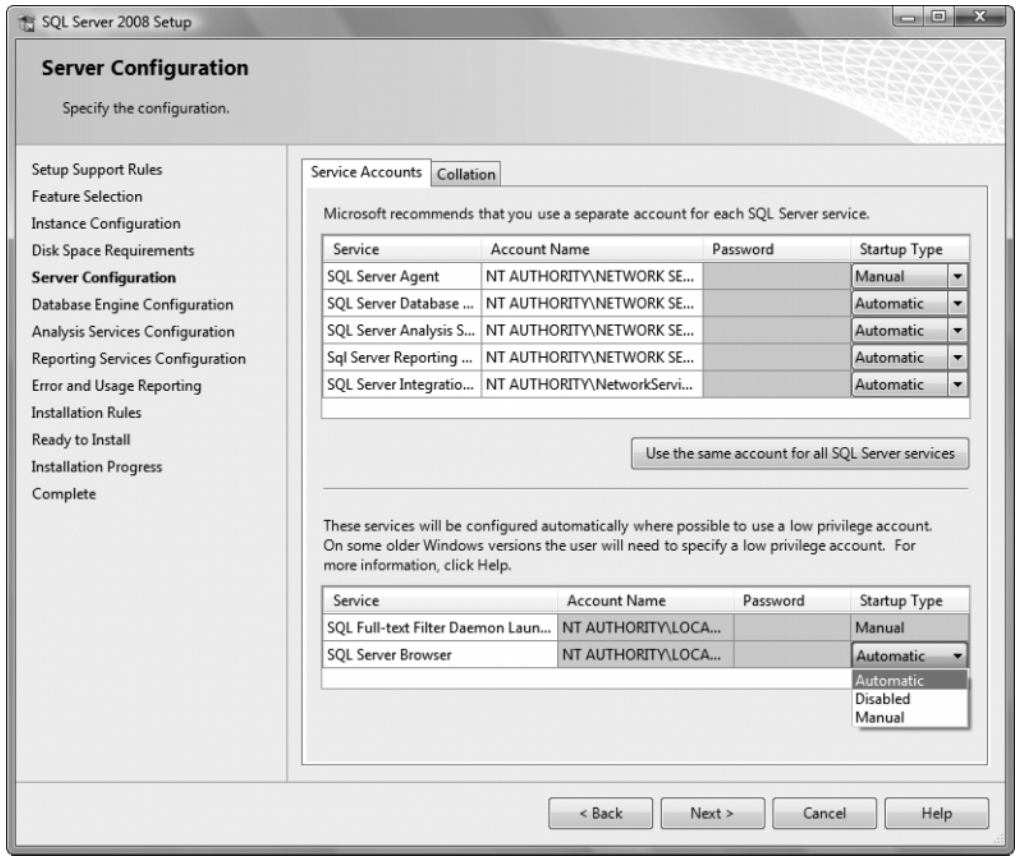

Figure 1-5. *Service account selection*

Selecting an Authentication Mode

We now come to how we want to enforce security on your SQL Server installation. As Figure 1-6 shows, there are two choices: Windows authentication mode and mixed mode. You will learn more about modes later in the chapter but very, very simply, Windows authentication mode denotes that you are going to use Windows security to maintain your SQL Server logins, whereas mixed mode uses either Windows security or a SQL Server defined login ID and password. We also need to define a password for a special login ID, called `sa`, if you are working with mixed mode. Again, you will learn more about this in a moment, but for now you must enter a valid password. Use a meaningful and impossible-to-guess password, but not one that you will forget.

It is also necessary to define a SQL Server Administrator account. This is a special account that you can use to log on in dire emergencies, such as when SQL Server refuses connections. This special account allows you to log on, debug the situation, and get SQL Server back up and running. Normally, this Administrator account would be a server account ID, but for now, use the account you have used to log on to your computer.

You will also see a similar screen for Analysis Services, and the settings are the same.