Microsoft SQL Server Black Book - Table of Contents



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

**Introduction**

**What's on the CD-ROM**

**Dedication**

**Chapter 1—Preinstallation Considerations What Is A Device?**

**What, Then, Is A Database?**

**What Are Character Sets And Sort Orders?**

**What Is The Recommended System Configuration? Where Should The Microsoft SQL Server Be Installed? What’s Stored In The Master Database?**

**The Master Database**

**The Pubs Database**

**The Model Database**

**Tempdb**

**The Msdb Database**

**Be Careful With Memory**

**What Security Model Will Be Used?**

**Spring Cleaning**

**Protocols**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (1 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Services**

**What About The SQL Mail Client?**

**Should I Use The Default Location For My Devices?**

**What Hardware Should I Install Microsoft SQL Server On?**

**Finally**

**Summary**

**Practical Guide to Preinstallation**

**The Preinstallation Checklist**

**Chapter 2—Installing Microsoft SQL Server**

**Installing A Production Data Server**

**SQL Server A (The Base Level)**

**SQL Server B (Middle Of The Road)**

**SQL Server C (The High-End Beast)**

**Before Installing SQL Server**

**Creating Your SQLExec Account**

**One More Account To Go**

**Setting Up A Mail Client**

**One Last Time**

**Summary**

**Practical Guide to Installation**

**SQL Server Installation From Start To Finish**

**Post-Installation Issues**

**Chapter 3—Development Versus Production**

**Setting Up The Development Environment**

**Third-Party Tools**

**Data Modeling**

**Server-Level Parameters**

**User Connections**

**Tempdb in RAM**

**Sort Pages**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (2 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Resource Timeout**

**Read-Ahead Optimization Priority Boost**

**Max Worker Threads Lock Escalation Parameters Fill Factor**

**Application Parameters**

**DBCC PINTABLE**

**Registry-Type Tables**

**List-Type Tables**

**Setup Scripts**

**Sample Server Setup Script**

**Scripting Objects**

**Third-Party Management**

**Transferring Objects**

**Transferring Data**

**BCP**

**INSERT/SELECT**

**DBArtisan**

**Permissions**

**Users And Groups**

**Summary**

**Practical Guide to Transferring Objects**

**Registering Servers**

**Moving Objects From Server To Server**

**Warning Messages**

**Pitfalls**

**Chapter 4—Replication**

**Data Distribution Models**

**Two-Phase Commit**

**Replication Consistency**

**Terminology**

**Publisher**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (3 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Subscriber**

**Distribution Server**

**Transaction Log**

**Synchronization**

**Horizontal Partitions**

**Vertical Partitions**

**Articles**

**Publications**

**Push**

**Pull**

**Server Roles**

**Publisher Server**

**Subscriber Server**

**Distribution Server**

**Scenarios**

**Considerations**

**Central Publisher**

**Central Publisher With Remote Distribution**

**Publishing Subscriber**

**Central Subscriber**

**Multiple Publishers Of A Single Table**

**Events And Processes**

**Log Reader Process**

**Synchronization Process**

**Replication Distribution Process**

**Communication Failures**

**Prerequisites For Replication**

**Memory**

**Working Directory**

**Same Character Set**

**Protocol**

**Trusts**

**Disk Space**

**SQL Executive**

**User Connections**

**Primary Key**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (4 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Summary**

**Practical Guide to Replication**

**Installing The Distribution Database**

**Setting Publication Options**

**Creating Publications And Articles**

**Setting Subscription Options**

**Subscribing To A Publication**

**Chapter 5—Structured Query Language (SQL)**

**ANSI-Compliant SQL**

**Syntax**

**Comments**

**Pubs Database**

**Authors**

**Sales**

**Titleauthor**

**SELECT Statements**

**WHERE Clause**

**ORDER BY Clause**

**GROUP BY Clause**

**Join Conditions**

**Aliases**

**Aggregates And Functions**

**SUM()**

**MAX()**

**MIN()**

**AVG()**

**COUNT()**

**CONVERT()**

**GETDATE()**

**DATEDIFF()**

**DATEPART()**

**SOUNDEX()**

**SUBSTRING()**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (5 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**UPPER()**

**CHARINDEX()**

**RTRIM()**

**System Functions**

**ISNULL()**

**USER\_ID()**

**USER\_NAME()**

**DATALENGTH()**

**COL\_LENGTH()**

**Calculated Values**

**Optimizer Hints**

**Subqueries**

**Union**

**INSERT Statements**

**Identity Columns**

**Stored Procedures**

**Triggers**

**UPDATE Statements**

**DELETE Statements**

**Batches**

**Cursors**

**Summary**

**Practical Guide To SQL**

**Schema Changes**

**Backing Up Data**

**Renaming Objects To Be Modified**

**Scripting Objects**

**Converting And Inserting Old Data**

**Cleaning Up The Environment**

**A Word On Constraints**

**Chapter 6—Stored Procedures**

**Consistent Data Manipulation**

**Enter Stored Procedures**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (6 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Establishing Standards**

**Getting Data**

**Modifying Data**

**Modular Programming**

**Reduced Client Processing**

**Network Traffic**

**Calling A Stored Procedure**

**Query Optimizer**

**Query Plan**

**Parameters**

**Variables**

**NT Server Registry**

**Maintenance**

**Return Codes**

**Additional Rules**

**Nesting And Recursion**

**System Stored Procedures**

**Custom Stored Procedures**

**External Stored Procedures**

**Remote Stored Procedures**

**Startup Stored Procedures**

**Prior To Production**

**Summary**

**Practical Guide to Stored Procedures**

**Parsing A String**

**Redundant Code**

**Reduced Network Traffic**

**Calling Procedures Within Procedures**

**Chapter 7—Views**

**Syntax For Creating Views**

**Normalized Data**

**Partitioned Data**

**Vertical Partitions**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (7 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Horizontal Partitions**

**Multiple Tables**

**Computed Values**

**Security**

**Updates**

**Underlying Objects**

**Performance**

**Restrictions**

**Summary**

**Practical Guide to Views**

**Determining Column Needs**

**Partitioning And Combining Data**

**Checking Index Coverage**

**Modifications**

**Chapter 8—Triggers**

**Data Integrity**

**Syntax**

**Business Rules**

**Permissions**

**Nesting**

**More On Triggers**

**Virtual Tables**

**Inserted Tables**

**Deleted Tables**

**Virtual Table Usage**

**Global Variables**

**INSERT Triggers**

**UPDATE Triggers**

**DELETE Triggers**

**Limitations**

**Multiple-Row Considerations**

**Performance**

**Summary**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (8 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Practical Guide to Triggers**

**Remove The Foreign Key Constraints**

**Define The Business Rule**

**Identify The Child Records**

**Graphically Represent The Trigger Firing Order**

**Write A Test Script**

**Check The titleAuthor Table**

**Create The Trigger**

**Test The Trigger**

**Chapter 9—Rules, Defaults, Constraints, And**

**User-Defined Data Types**

**Rules**

**Creating Rules**

**Binding Rules**

**Changing Rules**

**Dropping Rules**

**Defaults**

**Creating Defaults**

**Binding Defaults**

**Changing Defaults**

**Dropping Defaults**

**Constraints**

**Primary Key**

**Unique**

**Foreign Key**

**Default**

**Check**

**User-Defined Data Types**

**Entity Definition**

**Create Table Statement**

**Dependency**

**Summary**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (9 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Practical Guide to Rules, Defaults, Constraints, And**

**User-Defined Data Types**

**Creating The Scripts**

**Printing Out A UDT Listing**

**Building A Table Structure**

**Maintenance And Troubleshooting**

**Chapter 10—Error Codes**

**Errors In Microsoft SQL Server**

**Method Or Madness**

**Service Packs**

**Research**

**Summary**

**Practical Guide to Error Codes**

**Query/Connection-Based Errors**

**Server Configuration Errors**

**Connectivity Errors**

**Transaction Log Errors**

**Table And Index Errors**

**Chapter 11—Performance Tuning And**

**Optimization**

**What Is Performance?**

**Performance Monitor**

**Windows NT**

**Data Models**

**Application Design**

**Establish A Baseline**

**Keep It Simple**

**SQL Server Trace Flags**

**SQL Trace**

**SQL Probe**

**Summary**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (10 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**Practical Guide to Tuning And Optimization**

**Using The Performance Monitor**

**Using SQL Trace**

**Chapter 12—Newsgroups And The Internet**

**Accessing The Internet**

**Browsers**

**Functions**

**Search Engines**

**Knowledge Base**

**Service Packs And Patches**

**TechNet CD-ROM**

**Microsoft SQL Server Books Online**

**Newsgroups**

**Summary**

**Practical Guide to Free Agent Installation**

**Installing The Newsreader**

**Configuring The Source News Server**

**Subscribing To Newsgroups**

**Preferences**

**Newsgroup Etiquette**

**A Coriolis Group New Title—Bonus Chapter**

**Chapter 13—Using Java To Access Databases**

**The GuestBook Program**

**Initializing The Application**

**Handling Window Events**

**Opening A Database Connection**

**Closing The Database Connection**

**Executing A SQL Command**

**Handling Errors And Exceptions**

**Setting Up The ODBC Data Source**

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (11 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book - Table of Contents

**An Improved GuestBook Program**

**The init( ) Method**

**The execSQLCommand Method**

**The moveDataToForm( ) Method**

**The actionPerformed( ) Method**

**The destroy( ) Method**

**The WindowHandler Inner Class**

**Java And Access Data Types**

**Summary**

**Appendix A**

**Appendix B**

**Glossary**

**Index**

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ewtoc.html (12 of 12) [1/27/2000 6:15:06 PM]

Microsoft SQL Server Black Book:Introduction



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Table of Contents

**Introduction**

Welcome to the world of Microsoft SQL Server! Here is finally a client/server database product that can deliver world-class performance at a price that most enterprises can afford not only to purchase, but also to support. SQL Server’s

ease of use, coupled with the incredible feature set that accompanies it, delivers enterprise-level client/server computing to everyone. *Microsoft SQL Server Black Book* will focus on the tasks involved in harnessing Microsoft SQL Server’s capabilities to create a solid production data server. This book focuses on the current release of Microsoft SQL Server while using many techniques that can be applied as far back as version 4.21.

Writing *Microsoft SQL Server Black Book* has been the most challenging task I have undertaken in a long time. I was asked to write it to fill a void in the market, to deliver a book that focuses on creating production servers with hands-on, step-by-step processes for installing, configuring, and troubleshooting Microsoft SQL Server. I have tried to keep the language of the book as plain-English and matter-of-fact as possible, because that is the way I teach. I have supplied you with substantial technical background, while also supplying numerous examples. This book can be used as a tutorial or desktop reference to help you get Microsoft SQL Server to fulfill your organization’s needs.

I have been through many classes as a student and as an instructor—about topics ranging from adult learning principles to Microsoft SQL Server Administration. From these classes, I have acquired a great deal of knowledge that can be applied to creating a solid production data server with Microsoft

http://www.itknowledge.com/reference/standard/1576101495/index.html (1 of 4) [1/27/2000 6:15:08 PM]

Microsoft SQL Server Black Book:Introduction

SQL Server. I want to share that with you and help you head off the problems that you may encounter configuring your servers.

I am a business owner, a consultant, a DBA, and a teacher’s most likely are

many of you. I have fought and continue to fight the same battles that you do on a daily basis. That is why I think this book can be such a great value to you! Hopefully my experience will help you develop solid database systems in your Microsoft SQL Server environment.

Each chapter is broken into two sections. The first part of each chapter

presents explanatory material about the chapter topics. The second page of this first part is a blank Administrator’s Notes page, for you to write on and refer

back to later. The first part of the chapter ends with a Summary section, which is a bulleted list of the important points of the chapter. The second part of each chapter (the Practical Guide) supplies you with some step-by-step tasks that

reinforce the content of the chapter and provide hands-on practice. Chapters

1to3 cover the installation and configuration of Microsoft SQL Server for both development and production environments. Chapter 4 explains the setup and

terminology needed to implement replication between SQL servers. Chapters 5 to9 discuss the SQL language and the many objects that can be created for and utilized in client/server applications. Chapters 10, 11, and 12 cover the

troubleshooting and tuning skills you will need to support your system over

the long haul.

One of the points I emphasize in this book is that you can solve any technical problem you are facing with the tools available to you. What are those tools? How do you research answers to your questions? How do you know if you can trust the sources you consult? How do particular features really work, and will they work for you? I cover all these questions and more in the pages of this

book. I hope that you enjoy reading it this as much as I have writing it.

**Prerequisites**

This book is geared toward readers with a broad range of backgrounds. Many readers may have never worked with Microsoft SQL Server before, so I have tried to write a book that can transform a beginner into a power user. At the

same time, I have added plenty of advanced concepts and techniques to each

chapter that experienced DBAs can use to get your server running like a

thoroughbred.

The book assumes a basic understanding of Windows NT. The exercises and examples will run on any machine that can run the client utilities for Microsoft SQL Server.

**Technical Support**

If you find any errors or need further help with any topic in this book, you can reach me on the Internet through my email account: pDalton@msn.com.

Please do not hesitate to send me feedback—whether positive or

negative—concerning this book. If you have technical difficulties with the

CD-ROM, please contact Robert Clarfield at The Coriolis Group:

techsupport@coriolis.com. If you find a problem with one of the products on http://www.itknowledge.com/reference/standard/1576101495/index.html (2 of 4) [1/27/2000 6:15:08 PM]

Microsoft SQL Server Black Book:Introduction

the CD-ROM, contact the appropriate vendor through its Web site.

**Acknowledgments**

I would like to thank a few people that have been key in my personal and

professional life. Without each of these people I would not have been able to

write this book or to have the drive required to succeed in today’s

fast-changing technical environment.

First, I would like to thank my wife Diane. She has provided support day in

and day out for years now. She puts up with my technical side and still loves

me for the great big kid that I can be at times. I would also like to express my gratitude to my sister, Cathy, and her husband, Lee. They have both helped me through some troubled times of my life and have given me unwavering

support.

I would like to thank a few other individuals who have played significant roles in my life. In my early military career, I had the good fortune of being under

the tutelage of Sergeant First Class Howell. He was a Vietnam veteran with

great wisdom and patience and a strong work ethic; he has had a profound

influence on me. Later, at Cray Computer Corporation, I came to know Bob

Hoeglund, for whom I held and still hold a great deal of respect. He gave me

the opportunity to learn as much as I could about every aspect of databases and computers and how they work.

I would also like to thank all the students in my classes. Without your hard

questions and late evenings after class, I would not be the DBA that I am.

I would also like to thank Kenny Simms for his valuable assistance in this

endeavor. Kenny is a former student. He has contributed hours of

research—both on the Internet and in the Knowledge Base—to ensure the

quality of the chapters in this book.

I would also like to thank Paula Kmetz, Senior Project Editor at The Coriolis Group. Her understanding and patience is the only reason I have completed

this book. She has even put up with my sense of humor and comments

throughout the project. Her efforts, and those of others at The Coriolis Group are greatly appreciated.

Finally, I would like to thank my parents. First, my mother, for her support

when I was young and for all the sacrifices she made to ensure that I had what I needed for school and all those projects. I would not be the person I am today without the help and confidence my mother gave me. Over the last 15 years or so, my father and I have come to know each other as adults, and I value the

friendship we share. He has helped me come to understand what it is like to

sacrifice for the good of the ones you love.

Table of Contents

http://www.itknowledge.com/reference/standard/1576101495/index.html (3 of 4) [1/27/2000 6:15:08 PM]

Microsoft SQL Server Black Book:Introduction

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/index.html (4 of 4) [1/27/2000 6:15:08 PM]

Microsoft SQL Server Black Book:What's on the CD-ROM



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced 

Search

Search Tips



**Search this book:**

Table of Contents

**What’s on the CD-ROM**

The companion CD\_ROM contains the source code (SQL) used in each of the chapters of *Microsoft SQL Server Black Book.* The source code is in text-file format, which can be run on your server. Also included on the CD\_ROM are some example applications, written for Microsoft SQL Server and the Internet, that can be used to help increase your productivity. They include:

**•** SQL probe: An analytical tool that can be used to monitor the health of your Microsoft SQL server

**•** DBArtisan: A database management tool for managing many different SQL databases through a single interface

**•** Team SQL: Source code control software for SQL objects—a must for any medium-to-large MIS department

**•** ERStudio: A data modeling tool with a great user interface **•** Free Agent: A newsgroup reader with some very impressive features that make it a breeze to keep current in the newsgroups

**•** Acrobat reader: An application that allows you to view PDF files

See the files on the CD\_ROM for more information on each of the products. Each product is supplied directly from a third-party vendor and should be installed and configured based on the specific requirements stated in its resspective readme and installation files.

**Requirements**

http://www.itknowledge.com/reference/standard/1576101495/about.html (1 of 2) [1/27/2000 6:15:08 PM]

Microsoft SQL Server Black Book:What's on the CD-ROM

Some products are designed to run on Microsoft Windows NT and/or

Windows 95. See each application for the system and hardware requirements

recommended by the vendor. The SQL source code on the CD\_ROM is in

ASCII text and can be loaded into any query tool that supports ASCII text

files.

Table of Contents

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/about.html (2 of 2) [1/27/2000 6:15:08 PM]

Microsoft SQL Server Black Book:Dedication



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced 

Search

Search Tips



**Search this book:**

Table of Contents

**Dedication**

*This book is dedicated to my children and to the memory of Jeffery Simms.*

*To Jennifer, Jamie, and Sterling, with love and thanks for being my life. You can accomplish anything you set your minds to.*

Table of Contents

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/about\_author.html [1/27/2000 6:15:09 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Chapter 1**

**Preinstallation Considerations**

**The Preinstallation Checklist**

**•** The Windows NT Section

**•** Determining Memory Requirements For The Server

**•** Microsoft SQL Server Installation Issues

**Administrator’s Notes...**

Today’s database administrators face many challenges in setting up a data server, regardless of version or manufacturer. An often-difficult task is to get the installation to match the needs of the production environment. I like to look at as many things as possible prior to running Setup to ensure that the production machine is configured to the requirements of the user and the application load placed on the server.

Let’s look at some of the basics of Microsoft SQL Server. Understanding these basic components is important to the planning process. This process is valuable for a first-time installation of Microsoft SQL Server. If you are planning your first installation or reinstallation, the following pages might help clear the fog.

**What Is A Device?**

The terms *device* and *database* are often confused. The basic storage container

http://www.itknowledge.com/reference/standard/1576101495/ch01/001-004.html (1 of 3) [1/27/2000 6:15:11 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

for Microsoft SQL Server is a *device,* which is an operating system file that

resides on the physical disk, or hard drive, of the server. A device is the

container that allocates space to Microsoft SQL Server on the server’s hard

drive. Microsoft SQL Server does not acquire disk space on the server

dynamically. You must specify the amount of disk to set aside for it to use.

This allocation is accomplished through the device.

The space that you set aside for devices is essentially lost to the rest of the

machine. A device cannot be made smaller. You can, however, expand a

device to make more room for your databases to grow—provided you have

enough free disk space. You can—and will—have multiple devices on your

server. Databases can span multiple devices to accommodate their growth.

A device carries with it a file extension of .DAT. This is important to know if you are in a multiple-programmer environment and are using the data server

for file services as well as data services. For example, in File Manager or

Windows NT Explorer, note the physical file C:\MSSQL\Data\master.dat. You can highlight this file, hit the Delete key, and if it is not currently being used

by Microsoft SQL Server, it will be deleted like any other file. If it is in use,

Microsoft SQL Server and the operating system will not allow it to be deleted. This prevents an accidental delete.

The only acceptable way to recover the space given to a device is to drop the

device and re-create it with a smaller size. When you drop a device, ensure that you go to the file system and delete the physical file. If you do not remove the device file, you will receive an error message when you re-create the device

with the same name. Once you remove the file, you use the Enterprise

Manager to re-create the device with a smaller size. You can then restore any contents of the old device to the new device, provided all the objects fit in the new space.

Try to avoid creating one big device that takes up the whole hard drive. Doing so will not give you the flexibility you need from the server. You will be very limited in your options down the road and will have to jump through some

fairly complicated hoops to change this configuration on a production

machine.

From a slightly different perspective, a device can be thought of as a large,

empty office. This space is rented or leased by your company. If your business expands beyond the square footage you have set aside, you must acquire more office space to accommodate the growth. This can be achieved by expanding

the existing office into adjoining space in the same building or perhaps in

another office building altogether.

This growth scenario applies to your devices as well. I will use this and other analogies a lot throughout this book to help you associate Microsoft SQL

Server to a real-world example. Many of my students have found they

frequently do not remember the exact piece of information they need but can

draw on these analogies to figure out what to do.

Previous Table of Contents Next

http://www.itknowledge.com/reference/standard/1576101495/ch01/001-004.html (2 of 3) [1/27/2000 6:15:11 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/001-004.html (3 of 3) [1/27/2000 6:15:11 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**What, Then, Is A Database?**

Databases are also considered containers. They hold the objects that make up your server’s purpose in life. Tables, views, indexes, and stored procedures are all objects that reside in your database. You can, and often will, have multiple user-defined databases residing on your server. These databases are where the production information and code reside. Other databases are installed on your server to give it the intelligence it needs to function; I will cover these databases in a few different areas throughout the book. However, our focus will be on setting up a production system, not on the inner workings of Microsoft SQL Server.

One of the most common mistakes new users make is to confuse the device and the database. You place your databases *within* your devices. To understand this, think of a database as a division within your company. For instance, Human Resources deals with very specific kinds of information, so you would logically put all of that type of information in a container for centralized management and access control. Accounting is an area that often requires more security than others, and the information generated from this area would justly be placed in a separate container for security reasons. You would not scatter information for the Human Resources department throughout all the offices; instead, you would put all those functions and resources in one place. The same applies to databases and good database design.

An interesting point for all PC-based database programmers is that Microsoft SQL Server does not store the information or data in the database. Remember, the database is a container. Instead, the server stores your data in a table. The

http://www.itknowledge.com/reference/standard/1576101495/ch01/005-006.html (1 of 2) [1/27/2000 6:15:12 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

index you create for fast access to data is not stored in the table with the raw

data; it is stored as another object within the database. A database is a

collection of objects. This concept is not hard to follow, but it is different

enough from the organization of other database programs that it is sometimes a

stumbling block for the small-system programmer. An MIS department

accustomed to dBASE or Microsoft FoxPro databases will struggle with this at

first. Since this structure is common to most large database systems today, you

should become familiar with it.

In addition, you should focus on the database level when administrating your

system’s security. Your users will be granted a logon ID for connecting to the

server, but this does not allow them to get to the data they need. This is done

by adding users and groups to each database individually on a need-to-know

basis. This method of security keeps unwanted users from browsing where

they should not while allowing others to do their jobs.

Returning to the office analogy, let’s compare a database to the Accounting

department in your company. This department might have a door you must

pass through, and once you pass through that door, you would see all the

cubicles and desks where the actual work is done. This door might be locked

in the evening or even require a passkey to enter during the day.

The same idea can be applied to a database. The records and files are not

strewn around the office; they reside in filing cabinets and in folders or ledgers

for ease of access. These organizational tools can be related to Microsoft SQL

Server objects. You use tables, stored procedures, and indexes to find what

you need when you need it.

The security model that Microsoft SQL Server uses is also similar to the

passkey entry requirement. No one gets access without a valid key or

password. I will not try to recommend a security method here because of the

diverse requirements in the market today. However, I will say that Microsoft

SQL Server will accommodate a strict security model very well and still allow

for the simple, trusting models required by smaller companies growing into

Microsoft SQL Server.

During installation of Microsoft SQL Server, you will not be concerned with

these divisions or security, but you should make a few assumptions on the

amount of disk space you will need to accommodate these areas and how you

will accommodate these needs.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/005-006.html (2 of 2) [1/27/2000 6:15:12 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**What Are Character Sets And Sort Orders?**

Another preinstallation issue is choosing a character set and sort order. A *character set* is the basic text and symbols that are loaded in your system. Regardless of the character set you choose, the first 128 characters are the

same. The extended characters, including language-specific characters, reside in the remaining half of the character set. Your decision depends on whether you are doing business overseas or in other languages and need to store text and special characters. In most cases, the default is fine and should provide you with what you need to function.

You should make this determination prior to installation. Changing character sets can be a daunting task with many system ramifications. If your company is concerned about character sets, chances are you are experienced in these issues and this feature should be nothing new to you.

Another interesting issue concerns *sort orders*. Sort orders determine the way the data is organized when stored by Microsoft SQL Server. The default sort order for Microsoft SQL Server is dictionary order and case-insensitive. This

is fine and probably the best default setting. It is not, however, the fastest setting you can use on your system.

**Note:** Microsoft is not trying to slow you down. Most programmers are not as careful as they could be and do not always exercise consistent case sensitivity when they write code. The default for Microsoft SQL Server should be used if you have legacy systems that might contain this kind of SQL code.

http://www.itknowledge.com/reference/standard/1576101495/ch01/006-008.html (1 of 3) [1/27/2000 6:15:13 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

The fastest sort order is binary. The use of this setting has some impact on how you perform certain tasks down the road, so choose it carefully. It will change all of your SQL scripts, stored procedures, and client pass-through code to be case-sensitive. If you type a statement and use a different case than was

specified when the table was created, you will get an error message. Say, for

instance, you have a table called *MyTable* on your system. To access it, you

type “mytable”. An “Object Not Found” error is returned.

Binary sort order poses a few issues in developing client software, and great

care should be taken when using it. Your ad hoc queries might not return what you expect back from the server, either. A capital “F” does not equal a

lowercase “f”. Reports are not inherently smart enough to tell the difference,

and your code needs to allow for this.

If you store, access, and check for case sensitivity on your entire system,

binary is the way to go. I have configured two identical machines installed

from scratch with the same data sets stored in different sort orders. My tests

have proven that binary is faster for a lot of common operations. If you are

putting third-party applications on your server, make sure they run as expected in this sort order. If in doubt, call the vendor or technical support for the

product in question.

I often use binary sort orders as an example of a setting that restricts

programmers in a way they might find difficult. Because of the case-sensitive nature, programmers must write code with more care than they would

otherwise. The end result is faster, but getting there might be more difficult.

Users of the system should also be considered when selecting binary sort

orders. If a system allows for ad hoc reports or queries and a user does not

know that the data is stored with case sensitivity, he or she might not get the

expected results. This can be dangerous when converting legacy systems.

Make the decision to use a binary sort order only after carefully weighing the impact on your entire organization.

**Note:** Under “Performance Comparisons” in the Microsoft SQL Server

Books Online, select Topic 5 from the SQL Server 6.0 Setup Guide for more

information.

Another consideration in choosing a character set and a sort order is whether

you are setting up a distributed server environment. If you are, you must use

compatible character sets and sort orders among your servers. If you are going to share, replicate, or distribute data, use a common character set and sort order throughout your enterprise. Do not forget that in business today we must

occasionally share data with other companies. If your system interacts with

another company’s system, again make sure the character sets and sort orders are compatible.

Previous Table of Contents Next

http://www.itknowledge.com/reference/standard/1576101495/ch01/006-008.html (2 of 3) [1/27/2000 6:15:13 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/006-008.html (3 of 3) [1/27/2000 6:15:13 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**What Is The Recommended System Configuration?**

Let me first comment on the Microsoft recommendations for your system and what I have found to be a more realistic configuration for your server. Microsoft’s recommendations should be taken with a grain of salt and applied with care to each environment. Likewise, my recommendations—or anyone else’s, for that matter—should not be followed blindly. Recommendations are intended to give you an idea of where to start and should not be considered the end solution or setting for your system.

The system requirements for installing Microsoft SQL Server are actually very easy to meet, often leading the administrator into a false sense of security with regard to how well the server will perform. See Table 1.1 for system requirements.

**Table 1.1** Microsoft system requirements for an Intel-based system.

CPU 80486

RAM Minimum 16MB

Minimum 32MB required for replication

Hard Disk Minimum 60MB

Additional 15MB for Microsoft SQL Server Books

Online

File System FAT or NTFS

OS Windows NT Server 3.51 or higher

http://www.itknowledge.com/reference/standard/1576101495/ch01/008-009.html (1 of 3) [1/27/2000 6:15:14 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

The minimum CPU recommendation of an Intel-based 80486 is, in my

opinion, a poor choice for a production machine unless you have very few

users and are going to configure the server as a data server only. This type of machine should be considered a candidate for development environments or

servers with only the lightest of loads in a production situation. Given the low cost of Pentium-based processors, I would recommend no less than a

Pentium-class machine in a production environment. Taking into account the upgrade and support paths for these machines, even a Pentium Pro system is

well within the reach of just about any enterprise.

I am not suggesting you throw your current machine away or scrap your plans for a cheaper alternative. I know budgets and real-world requirements often do not allow a top-of-the-line machine for your project. The idea is to put your

best-performing machine where it will do the most good.

If you are using existing hardware for your data server, take a good inventory of what makes the target machine tick. Know the particulars of the disk access time and memory configuration. Benchmark the machines where possible to

get an idea of how well it is performing against others in the same class. You might find a less-expensive alternative to the planned configuration.

RAM is another highly performance-sensitive item that can make or break

your server. The minimum recommendation of 16MB is for a bare-bones

server that will perform on a limited basis as a data server. The 32MB

reference for a replication server is more in line with a minimum memory

configuration for a production server. In most production environments, server configurations range on average from 64MB RAM to 128MB RAM, with the occasional 256MB machine. On a high-volume multiuser system, servers with a greater amount of RAM would be much more efficient.

Do not forget the option of moving to a multiple-processor machine. Some

existing servers can be upgraded to multiple-processor configurations very

reasonably. Many unique situations require individual configuration

considerations, but adding RAM to a machine is the best first step in getting

better overall performance. The best rule of thumb is to look at the system load and determine if you need more RAM in your system. If your server is starved for RAM, you will know very quickly, and increasing the RAM is relatively

inexpensive.

Hard drives are an often-overlooked performance bottleneck on database

servers. Consider the performance of your disk controller and disk access

times to make sure you have not slowed your fast machine to a crawl with an older disk thrown in a new box. The axiom that you are only as fast as your

slowest link really applies here. I have seen administrators spend extraordinary amounts of time troubleshooting performance issues on data servers with older disks or 16-bit network interface cards. Be sure to look at all the pieces. No

piece in the chain of client-server communications should be overlooked.

Previous Table of Contents Next

http://www.itknowledge.com/reference/standard/1576101495/ch01/008-009.html (2 of 3) [1/27/2000 6:15:14 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/008-009.html (3 of 3) [1/27/2000 6:15:14 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Where Should The Microsoft SQL Server Be Installed?**

Keeping in mind that the optimum configuration is not always possible, I will describe what I think is the best place to install a Microsoft SQL Server on your network. In a strictly Microsoft network environment (which we all know is not very practical with the number of legacy systems out there), Microsoft talks of *domain structures*. While this book will not stray into domain configuration issues, there are some fundamental pieces of information that will apply whether you are setting up in a totally Microsoft environment or a NetWare/Microsoft mix. Your data server should be used solely as a server on the network. Try not to place additional services or processes on your data server, because they will add to overhead and slow the performance of the data services.

Primary domain controllers (PDCs) have the useful role of logging people on and off your Microsoft network. They also handle synchronization with backup domain controllers (BDCs) on your network. Any type of domain controller is not the optimal location to install Microsoft SQL Server.

Gateway Services for NetWare is another of the services you should consider moving off your Microsoft SQL Server. This service allows for NetWare files to be shared through Microsoft shares on your server. Although this is often a convenient way to get to your files, putting these files on your database server

adds to the overhead of that machine.

You should strive to install your server on as clean a machine as possible—one that will only be used for database services. This means that you should not set

http://www.itknowledge.com/reference/standard/1576101495/ch01/010-012.html (1 of 3) [1/27/2000 6:15:15 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

up Microsoft SQL Server on a primary or backup domain controller. Keep

shared file access off your database server. Having users copy files to and from the server will move the disk heads unnecessarily. Disk I/O is the slowest thing your data server will do. Do everything you can to keep it to a minimum. Also avoid sharing printers, modems, or like services on your Microsoft SQL

Server. All of these processes are burst performance-related loads; Murphy’s

Law will always ensure that one of the biggest file transfers or print jobs will hit your server at the same time a large query is running, causing the whole

system to appear to hang.

As you might be noticing, Microsoft appears to be moving toward a distributed server network. All the servers do not have to be on independent machines, but this configuration will help distribute the load across your network, allowing

you to put lighter-weight and lower-cost servers in place for mail and file

services and put your money where production is, such as on data services.

This distribution can be a good thing, but many companies fail to recognize

this until they have put all their eggs (applications, services, and files) in one

or two baskets (servers). Plan for growth. By definition, databases will grow

given even normal use. Over time any system that is being used in production will expand not only in feature and function, but in the amount of data as well.

If possible, place Microsoft SQL Server on a machine by itself. Install it as a

server that is part of a domain (provided you are using the Microsoft domain model). Place any other applications on separate machines when possible. If

multiple applications are running on the same machine, you are complicating the process unnecessarily. In addition, beware of disk-intensive applications

running on your database machine. If an application is writing to disk and

Microsoft SQL Server is writing to disk, these processes will compete for disk I/O and slow down both applications.

**Note:** The cost of adding a low-cost machine with a good-size disk to the

network versus the cost in performance by having all these services running

on the same box quickly becomes a non-issue.

Prior to installing Microsoft SQL Server, you should create a domain or local user account under which the SQL Executive service will perform its tasks.

This account setup is covered in detail in the next chapter, which includes

step-by-step installation on a few different machines.

**What’s Stored In The Master Database?**

The server’s system catalog and all the environmental information is stored in the master database, which is contained within the master device. The master database is the brains of your server. Great care should be taken when

modifying any information contained in the master database. You should get in the habit of backing up your master database whenever you make

environmental changes to your server, including changing the sizes of

databases or adding users. The following items should trigger a backup of the master database:

**•** CREATE, ALTER, or DROP statements (SQL)

http://www.itknowledge.com/reference/standard/1576101495/ch01/010-012.html (2 of 3) [1/27/2000 6:15:15 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

**•** DISK statements (SQL)

**•** Altering a transaction log

**•** Adding or removing a mirrored device

**•** Adding or dropping remote servers

**•** Adding or dropping a login ID

**•** Any change in server configuration

**Note:** Check bu\_list.doc (the reminder document) on the CD-ROM. Print it

out and put it in a clear, visible place to help keep your system in a good

recoverable state.

The size of the master device is another important consideration. By default in

current versions of Microsoft SQL Server, the master is set to 25MB. This

value is totally dependent on the system that it must support. Many things

affect the size of the master device. For most production systems, you must

alter the size of the master device when adding major components to the

server. Most end up in the 30MB range unless they need an abnormally large

Tempdb. Upon installation, I usually change this setting to 30MB to avoid

having to resize it a few weeks down the road. The additional 5MB of disk

space will not hurt the server and provides more flexibility right off the bat.

Keep in mind, however, that the size of the master device can be increased

after installation.

Having a good understanding of the master device and its components will

help you in later configuration and troubleshooting issues. By default

Microsoft SQL Server stores the master database, the Pubs database, the model

database, Tempdb, and the Msdb database in the master device. Let’s touch on

each one of these databases for just a moment to ensure a solid understanding

of their functions.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/010-012.html (3 of 3) [1/27/2000 6:15:15 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**The Master Database**

System tables and environmental information are stored in the master database. Tables such as Sysdatabases, Syslocks, Sysprocesses, and Sysusages store critical information about your server. Other tables, such as Sysobjects, keep track of the objects that reside in each database on your server; each database has a copy of these tables.

The server will allow you to edit these and other important tables through raw SQL; however, I strongly recommend that you do not modify data in any of the tables in the master through SQL commands. Such modifications should be attempted only when absolutely necessary and only by someone with an intimate understanding of Microsoft SQL Server. Plenty of tools are available in Microsoft SQL Server to protect you from yourself. Use these tools at your disposal to make server changes.

This is not to say that you cannot check these tables for information needed to run your client-server applications effectively. I have often used information in system tables to find certain server-side permission or relation information. You can read data all day long without making direct modifications to these tables. By default all users of a database will have some kind of permission to access the system tables for that database. This is a requirement for the system to run well and cannot be avoided.

To clarify, let’s look at this kind of information in a different light. You probably have committed to memory the layout of all the furniture in your house or apartment. If you woke up in the middle of the night and made a trip to the kitchen to get a drink of milk, you would probably make that trip fairly

http://www.itknowledge.com/reference/standard/1576101495/ch01/012-014.html (1 of 3) [1/27/2000 6:15:16 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

well even without the lights on. The system tables store the information you

take for granted, similar to the location and size of the coffee table, the doors, and so on. Incorrectly changing these stored values by hand would in effect

move the furniture on you. This would not lend itself to a good environment

for getting to your data. It could in some cases crash your server, rendering it useless.

**The Pubs Database**

In a production environment, Pubs does you no good and should probably be removed. This database is used as a learning tool and for testing the basics of your installation. Once your production machine is up and running, you can

remove this database from the master device.

**The Model Database**

The model database is like a stencil for creating new user-defined databases.

This stencil gives you a starting point for your CREATE DATABASE

statements. The system tables for user-defined databases are stored in the

model. Any stored procedures or users that need to exist in all your user

databases should be placed in the model database. By placing them in the

model, they will be copied to each successive database that is created. Be

careful when placing things in the model. This action will increase the

minimum size of your databases and may add unnecessary objects to

databases.

**Tempdb**

I often refer to Tempdb as a pad of Post-it notes: very small scratch paper that you use for a short period of time and then discard when you no longer need

the information on each piece of paper.

Many things can affect the space required for Tempdb. This database is part of the master device by default and resides on disk. This “scratch pad” is shared by all the users on the server for worktable space and to resolve join issues in processing your queries. If you have many users on your system, you might

need a bigger Tempdb. You might also need a bigger Tempdb if your users

have the ability to write their own ad hoc queries or reports, or if a query

returns a large number of rows.

So how big is big enough? This is a newsgroup topic in itself. You really have to look hard at what your server is going to handle and make your best guess. Following are some guidelines to optimize performance:

**•** Keep your queries under control.

**•** Limit the ad hoc report and free-formed query abilities against your

system.

**•** Use indexed columns for your join conditions and sorting needs

whenever possible.

**•** Watch all ORDER BY statements. This is covered in more detail

later.

http://www.itknowledge.com/reference/standard/1576101495/ch01/012-014.html (2 of 3) [1/27/2000 6:15:16 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

You can place Tempdb in RAM to get better performance out of your server.

This is not always the best move, and I will tell you why in Chapter 11,

“Tuning And Optimization.” If you are going to place Tempdb in RAM, install

it to disk and test the performance. Then move it to RAM and test it again. If

your tests show a good margin of improvement, then leave it in RAM.

Otherwise, change it back to disk. You’ll wish you had later when you start

seeing the “Can’t Allocate Space in Tempdb” error messages.

After you have installed your server, made setting changes, and established

some of the basic configuration options, back up your master database. You

might as well get used to it now and make a habit out of performing a backup

whenever you change the server configuration. It’s better to restore your

master from a backup than to reconstruct it from memory and any notes you

might have in a folder somewhere.

**The Msdb Database**

The Msdb database is perhaps the most versatile piece of your server. This is

basically your server’s to-do list. You can add tasks to this database that will

be performed on a scheduled recurring basis. You can also view the history of

the defined tasks and their execution results. The Msdb database is the

component that allows you to proactively manage your data server. Used

primarily by the SQL Executive service, the Msdb is created on two separate

devices: one for your data and one for the transaction log. We will cover these

pieces later in the book.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/012-014.html (3 of 3) [1/27/2000 6:15:16 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Be Careful With Memory**

Microsoft SQL Server should not be installed with a memory footprint larger than available memory. This configuration option can be set to more than the system has installed. (Microsoft SQL Server will try to start with that setting, too.) In some situations the server will start and run *very* slowly, and in others

it will appear to hang. You can fix this memory setting by starting the server with the -f switch. This starts the server in a basic configuration and then allows you to go in and change the memory setting. This memory setting is configured after installation. You should pick a setting, make the changes on your server, and then monitor the impact of that setting. Never assume that your math is correct or that what you heard someone else has done is right for your situation. Test it first.

To set or configure the memory for your server, do the following: **1.** Start the SQL Enterprise Manager.

**2.** Select the Server menu option.

**3.** Select the Configure menu option.

**4.** When the Server Configuration dialog box appears, select the Configure tab.

**5.** Scroll to the Memory option and modify the memory settings (see Table 1.2).

**Table 1.2** Memory recommendations for Microsoft SQL Server. **Hardware SQL Server 2K Setting**

http://www.itknowledge.com/reference/standard/1576101495/ch01/015-017.html (1 of 3) [1/27/2000 6:15:18 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

16MB 4MB (—) 2048 (—)

24MB 8MB (—) 4096 (—)

32MB 16MB (18) 8192 (9216)

48MB 28MB (34) 14336 (17408)

64MB 40MB (46) 20480 (23552)

128MB 100MB (108) 51200 (55296)

256MB 216MB (226) 110592 (115712)

512MB 464MB (472) 237568 (241664)

Memory settings do not take into account other processes running on the

server. My recommendations for memory are in parenthesis.

**6.** Make the changes based on your hardware configuration.

**7.** Stop and start the MSSQLServer service to let the changes take

effect.

This memory setting is in 2K units and can be a little confusing. You must

convert the MB value of the RAM on the machine to the equal number of

kilobytes. This is done by multiplying the MB value by 1024. Subtract the

amount of memory that Microsoft Windows NT needs (at least 12MB), then

divide that number by 2. This result will give the amount of memory in 2K

units that you should give Microsoft SQL Server, provided no other services or applications are running on the server.

Whenever possible, allow yourself a small threshold of extra memory. By

doing this you will not bring your server to a crawl by turning on a service or adding a process to your server without changing the memory setting for

Microsoft SQL Server. This can be as small as 2MB or as large as 20MB

depending on your hardware. See Table 1.2 for memory recommendations.

I do not recommend setting a Microsoft SQL Server up with less than 32MB

of RAM in a production environment. My settings reflect a system with only

minimal services running to allow for best performance. No other applications should be running on the database server.

**What Security Model Will Be Used?**

In Microsoft SQL Server, you have three choices when it comes to data

security: standard security setting (default), integrated security, and mixed

security. The *standard security* setting requires each user to supply a valid

login ID and password to attach to the server. This validation is separate from the network login scheme. This setting supports connections by non-Windows NT validated users accessing your data server.

*Integrated security* allows you to use the network login and password supplied to Microsoft Windows NT as a security mechanism for access to your data

server. If users are validated with a login and password by Microsoft Windows NT, they can connect to the server. This provides you with the one login, one password scenario that many companies are looking for. Keep in mind that just because a user can connect to the server does not mean he or she has access to your database.

http://www.itknowledge.com/reference/standard/1576101495/ch01/015-017.html (2 of 3) [1/27/2000 6:15:18 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

*Mixed security* is used when you want your Microsoft Windows NT users to

supply only one login and password to be on the network and connect to your

server. This method would also allow other network users to connect to the

server as long as they can provide a valid database login and password. In the

mixed-legacy environments of today’s businesses, this is a very popular

method of implementing security.

Microsoft SQL Server uses an interesting security model that has two levels of

security. First, you must be allowed to connect to the server. Then, for each

database you are granted access to, you are granted rights and permissions on a

case-by-case basis.

To explain this concept using our office example, say that you have been given

a key to get into the office building. This key gives you the right to enter and

walk through any hallways and public areas to function and find your way

around. Then, for access to certain areas in the building, you need an access

card (or permission) to get into each office or room (database) you do business

in. If you are not granted access to, say, the Human Resources department, you

simply cannot access this area (database). By assigning security on a

departmental level, you can give your users freedom to do their jobs while

protecting sensitive data from people who should not see it.

This model is very good for a few reasons. In a lower-budget design, you can

have both the production databases and training or development databases

coexist on the same server. You don’t have to worry about adding to an

existing system and having users gain rights by association to other databases.

Users are restricted by default and granted access by the owner of the database

to do what they need. No one except the SA (system administrator) has rights

in a database unless they own it.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/015-017.html (3 of 3) [1/27/2000 6:15:18 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Spring Cleaning**

As mentioned, in a production environment, Pubs does you no good and should probably be removed. Likewise, you should periodically look for things like Pubs within your system. The tables, or copies of tables, store procedures that are left to sit until no one knows any longer what they are or what they do. In over half the systems I have worked on (even in the one I am developing right now), I have made a copy of something and left the original in place, changed my copy until it was just the way I wanted it, and forgotten to remove the original or the test copy I ran to see if the system was faster. Keep your system as clean as possible and you will have less garbage to clean up later. Each object you define in your system takes up resources of some kind.

**Protocols**

I have read a few good white papers on benchmarks of protocols and which runs what type of operation best. You might have to support multiple protocols on your network. Keep in mind the default Named Pipes is slower than IPX/SPX or TCP/IP. You should try to use one of the latter two for client connections because they connect faster and transfer results better. Use as few protocols as necessary to reduce network traffic. I will not try to cover Windows NT tuning and optimization in this book; several good books are currently available that address this topic thoroughly. Microsoft SQL Server allows for multiple protocols to be supported and used simultaneously. Obviously, the number of protocols you are trying to support will have an impact on performance. Keep the list as small as possible, and you will be just fine. You can change your network support at any time after installation by

http://www.itknowledge.com/reference/standard/1576101495/ch01/017-019.html (1 of 3) [1/27/2000 6:15:19 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

rerunning Setup and selecting the Change Network Support radio button. See Table 1.3 for a list of the available protocols and the purposes they serve.

**Table 1.3** Microsoft SQL Server protocols (Net-Libraries).

Named Pipes SQL Server default protocol.

Multi-Protocol Required to use integrated security. Supports

encryption.

NWLink IPX/SPX Allows Novell IPX/SPX clients to connect.

TCP/IP Sockets Allows TCP/IP clients to connect. Uses port 1433.

Banyan VINES (Check SQL Books Online or Banyan documentation for configuration issues.)

AppleTalk ADSP Allows Apple Macintosh-based clients to connect.

DECnet Allows PATHWORKS connectivity. (Check SQL

Books Online or the DEC documentation.)

Microsoft SQL Server always listens on Named Pipes by default.

You may drop support for Named Pipes altogether. Before doing this,

however, make sure you have another protocol installed for client connections to your server. Also ensure that the client configuration utility is installed and returns the expected values on the server. All software that runs on your server runs as a client. Administrators often take this for granted and have the

perception that the Enterprise Manager, for example, is really the server. It is just a client and must connect like any other.

**Note:** You cannot change Net-Library configurations during upgrades. Any

existing configuration is carried over and can then be changed by rerunning

Setup.

**Services**

As mentioned, try not to have a lot of extra services running on your machine. Each of these services takes up processor time and resources. Administrators often forget that these services run all the time and automatically unless they

are changed. I will cover the services required for Microsoft SQL Server in

later chapters.

**What About The SQL Mail Client?**

Having your Microsoft SQL Server send you a mail message or report

automatically is a great feature. I have found this to be a tremendous benefit in setting up a new system. Microsoft SQL Server will interact with a number of mail clients through MAPI (Mail Application Programming Interface). Good step-by-step setup instructions are given in the Microsoft SQL Server Books

Online. Perform a search on Mail, and look up your particular mail system and how to configure it to run with Microsoft SQL Server. Do this early in the

process, and it will help keep you informed of just what your server is doing. Keep in mind, however, that too much of a good thing will slow processes

down. Making the call to the external stored procedure for mail does take time. http://www.itknowledge.com/reference/standard/1576101495/ch01/017-019.html (2 of 3) [1/27/2000 6:15:19 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

Use it with some thought.

**Should I Use The Default Location For My Devices?**

Whether to use the default location for devices depends on whether you have a

disk configuration that will better support a separate area for your data. In most

single-disk situations, the default directory is fine. If you are installing on a

machine with a multiple-disk subsystem or RAID system installed, then

putting the data files on high-performance disks will improve performance and

should be done at installation.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/017-019.html (3 of 3) [1/27/2000 6:15:19 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**What Hardware Should I Install Microsoft SQL Server On?**

This question comes up at least once in every class I have taught. I do not like to generalize all servers into one hardware configuration. But usually there are things I like to have on a server and things I do not.

Any server purchase you make today will be outdated by the time you unpack the box. Hardware changes on a daily basis, which is very frustrating. I like to buy servers with a good expansion path. A lot of potential expansion allows me to keep up with changes in the industry better. I buy brand-name servers because I don’t like to invest money in machines that have poor technical support and might not be supported next year. I *always* check the hardware compatibility list for Windows NT Server. This is a must. I check each component, from CPU to disk controller, when needed. This ensures that I will not have an operating-system problem with the server I am configuring.

I like to configure my servers with a RAID disk subsystem for my data. When reliable access to the data is critical, I require some sort of RAID configuration for the data to reside on. With the ability of Microsoft Windows NT to implement RAID at the operating-system level, this is easily accomplished with even a limited budget. I try to keep the operating system and program files separate from the data. I usually place these files on a separate disk and controller from the data, and I mirror the disk and controller when budget allows. This provides the maximum amount of protection from hard drive failures while keeping performance at the highest-possible levels (see Figure 1.1). The number of disks in the RAID array can be as small as three and as

http://www.itknowledge.com/reference/standard/1576101495/ch01/020-023.html (1 of 3) [1/27/2000 6:15:21 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

many as the disk subsystem will support.



**Figure 1.1** Sample hard drive configuration.

Not everyone can afford this type of configuration for their hardware. It is,

however, what I recommend for a fault-tolerant data server. If money is an

issue, cut the mirrored disk and controller out of the boot drives. The RAID

disk subsystem would be the last thing I would give up. Use the built-in,

software-driven, RAID option on Microsoft Windows NT servers only as a last resort. I would use that feature and set up my own fault-tolerant disk system

for storing my data only if no other option were available.

Given the many ways Microsoft SQL Server can write backups of databases to shared drives on other machines, a tape drive for backups is not required on

the data server as well. This can be a nice feature if you run around-the-clock operations and need to keep performance at high levels 24 hours a day.

Removing the backup load to another machine is in keeping with the

distributed-load concept becoming popular in many enterprises today.

A brief comment on backup strategy: Use the Dump and Load commands to

create backups of your databases, then back up the resulting backup device

file. Microsoft SQL Server uses the file system to store data, utilizing 2K

pages and a starting point and offset algorithm to access data. Your file backup software does not care about 2K pages, and this can cause problems for

Microsoft SQL Server. Your backup files can be damaged when you need

them to be stable and error-free. For example, a SCSI disk can use bad-sector remapping to fix a problem automatically, and this can confuse Microsoft SQL Server.

I always specify more RAM than I think I will need for the server. I like to

have at least 64MB of RAM for a production server. This gives plenty of space for Microsoft SQL Server and a good amount of data cache and procedure

cache for performance. You cannot go wrong by adding RAM. Even on a

80486 processor, giving it additional RAM will allow it to run very well.

Invest in a good network interface card—something with as much bus speed

and bandwidth as possible. If your data server were a fire engine, you would

not want to use a garden hose on it to put fires out.

I usually do not go overboard on the CD-ROM. Since I rarely use it for

production purposes, I try to keep this to whatever speed comes with the

server. The only time I look at a faster CD is when I plan to mount a database that resides on a CD-ROM for read-only purposes. In this situation, I look at

getting the fastest CD-ROM I can afford.

Do not get caught up in expensive monitors or sound cards. This is a place to save your money and buy some more RAM. Keep it simple. Video memory is

http://www.itknowledge.com/reference/standard/1576101495/ch01/020-023.html (2 of 3) [1/27/2000 6:15:21 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

not an issue on a data server. There is no reason for a sound card in a data

server unless you want it to say, “I am slowing down now. Please let your

users know. Never mind, they already know.” If you follow the

recommendations in this book, your server will be just a data server. This is

best for your users.

Verify you have network functionality before installing Microsoft SQL Server.

One of the hardest things to do is troubleshoot problems when you have no

idea what is working and what is not. Assume nothing. I have always used the

break-it-down-into-the-simplest-form approach in troubleshooting. If you

cannot get out on the network, no one will be able to connect to your server for

data.

Consider using a redundant power supply and surge protection for your unit.

Keeping the data available should be any DBA’s primary focus. Use a UPS

that is reliable and test it occasionally. A untested backup strategy is just that:

untested. If you think you’re under pressure now, wait until the UPS or backup

won’t restore and the system your boss has invested thousands of dollars in

does not work.

**Finally**

If you are setting up a system for a third-party application, beware of the

requirements for that system. Find out ahead of time what special

configurations will need to be set to let a particular piece of software run well.

You might find that a system configured to run one application well might not

allow another to run well at all. If this case arises, contact the vendor to

determine how you can adjust your server configuration to best provide for

both situations.

The old saying “hindsight is 20/20” is a testament to being prepared. You

cannot see all failures or issues ahead of time; we are all human and make

mistakes. By preparing yourself for these mistakes, you will know where to

look, how to look, and what action to take to solve the problems that arise. The

largest compliment I can get as an instructor is to have students call me or

email me with a problem, tell me what they think they should do to fix it, and

be right on target. This shows they are thinking things through and coming to

the right conclusions; they are just lacking the confidence (which will come

with time) to take the actions needed to fix the problem. The fact you are

reading this book shows you are making an effort to be prepared.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/020-023.html (3 of 3) [1/27/2000 6:15:21 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Summary**

The information in this chapter might seem like a lot to think about before you install the server software, but it will help you make better decisions about the type of machine you need. Following are the major points:

**•** Ideally, you should install Microsoft SQL Server on a standalone server that does nothing but serve data to clients. This is not always possible, but it should be the goal to work toward.

**•** More RAM is better. Use at least 32MB in a production server for best results. Upon installation, set your master database to 30MB to save time later.

**•** Be careful when choosing a character set and sort order. Faster is not always better.

**•** Use a RAID disk subsystem for your data when possible, with hardware-level RAID being the choice over software-level RAID on Windows NT.

**•** Use a separate drive for boot and system information, and mirror that drive with a separate controller for each drive. This disk configuration will give you a very flexible and fault-tolerant system.

**•** Configure your memory setting and check it to ensure it is correct. Remember that the memory option is in 2K increments.

**•** Use the Microsoft SQL Server Books Online often. Even though there are some known inconsistencies in them, they are a great resource that is underutilized by many professionals.

**•** You’ll need to increase the size of Tempdb. How much depends on

http://www.itknowledge.com/reference/standard/1576101495/ch01/023-027.html (1 of 4) [1/27/2000 6:15:23 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

your system and the way it will be used. Be cautious about putting

Tempdb in RAM. Test this before doing it; you might not gain as much

as you think.

**•** Back up your master database every time you change the server

environment.

**Practical Guide to Preinstallation**

*This section will walk you through gathering some preinstallation*

*information about your system that will help you install your*

*server right the first time.*

**The Preinstallation Checklist**

The preinstallation checklist (preinstall.doc) contained on the CD-ROM

accompanying this book can be printed out for your convenience. This

document will help you get through the installation process as painlessly as

possible. Let’s walk through this document step by step and fill it out for an

example server. I will choose an Intel-based Pentium 133MHz

single-processor machine with two hard drives—one with 1.2GB and the other with 4.3GB. Our machine will have two controllers, one for each hard drive,

and 64MB of RAM. It will run as a regular data server in a Microsoft

Windows NT TCP/IP network environment.

The preinstallation checklist is a memory-aid document. Each line of the

document has a comment or question that should be answered in whole or in

part before installing Microsoft SQL Server.

**The Windows NT Section**

The Windows NT section is intended to remind us of the operating system

issues that might come up during the installation and configuration of our

server. Check the Hardware Compatibility List (HCL) published by Microsoft for each component used in your server. If your hardware does not exist on the approved list, you should be careful; things might go wrong. However, just

because a piece of hardware is not on the list does not mean it will not work.

You should check whether NT will install and run properly. Keep in mind that Microsoft will not support hardware configurations that contain any

component not on the Hardware Compatibility List. This fact should be

enough to make you stay away from nonsupported components when running Windows NT.

Most of the issues in this section will become common sense once you have

set up a NT server and configured it for running SQL Server. This is intended to help those of you who might not be as familiar with Microsoft Windows NT as you would like. For more information on installation and configuration of

Microsoft Windows NT 4, see *Windows NT 4 Administrator’s Black Book*,

published by The Coriolis Group. The first area of the preinstallation checklist is displayed in Figure 1.2.

http://www.itknowledge.com/reference/standard/1576101495/ch01/023-027.html (2 of 4) [1/27/2000 6:15:23 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations



**Figure 1.2** Preinstallation checklist showing Windows NT issues.

Configuring a TCP/IP network is not covered here. What is covered in this

section are the key entries necessary for your server to exist on an IP-based

network.

Each machine on an IP network needs a unique address. This address is like a street address for your house. The mail carrier needs to know what address to deliver mail to—the same applies to your machine. The network must know

your address to deliver packets.

Then there is the default gateway entry. This entry is the primary exit point for your network. (To understand this concept, you can compare it to your local

post office knowing what other post offices to send packages to.) In a lot of

networks, this entry is reserved for the 123.123.123.1 address (the final “1”

being the machine, and the rest being address space). Putting the gateway at

address 1 makes life easier when you need to remember the address of the

gateway. There is no other significance to the “1” in the address.

Next is a WINS Server IP address entry. In a Microsoft network, a WINS

server provides Windows Internet Name Services or Resolution of your

machine name to an IP address. Internet names are the 123.123.123.4 address that the network knows you to be, and the Window’s name is the MY\_PC

name the users know your machine as.

Next is an address for a DHCP Server. This is not a required entry, but it is a valuable one from a network administration standpoint. The dynamic host

configuration protocol server leases addresses dynamically to machines on

your network that are DHCP-aware. This eliminates the need for static IP

addresses on a network and can be a great benefit in a large enterprise. Last in the list is an entry for the DNS server, or the domain name server. This server allows you to have a central list of names and IP addresses stored in a server

environment so that you can query this server for a name or IP address instead of maintaining a separate LMHOSTS file on each machine. Figure 1.3 shows the checklist with these settings filled in.



**Figure 1.3** Preinstallation checklist showing TCP/IP address settings.

Previous Table of Contents Next

http://www.itknowledge.com/reference/standard/1576101495/ch01/023-027.html (3 of 4) [1/27/2000 6:15:23 PM]

Microsoft SQL Server Black Book:Preinstallation Considerations

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch01/023-027.html (4 of 4) [1/27/2000 6:15:23 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Chapter 2**

**Installing Microsoft SQL Server**

**SQL Server Installation From Start To Finish**

**•** Creating The SQL Executive Service Account

**•** Creating The Microsoft SQL Server Service Account

**•** Granting Service Rights

**•** Installing Microsoft Exchange Client

**•** Running Setup

**•** A Quick Tour Of The SQL Server Program Group

**Post-Installation Issues**

**•** Configuring The Server Service

**•** Changing The Memory Setting To Reflect The Hardware **•** Testing SQL Mail

**Adminitrator’s Notes...**

**Installing A Production Data Server**

Because of the incredible number of different server configurations in today’s business community, I will use three different machines as examples in this chapter and show how each one can be configured. Choose the configuration that most closely matches your server and follow the information regarding

http://www.itknowledge.com/reference/standard/1576101495/ch02/033-037.html (1 of 3) [1/27/2000 6:15:24 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

that machine type.

This chapter is intended to be an installation tutorial. If you feel confident

installing Microsoft SQL Server, you might want to skip to the Summary

section for this chapter, located just before the Practical Guide. By reviewing the Summary, you can see if you need to go back into the chapter to review

any installation steps you might be unsure of.

**SQL Server A (The Base Level)**

This server is a good candidate for a small-enterprise production server or a

development server for a small to medium MIS department. Frequently such

departments have grown into a need for client-server databases. This type of

machine would also handle replication distribution tasks very well for a

medium-sized enterprise. (See Chapter 4 for an extensive discussion of

replication.)

One other point to consider for this type of server is using it in conjunction

with the Distributed Transaction Coordinator (DTC). With Microsoft’s

increasing focus on the distributed server environment, you should consider

smaller servers in your plans. These servers can become valuable in supporting your needs. Also, by separating processes across multiple servers, you can

include some legacy systems and save money.

With a little planning, even these smaller servers can be very useful in a

production environment. From an audit trail or tracking standpoint to a

centralized error-handling and task management scenario, these servers are

becoming a part of many enterprises. Learning how to integrate them into your plans will make you look very good in the eyes of any management group

trying to resolve growth issues at lower costs.

**Note:** You may want to look for a network task server application to help

manage your network resources and extend the life of your legacy hardware.

Base-level server configurations would typically involve an i486-based server, possibly with 48 to 96MB of RAM. Even a smaller amount of RAM should

not pose a real problem for most basic environments. As long as the target

system meets the minimum RAM requirements for Microsoft SQL Server, you will be fine. Keep in mind, however, that adding RAM to one of these servers can have a great impact on performance and should be one of the first things

you consider. Also included in this group are slower Pentium machines. These lower-speed Pentium servers will surely become commonplace in smaller

enterprises.

When specifying a server for your needs, always consider the environment and the clients your machine will be servicing. Most desktops and workstations

being purchased today are Pentium 100MHz machines with 32MB RAM. If

you have a dozen of these Pentium workstations picking on a poor little

486/66MHz server, your SQL server might not be able to keep up with the

requests for data during peak periods.

When choosing a smaller server, you should keep the amount of available

http://www.itknowledge.com/reference/standard/1576101495/ch02/033-037.html (2 of 3) [1/27/2000 6:15:24 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

RAM in the server as high as possible. Keep the additional tasks of file, print,

and gateway services off these types of servers. You may notice a slower

connection time with these lighter-weight servers. However, once you *have*

connected to one of these servers, you should get acceptable query response

times under a light load. This is usually exactly the machine a small office or

satellite location needs from a data server.

**Example Configuration For Server A**

Our example server in this class will be an Intel-based 486/100MHz processor

with 48MB RAM. Since this is an entry-level server, we will use a single hard

drive with 740MB of free space left after installing Windows NT. This server

will run Windows NT Server 4.0 (version 3.51 would do fine as well). The file

system for this server will be NTFS and not FAT for file-level security

reasons. This server will be running on a TCP/IP network with a PCI network

interface card and will run as a standalone server in a Microsoft Windows NT

domain. Take special care during installation of the operating system to make

the correct installation choices for the network. Accidentally installing NT

servers as domain controllers is easy to do and should be avoided.

**Note:** Avoid putting any additional overhead on this type of server. Starting

an application or service can place a burden on the functionality of the

operating system or Microsoft SQL Server and bring your data services to a

crawl.

We will take advantage of Microsoft SQL Server’s ability to write backup files

to network drives. This will negate the need for installing any type of tape

backup system. Since the system will back up data while users are in the

database, you should be aware of when these backups will occur. Scheduling

your backups around peak activity is important with a server of this size. By

using this backup strategy, you eliminate the need for a tape drive and the disk

and/or process overhead of the entire backup routine (with the exception of the

actual data dump). Chapter 11, “Tuning And Optimization,” discusses backup

strategies and how they can affect performance.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch02/033-037.html (3 of 3) [1/27/2000 6:15:24 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**SQL Server B (Middle Of The Road)**

This server is a good medium-duty server. Usually at this point, many companies begin to consider multiple-processor machines—even if they don’t yet actually purchase them. At this stage in a company’s growth, purchasing machines that support multiple processors as expansion options is a smart move. Take the time to ensure you can expand your server down the road. By definition, databases will always grow, and there are few things that you can do to prevent a good system from slowing down over time with increased use. Making good purchasing decisions is as important as hiring good people to fill critical positions in your company.

In addition, spending money on as current a processor architecture as possible at this level is a good idea. Take the time to look at what machine will supply the most processing bang for your buck. Check the results of independent tests for different machines. You’ll find that many show that Intel-based servers are

closing the performance gap with other architectures—and are less expensive and cheaper to support. I have had very good luck going with the established name brands. I also insist on good technical support.

The CD-ROM accompanying this book includes a few excellent white papers from Compaq and other sources. These white papers can be found on Compaq’s Web site at www.compaq.com or on any of the appropriate sites listed in the documents themselves. Regardless of the make of your server, these white papers do a very thorough job of helping you configure it to run well in a production environment. (In Chapter 12, you’ll find more on using online sources to answer your configuration questions.) There is a wealth of information out there for you to use—some of it is good and some is junk. You

http://www.itknowledge.com/reference/standard/1576101495/ch02/037-039.html (1 of 3) [1/27/2000 6:15:25 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

should take the time to learn what is and is not valid advice.

**Note:** Configuring data servers can be a difficult task. Adding one

nonconforming task or application can throw your performance into a

downward spiral. Always begin by breaking down the problem into

simplest-possible form. Then look for the obvious problem. Never assume

that something is working fine.

When I am asked to recommend a server, I try to get a good feel for how the

server is going to be used in the future. With this middle-of-the-road class of

server, you must start to look at fine-tuning your hardware choices for

maximum performance. It is common to have multiple drives in these servers; this should be considered, since multiple drives will increase your system’s

fault tolerance (an important goal of any database administrator). If you cannot afford to purchase all of the components for a fault-tolerant server at one time, you should plan their addition as budgets permit. In many scenarios in this

range, it is a good practice to propose during the bid process some sort of plan to upgrade or migrate the server hardware as the use and load of the server

matures. Management does not view this approach as overspending but as

good business planning. In most of the system consulting I have done over the last few years, I have found that including estimated costs for migrating and

improving the server over time actually helps “sell” the idea of purchasing

hardware and software to prospective clients. These estimates add value to

your recommendations, both short- and long-term, and they give decision

makers the realistic information they need to plan and budget IT resources.

**A Word On RAID**

A middle-of-the-road server should incorporate the ability to add disk space as painlessly as possible. Be leery of using the Microsoft Windows NT Server

software implementation of RAID as a solution for your disk subsystem needs. Although this is better than not having a fault-tolerant disk system at all, a

hardware-based solution is preferable. Using a hardware-based RAID system relieves your server from the burden of disk I/O while decreasing the amount of CPU processor time spent handling disk activities. Some of the better RAID systems have some very impressive throughput with high capacity and built-in caching. Some even support hot swap disks that, in the event of a failure, allow you to replace the bad drive in your system with an off-the-shelf disk without ever having to power down your server. Not all servers justify this kind of

expense, however. Use your best judgment and choose a disk subsystem with great care.

If you must use the software-level implementation of RAID, I suggest starting with a mirrored disk configuration. By choosing the mirror configuration, you will keep performance up as much as possible while maintaining some kind of tolerance. For more information on how to configure Microsoft Windows NT for the software implementation of RAID, see your Windows NT

documentation.

**Example Configuration For Server B**

http://www.itknowledge.com/reference/standard/1576101495/ch02/037-039.html (2 of 3) [1/27/2000 6:15:25 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

Let’s take a moment and look at an example server configuration in this class.

Typically these servers have up to 128MB RAM and are configured with

RAID disk subsystems. For our example, we’ll use a Pentium Pro 200MHz

single-processor server with 128MB RAM. We will again be using a good PCI

network card for maximum throughput and two disk drives. The first drive will

be for installing the operating system and program files; the second will be for

data. Due to costs, we are not going to go with a RAID system. Instead, I will

use a 1.2GB hard drive for the C: drive and a 4.3GB hard drive for the D:

drive. I will use two separate controllers for the hard drives for maximum

throughput of data.

This system will be a great candidate for adding a mirrored disk down the road

as funding becomes available for adding fault tolerance. Choose a good disk

controller that supports multiple disks or a system that will support the

addition of a disk and controller for establishing a mirror—or, in the case of

adding the extra controller, for duplexing the data drive.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch02/037-039.html (3 of 3) [1/27/2000 6:15:25 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**SQL Server C (The High-End Beast)**

Because of the prohibitive cost of a quad-processor or higher monster server, most small- to medium-sized companies do not purchase a high-end server. These machines, of course, do provide some incredible numbers when it comes to throughput and horsepower. Having recently rolled out a major system utilizing a Compaq Proliant 200MHz dual-processor Pentium Pro system at the heart of the data services, I am impressed with the performance of both Microsoft SQL Server and Windows NT on this platform.

Despite their high cost, multiple-processor machines are of great interest to the majority of students in my classes. Given the cost difference between these high-end Pentium machines and the mainframe/RISC machines, as well as the performance of Microsoft SQL Server, the multiple-processor machine is quickly becoming a hot topic. Many companies have purchased expandable machines and are looking for answers on how to take advantage of this architecture. As we explore topics later in this book, I will describe how efficiently these monster machines can run your queries.

**Example Configuration For Server C**

Although Microsoft SQL Server runs on other hardware (Alpha machines, for example), it must run on the Microsoft Windows NT operating system. I am focusing on Intel platforms because of their cost-to-performance advantages over other hardware platforms. These machines should be running with between 128MB and 256MB of RAM. Our example server in this class will use 256MB RAM. In addition, RAID level 5 subsystems are usually the order of the day for these machines. These servers usually have between 8 and 32GB

http://www.itknowledge.com/reference/standard/1576101495/ch02/039-041.html (1 of 3) [1/27/2000 6:15:27 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

of hard drive to work with. Redundant power supplies and a replacement drive on the shelf are musts when these systems go online.

These high-end servers are not something that the average programmer should jump into. Great care should be taken to configure these machines to suit their final environment. I recommend having a consultant handle the initial

hardware configuration for you if you have even the smallest doubt as to

whether you can do it on your own. It’s not that difficult, but this kind of

system is expensive enough that a mistake can cost you thousands of dollars to fix. Also, I have seen many servers that run at a much lower speed than they

should because of one incorrect setting or configuration option. The most

common question I get is: “Why is this fast machine running so slowly?”

**Before Installing SQL Server**

Let’s take a few minutes and go over in detail a few installation and

configuration issues before SQL is installed. The topics covered in this section can help you solve many of the problems associated with installing Microsoft SQL Server without any training or guidance. I have seen many newsgroup messages that could have been avoided had the reader followed these

suggestions.

**Creating Your SQLExec Account**

Before you install the server software, you should take a moment to create a

user account for the SQL Executive account to use once you have installed the server. This account is created with the User Manager for Domains. The name you choose is not as important as the permissions you give.

I usually choose a name that makes sense to me. “SQLExec” is used in

Microsoft’s training and course materials, so we’ll use that name here. This

domain account should be granted the Log On As A Service right so that it can get to the server when it needs to. Do not place any account restrictions on this login. Select a password in accordance with good security practices and make sure that the checkboxes for User Must Change Password and Account Never Expires are set properly. See Figure 2.1 on page 45 (in the Practical Guide

section of this chapter) for the proper settings.



**Figure 2.1** User Properties dialog box showing the SQL Executive account

settings.

**One More Account To Go**

While I am on the subject of creating user accounts, I usually create a user

account for the SQL Server service to use. It’s not a requirement for the server to use an account other than the local system, but there are some compelling

reasons to do so. Having the server run under the permissions of an account

http://www.itknowledge.com/reference/standard/1576101495/ch02/039-041.html (2 of 3) [1/27/2000 6:15:27 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

with a password is much more secure. In the event that some unauthorized user

gets access to your server, this extra security precaution can save you some

headaches. When you create an account for the server, make sure you grant the

same permissions as you did for the Executive service account.

This account can also be used for a mail client application. However, you

should create the account before attempting to configure the SQL Mail Client

or your server. In practically all mail systems, SQL Server must run under the

same account as the mail client you install and configure.

Changing the account that SQL server runs under is done through the Control

Panel under Services. Select the Services icon and when the window opens up,

double-click the MSSQLServer service. In the Services dialog box, click the

Startup button. In the bottom of the User Properties dialog box, fill in the

domain name, the account name, and the password, then select OK. See Figure

2.2 on page 46 in the Practical Guide for the User Properties dialog box.



**Figure 2.2** User Properties dialog box showing the SQL Server account

settings.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch02/039-041.html (3 of 3) [1/27/2000 6:15:27 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Setting Up A Mail Client**

If you are going to take advantage of the SQL Mail Client and email notifications on your SQL server, you should take the time to install and test the mail client for the mail system you are running before you install Microsoft SQL Server. Using email in your applications can provide you with a proactive management tool that many systems lack.

Using the **xp\_sendmail** external stored procedure in your triggers and user-defined stored procedures can be done very efficiently. Be aware that each mail system is configured slightly differently; you should consult the section on installing the specific mail client in SQL Books Online for any issues associated with your mail system. Microsoft SQL Server will send and receive mail from any MAPI-compliant mail system running on your network. Some additional overhead is involved with sending mail from inside your SQL code, so expect a slight delay in the execution of scripts or triggers that send mail. Since the use of email in your code has little to do with installation, I address this topic later in the book. Mail does, however, play a huge part in setting up a proactive server that will alert you to potential problems before they get out of hand.

Make sure that you create the accounts that mail must run under, and be sure to log in as those accounts when installing the mail client software. With Microsoft Exchange, for example, you need to set up a profile that matches the account that Microsoft SQL Server runs under as a service. Failing to do this will cause configuration problems when attempting to get mail features to work properly. I will walk through setting up a Microsoft Exchange client later in this chapter.

http://www.itknowledge.com/reference/standard/1576101495/ch02/041-046.html (1 of 3) [1/27/2000 6:15:28 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

**One Last Time**

I know that budgets and office politics come into play when deciding what

kind of server to buy or what upgrades are required to meet your needs. Take what I have said here with a grain of salt. Few scenarios allow the

best-possible design to be implemented. Fight only the battles you feel need to be fought.

Fortunately, Microsoft SQL Server is a very robust application. You can put an awful lot on a SQL server and make it do just about anything you can imagine. It runs well on every machine I have ever installed it on—from a 486/66 with 32MB of RAM to a huge multiple-processor beast. I have not found a bad

system yet—only ones that are improperly configured.

Regardless of the server you choose to install Microsoft SQL Server on, you

will be tempted to add a service or two to this server. Placing additional

services on any of these servers will cause changes in the way you should

configure Microsoft SQL Server. Most administrators will be tempted to place some extra files or network tasks on the higher-end machines I have talked

about here. Fight that urge!

**Summary**

**•** Take time to analyze your server requirements—both current and

future needs. Do not forget to consider distributed processing in your

plans.

**•** Check out the Man-O-War Task Server on the CD-ROM

accompanying this book. This application can be a valuable tool on your

network.

**•** Place as much RAM in your server as budget allows. The more the

better, up to the 128MB range. Insufficient memory limits the number of

users and lengthens query response times.

**•** Install your operating system and verify network connectivity before

installing Microsoft SQL Server.

**•** Create an account for the SQL Executive and the Microsoft SQL

Server service to run under. Creating these accounts will allow you to

implement better security on your servers and to easily configure email

services. These accounts should be granted the Log On As A Service

right with a good password assigned for security purposes.

**•** Install, configure, and test your mail client software prior to installing

Microsoft SQL Server. It is very important to ensure that mail can be

sent through your client software before assuming that the SQL mail

client can send mail. If you cannot send mail from the client, Microsoft

SQL Server will not be able to send mail.

**Practical Guide to Installation**

*This section presents step-by-step instructions for the installation*

http://www.itknowledge.com/reference/standard/1576101495/ch02/041-046.html (2 of 3) [1/27/2000 6:15:28 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

*and configuration of Microsoft SQL Server and its related*

*services.*

**SQL Server Installation From Start To Finish**

The first step in installing Microsoft SQL Server is to create the accounts

needed for the services themselves to run under. To install the product, it is not

required that this be done first, but I have found that most easy-to-miss service

configuration problems can be avoided by creating the correct user accounts

before installing any software. All three of our example servers would use the

same account information and would require the same information regardless

of hardware.

**Creating The SQL Executive Service Account**

To create an account for the SQL Executive service, follow these steps:

**1.** Log on to Windows NT as Administrator.

**2.** In the Administrative Tools program group, open the User Manager

For Domains.

**3.** Select New User from the User menu.

**4.** In the User Properties dialog box, input the information in the

following steps (see Figure 2.1). Make sure that you provide a secure

password for this account.

**5.** Clear the User Must Change Password at Next Logon checkbox.

**6.** Select the User Cannot Change Password checkbox.

**7.** Select the Password Never Expires checkbox.

**8.** Click the Groups button.

**9.** In the Group Membership dialog box, add this account to the

Administrators group.

**10.** Click OK to close this window. Click Add, then Close to complete

the user account.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch02/041-046.html (3 of 3) [1/27/2000 6:15:28 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**A Quick Tour Of The SQL Server Program Group**

Now let’s take a brief tour of the SQL Server program group, shown in Figure 2.22. The ISQL/W icon represents the graphical query tool that allows your query tables on the server. This tool is similar to many popular manual code entry query tools. ISQL/W has some nice features that I will discuss as we progress through writing queries and optimizing code later in the book.



**Figure 2.22** The Microsoft SQL Server program group.

The yellow question mark next to ISQL/W is a Help file that answers some questions and holds some key information about accessing Microsoft SQL Server with ODBC. I highly recommend this tool to anyone developing ODBC clients.

The next icon displayed is only available with version 6.5 and above. This is a RQBE tool that many of your users might be used to using. Microsoft Query uses ODBC to connect to the server and allows you to build queries graphically by clicking on the fields you want and selecting the order through its interface. This is a good tool for testing any ODBC connection since the server is accessed in this manner.

The readme file contains release notes for running some SQL files installed on

http://www.itknowledge.com/reference/standard/1576101495/ch02/060-063.html (1 of 3) [1/27/2000 6:15:30 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

your server to expand your management abilities. This file also contains some Distributed Transaction Coordinator notes that you should read before

configuring DTC services on your server.

The SQL Client Configuration utility is perhaps the most overlooked program you install. This utility should be run any time you have connection problems with the server. Remember that the Enterprise Manager is also a client

program, even though it runs on the same machine as the server itself. This is the first place I send people who call me and tell me they cannot connect to the server with a client or the Enterprise Manager.

The next yellow question mark is a Help file that explains the distributed

management objects (DMO) at the root of this system. These objects can be

used by client applications to perform many tasks without having to log on to the server as the administrator. I strongly recommend that client developers

become familiar with the DMO structure outlined in this file.

Next is the Enterprise Manager. This is a client utility that can be run on the

server or remotely to administer your entire enterprise. This tool does an

excellent job of showing you the remote management abilities of your servers (using the DMO). I will go into more detail on the Enterprise Manager as we progress through the book; for now, just note that this tool performs many

tasks that you can add to your client applications for increased management of your server.

The SQL Performance Monitor icon represents a standard set of Windows NT counters saved in a file that you can open to test the performance of not only

your SQL server, but the operating system and network as well. This tool

warrants a book unto itself; I will cover some of its uses in Chapter 11.

The SQL Security Manager is a small but powerful application that allows you to easily map NT user accounts to SQL server login IDs and databases. This

tool is a huge timesaver and should not be overlooked when configuring users and their ability to connect to the server.

The SQL Server Books Online icon is next, and second only to the Enterprise Manager, this is probably the most-used icon in the group. This calls a

TechNet CD-type search engine that holds all the information and

documentation that ships with Microsoft SQL Server. This resource will help you troubleshoot 85 percent of your problems without having to open a book (well, except for maybe this one).

In the bottom row is the SQL Server Web Assistant. This utility allows you to publish static HTML pages to your Web site. I have found the tool easy to use and fairly powerful when coupled with stored procedures. If you need dynamic data access to the Web, you will still need to use the IDC or ADC features of your Web server for interactive database query capabilities. This Web utility

gives you good basic reproducible results with little effort.

The SQL Service Manager is a utility that performs the same actions as the

Services icon in the Control Panel. This is an easy way to stop and start SQL

Server-related services.

http://www.itknowledge.com/reference/standard/1576101495/ch02/060-063.html (2 of 3) [1/27/2000 6:15:30 PM]

Microsoft SQL Server Black Book:Installing Microsoft SQL Server

The SQL Setup icon allows you to run the setup program while Microsoft SQL

Server is running. This allows you to change configuration options or network

support through the setup program. Keep in mind that some changes you make

in this program require you to restart the MSSQLServer service before they

take effect.

Last is the SQL Trace utility. This tool is crucial to monitoring the use of your

server and determining what raw SQL is being passed to your server by its

clients. I will walk you through this utility in Chapter 11.

**Post-Installation Issues**

To properly finish the installation process, you’ll need to check the following

items. If you have other SQL Server-based applications to install on this

server, you should check these items first before moving on to any other

software.

**Configuring The Server Service**

From the Control Panel, open the Services application and configure the

MSSQLServer service to run under the same Windows NT user account that

Microsoft Exchange is set to use. Start or restart the MSSQLServer service.

This change will not take effect until you restart the service. This particular

setting is considered to be static during the current session that Microsoft SQL

Server runs under.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch02/060-063.html (3 of 3) [1/27/2000 6:15:30 PM]

Microsoft SQL Server Black Book:Development Versus Production



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Chapter 3**

**Development Versus Production**

**•** Registering Servers

**•** Transferring Objects

**•** Warning Messages

**•** Pitfalls

**Administrator’s Notes...**

When management is faced with configuring a development environment, their latitude often depends on the size of the operation. Larger MIS departments usually have the resources and backing to provide a development server that mirrors the production machine fairly closely. Small- to mid-sized shops tend to have tighter budgets and fewer resources with which to accomplish this task. With Microsoft SQL Server’s lower cost of ownership, this does not have to be the case any longer.

Microsoft SQL Server gives you the freedom to be creative when approaching the development process. In planning and budgeting the development environment for Microsoft SQL Server, you have many freedoms that some other systems do not allow. You can place a test server into your environment at a very reasonable cost. By scaling your SQL server, you can simulate many production environments. Microsoft SQL Server lends itself to scaling very well.

http://www.itknowledge.com/reference/standard/1576101495/ch03/067-071.html (1 of 3) [1/27/2000 6:15:31 PM]

Microsoft SQL Server Black Book:Development Versus Production

I have found that even on inexpensive machines, Microsoft SQL Server can

play a very productive role in the development process. For some

environments, I have recommended having multiple SQL servers in the

development environment to lower the impact of the development process.

**Setting Up The Development Environment**

For many projects, having many programmers simultaneously developing

client applications to run against a server is common. In most cases, one or two database programmers can support the needs of even a mid-sized MIS

department. That is not to say that each project shouldn’t be considered on a case-by-case basis. I have found that using a multiple-server development

environment provides the greatest flexibility when creating stored procedures, triggers, and data modeling.

One benefit of using multiple servers is the isolation your SQL code

development can enjoy. Your database programmers can write and test code

against their development server, then copy tested and debugged code out to

the server being used by the client development team. I have used a

lightweight machine similar to the example Server A in the previous chapter to facilitate just such an environment.

In the event you cannot use this concept, you might want to install Microsoft

SQL Server on your own workstation. Providing you have enough RAM to

meet the minimum requirements and you install Microsoft Windows NT

Server as your operating system, you would then have a machine you can

develop on without impacting other programmers’ productivity.

This type of development is a sharp contrast to the databases running in the

UNIX environment. You are no longer forced to develop on the same machine as everyone else. One important note is to take additional care to ensure the

correct version of the objects you develop and deploy. With the Enterprise

Manager client provided with Microsoft SQL Server, you can effectively

migrate or transfer your fully debugged objects to other servers rather

painlessly.

**Third-Party Tools**

With the increased popularity of client/server development, many good

third-party tools have become available. Embarcadero Technologies, for

instance, has developed a series of excellent tools for developing SQL-based

client/server solutions. The DBArtisan product is very similar in feature and

function to the Enterprise Manager and will run against many servers other

than Microsoft SQL Server. In a multiple-vendor product environment, having a single product to interface with your data servers can be a great benefit. I

have included demo copies of each of these tools on the CD-ROM

accompanying this book. You can also reach Embarcadero on the Internet at

www.embarcadero.com.

The amount of thought and planning that has gone into these products is

impressive. As with other tools of this type on the market, you will experience http://www.itknowledge.com/reference/standard/1576101495/ch03/067-071.html (2 of 3) [1/27/2000 6:15:31 PM]

Microsoft SQL Server Black Book:Development Versus Production

a short learning curve. Throughout the book I will mention where the

Embarcadero products have proven useful. The use and methods of third-party

applications are not the focus of this book; I will use them in examples where

possible, however, because tools like these do provide significant benefits.

Most of these third-party modeling and development tools do a very good job

at what they were designed to do. Take some time to learn what each target

design is before committing to a purchase.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch03/067-071.html (3 of 3) [1/27/2000 6:15:31 PM]

Microsoft SQL Server Black Book:Development Versus Production



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Data Modeling**

When you begin the design process, good planning is essential. Pay particular attention to the underlying data structures and the end user’s needs. I have been contracted to fix many systems that were written by and for programmers and the user was the first on the list of considerations. Many projects start with the user’s needs in mind, but that focus soon shifts to the method of supplying the user with what they really need and the system requirements to meet those needs. When faced with the task of actually implementing the methods, programmers (myself included) will always take the path of least resistance. This path will cause you to stray from your focus of providing the user with a working product. The user is the focus of any project a developer undertakes. Users pay for our services with patience and feedback, as well as with money. The projects we develop are evaluated as to how well the user can perform the tasks we seek to automate with our creativity.

To keep myself on the right path when developing systems, I try to view the database objects as entities for as long as possible. Just as the term *entity* suggests, I give each order, load, or message a human-like attribute—a “personality.” If you think in these terms, you can treat each entity and how it relates to other objects or processes in a very complete, well-thought-out manner; you can spend more time planning and less time fixing.

I like to model either at the physical or conceptual layer a bit longer than most developers. I have found that by spending more time at these levels, less revision time is required down the road. Use the data-modeling technique as a road map. Follow the design as closely as you can, but keep in mind it is only

http://www.itknowledge.com/reference/standard/1576101495/ch03/071-073.html (1 of 3) [1/27/2000 6:15:33 PM]

Microsoft SQL Server Black Book:Development Versus Production

a guide. When making changes to your data structures, update your data model to reflect your modifications. Plan a few hours a month for keeping your data model as current as possible, whether it needs it or not.

Many of the third-party modeling tools available support designing and

modeling of your data. Most have a base set of features that can aid you in

staying current as long as possible. There will come a time when you lose the connection between your model and your data structure. Many things can

cause this separation. Typically, problems arise when you use a tool for data

modeling that does not allow you to create all the objects that your design

requires. I have used most of the major modeling applications on the market

today and have found limitations to each one. You’ll need to overcome each of these. Be creative; use scripts to create or modify the objects that you cannot

create with the modeling tool. Spending the time keeping your model current is time well spent.

When choosing a data-modeling tool, research each tool thoroughly. Keep in mind that these tools are not cheap. Look for a tool that allows you to model at either the conceptual and physical layer, and that allows you to create views

and triggers. These components are crucial to taking advantage of your

server’s ability to implement security and data integrity. The tool you choose should support modifying triggers and views beyond the basic cascading

INSERT, UPDATE, and DELETE statements.

Whether development or production, all systems have two levels of

configuration that need attention: server-level parameters and application

parameters.

**Server-Level Parameters**

Server-level parameters should be checked and verified when you install the

server. When creating a mirrored server to your production environment, you can use these server-level parameters to help tune or scale your server to react to queries in the same way.

One important point to mention here is that some server-level parameters are dynamic and some are static. Dynamic parameters can be changed on the fly

programmatically or through the Enterprise Manager. Other parameters are

static, which is to say that you must stop and start the MSSQLServer service in order for the changes to take effect. Check the Microsoft SQL Server Books

Online for details of each of these parameters before making any changes.

Let’s take a moment to look at a few of the more important server parameters. **User Connections**

The User Connections parameter should be configured after installation to

allow for the number of users on your system. You should set this value equal to the number of expected connections to your server.

Keep in mind that one client may have more than one connection to the server at any given time. When a multithreaded application connects to Microsoft

http://www.itknowledge.com/reference/standard/1576101495/ch03/071-073.html (2 of 3) [1/27/2000 6:15:33 PM]

Microsoft SQL Server Black Book:Development Versus Production

SQL Server for data access, the client may open several connections. In

addition, the SQL Executive and the other management-type accounts on your

server will use between five and seven connections.

The maximum value you can set for this option is a theoretical limit of 32,767.

This limit assumes you have the hardware and memory to support that kind of

burden. Each connection will take a small amount of memory away from the

amount of memory that Microsoft SQL Server will use.

In some of its documentation, Microsoft recommends approximately 37K per

user for each user connection. In other Microsoft documentation, however, I

have seen 40K and 42K used as the magic numbers for calculating the amount

of memory user connections will take. To be safe, I assume 42K per user

connection. By choosing the 42K amount, I am allowing a small bit of

memory to spill over into the caches for each user. That way, the most you can

be off is a net of 5K. Although that may not seem like much, using the 42K

value does give you a cushion.

**Tempdb in RAM**

Tempdb in RAM is another setting that can get you into trouble. There are

some good reasons to place Tempdb in RAM, but in all cases, you should test

the move to RAM thoroughly and make sure it really is providing an

improvement. As a general rule, if you do not have at least 64MB of RAM in

your server, you should leave Tempdb on-disk. When deciding whether to

place Tempdb in RAM, research this option very thoroughly in the Microsoft

SQL Server Books Online.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch03/071-073.html (3 of 3) [1/27/2000 6:15:33 PM]

Microsoft SQL Server Black Book:Development Versus Production



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Sort Pages**

Next is Sort Pages, which specifies the maximum number of pages that will be allocated to sorting query output on a user-by-user basis. On machines that execute large sorts, increasing this number can improve performance. Since additional sort pages will deplete your server’s available memory, you may need to adjust the amount of memory dedicated to SQL Server.

**Resource Timeout**

Resource Timeout is used by Microsoft SQL Server to determine the number of seconds to wait for a resource to be released by another process. The default setting for this parameter is 10. Increase this value only if the SQL Server error log has a lot of logwrite or bufwait timeout warnings in it.

**Note:** I recommend that the DBA of Microsoft SQL Server check the server error log and the Windows NT event log on a daily basis. Checking these logs each day can help you proactively troubleshoot events that occur on your server. Remember that your server’s load will fluctuate and grow over time. I have not found a system yet that you can set up and forget.

**Read-Ahead Optimization**

This subject will be covered in more detail later in Chapter 11. Most of the parameters in the RA section should be changed only if you are instructed to do so by a qualified support technician. Only two of the parameters are used for tuning the RA management for your server with any regularity: RA worker threads and RA slots per thread.

http://www.itknowledge.com/reference/standard/1576101495/ch03/073-076.html (1 of 3) [1/27/2000 6:15:34 PM]

Microsoft SQL Server Black Book:Development Versus Production

**RA Worker Threads**

The number of read-ahead worker threads you specify can impact the

performance of your server. RA worker threads are used by Microsoft SQL

Server to accommodate read-ahead requests. Microsoft recommends that this option be set to the maximum number of concurrent users on the server. (By

the way, a warning will be written to Microsoft SQL Server’s error log if the

number of threads that request read-ahead scans exceeds the number of

configured read-ahead slots.) Check the Microsoft SQL Server Books Online to find out more about how to interpret read-ahead settings.

**RA Slots Per Thread**

RA Slots Per Thread is another configuration option that should be approached with caution. The slots-per-thread option controls the number of simultaneous requests each read-ahead service thread will manage. The total number of

worker threads multiplied by the number of slots equals the number of

concurrent read-ahead scans that Microsoft SQL Server will support.

The default value of 5 should be fine. However, if your server has a very good disk subsystem, you might be able to increase the number of scans that a single thread can handle by adding to the default in small increments.

**Note:** As with any change you make to your server, you should change

options slowly and carefully. Write down the old value, then change one

value at a time and test the impact on performance. Test and benchmark your

changes and verify that they, in fact, did what you expected.

**Priority Boost**

You can increase the thread priority for SQL server within Windows NT by

changing the Priority Boost option. Only change this option on a machine

dedicated to SQL server, or you might find yourself being unable to launch

other applications or tasks on your server. Be leery of this setting when you

have a dual-processor machine as well. Setting this option can have a serious impact on your server’s ability to service login requests and manage printer or file access.

**Max Worker Threads**

Max Worker Threads configures the number of threads Microsoft SQL Server has to service SQL Server processes. Microsoft SQL Server makes use of the native thread services of Windows NT Server. This is one reason why SQL

Server is capable of such high performance—and why it only runs on

Windows NT.

Instead of the database having to create and manage threads internally, the

operating system shares threads with SQL Server. Other systems that are not

tied to the operating system in this way must maintain their threads at the

application level, thus slowing the application down (even a small amount) by adding additional overhead.

http://www.itknowledge.com/reference/standard/1576101495/ch03/073-076.html (2 of 3) [1/27/2000 6:15:34 PM]

Microsoft SQL Server Black Book:Development Versus Production

By changing this option, you control the number of threads allocated to the

user pool. When the number of user connections is less than the worker threads

setting, one thread handles each connection. However, if the number of

connections surpasses the Max Worker Threads value, thread pooling occurs.

You might think that pooling threads would slow down the operation of your

server and that this value should be set high. However, the default of 255 is too

high for most systems. Independent third-party studies have discovered that

setting this value to around 100 actually allows your server to operate much

more efficiently. Check the white paper by Compaq Corporation (included on

the CD-ROM) for more details.

**Lock Escalation Parameters**

Lock Escalation parameters is one of the hottest topics being discussed in

newsgroups on the Internet today (and probably for a long time to come as

well.) This set of options can lead you down the path of eternal server

adjustments if you do not look at all the reasons you are having locking issues

on your server in the first place.

It is easy to assume that you need to modify one of these parameters to thwart

deadlocks and hung processes. In some situations, the source of your problem

is in fact a misconfigured locking scheme, but in reality, most locking

problems are the result of questionable query methods and database design.

The Lock Escalation Threshold Maximum is intended as an upper boundary

for the server to use. The server will hold this maximum value of actual 2K

page locks per statement executed before attempting to escalate the lock to a

table lock. The lower boundary (the Lock Escalation Threshold Minimum) is

used to help keep SQL Server from locking a small table with only a few rows

in it every time a query is run against it. The minimum is used in conjunction

with the Lock Escalation Threshold Percentage to control this situation and

keep locks from getting out of hand.

All of these options can be overridden on a statement-by-statement basis with

optimizer hints. Including hints in your SQL statements will cause these

thresholds to be ignored. Great care should be taken when overriding the

Query Optimizer. The only hint I supply with any regularity is the NOLOCK

hint, which tells SQL Server not to lock any pages with this query at all. (The

use of optimizer hints is covered in Chapters 5 and 6.)

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch03/073-076.html (3 of 3) [1/27/2000 6:15:34 PM]

Microsoft SQL Server Black Book:Development Versus Production



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Fill Factor**

Fill Factor is a parameter that you should learn about almost as early as opening the shrinkwrap on your software. Changing this value affects how indexes will be rebuilt, as well as how the space on your server is used. See Figure 3.1 for the effects of using different fill factors on your databases. Both examples show the same index, yet the bottom set of 2K pages actually requires more disk space, due to the limit of filling the page only halfway.



**Figure 3.1** Fill factor’s impact on disk space used for an index.

The default for SQL Server is 0, or 100 percent. I usually change this to 80 or 90 percent, depending on the type of server that I am configuring. When you change this server option to a lesser value, the effect on your existing data is not apparent until you rebuild your indexes.

When an index is created on your SQL server, the fill factor is used to leave space available on pages for your index to grow into. This growth is what the fill factor allows to happen without fragmenting your index and slowing response down. When a page is filled, any other data that should be placed on that page has no place to go. Therefore, the data must be split and placed in other pages or even other extents. (*Extents* are eight contiguous 2K pages stored in your database.) When a page split or an extent switch occurs, you’re

http://www.itknowledge.com/reference/standard/1576101495/ch03/076-079.html (1 of 3) [1/27/2000 6:15:36 PM]

Microsoft SQL Server Black Book:Development Versus Production

spending extra time moving the disk head for read operations or write

operations.

I have seen some messages on the Internet that recommend using a 50 percent fill factor or even less, to help keep performance up on a system. This is fine if you have a lot of disk real estate and are not concerned with the amount of

space you are losing to indexes. I recommend keeping this value as high as

possible and using specific indexes with lower-than-normal fill factors to

lessen the impact on your disk space.

The only time I use a fill factor of 100 percent is for read-only data or for data that will be modified or changed so rarely that the impact on performance is

not an issue. The fill factor is only checked when you build or rebuild an

index. When you add data to an index, the fill factor is ignored.

These are the main server parameters that you should concern yourself with at this point in the process. That is not to say, however, that your project won’t

require one or more of the other parameters to be configured differently from SQL Server’s default setting. You should become aware of each parameter and its function on your server. The more you know about these parameters, the

more you can understand why things happen the way they do on your server. 

**Application Parameters**

Application-specific parameters represent the second level of configuration.

Microsoft SQL Server does not allow you to configure global client variables; therefore, you should develop a table-driven approach to application

parameters.

**DBCC PINTABLE**

A good technique for using table-driven parameters is the DBCC PINTABLE option. This option allows you to place the table containing your parameters

into the data cache and keep it from being flushed, thereby increasing

performance of queries against this table.

DBCC PINTABLE marks a table, once it is used, to remain in the data cache until “unpinned.” DBCC PINTABLE does not read the table into cache;

instead, as the table is queried, the data pages are put into cache, then marked so they do not get flushed later. Once the table is pinned, changes to the data

are logged, and the table can be recovered in the event of a media failure.

Use this feature with caution. If you pin a large table, it will consume your

available data cache and impact the performance of queries against all your

other tables. When storing your application variables on your SQL server,

beware of creating a table that is high maintenance and low

return-on-investment. Create only the indexes needed and do not create a huge, wide, and hard-to-understand table structure.

**Registry-Type Tables**

There are two good methods for using server-based client application

http://www.itknowledge.com/reference/standard/1576101495/ch03/076-079.html (2 of 3) [1/27/2000 6:15:36 PM]

Microsoft SQL Server Black Book:Development Versus Production

parameters. First is the creation of a registry-type table. This table allows you

to store a broad range of data in a very flexible format. The Windows NT or

Windows 95 Registry can be used as a model for this kind of table. Beware,

however, of registry pitfalls. You have seen what can happen to a machine

running Windows 95 or Windows NT when applications write to the registry

in a haphazard way—disasters and the dreaded blue screen of death. Clean up

after yourself and pay attention to keeping the data and keys clean and

up-to-date. Remove unused keys and perform maintenance on this table

regularly. Store only single-value data in this type of structure.

A registry-type table does not lend itself to relational or set-based data queries.

Using a list-type table structure to return result sets to your client application is

much more efficient.

**List-Type Tables**

A list-type table can be joined with other tables or used to return rows to a

client application. One possible use of this kind of information might be a

current user table. This table may store such information as the key values of

the records the user is accessing and permission variables on those objects. By

storing this information in a table, users can very easily be guided back to

where they were last working, making the application appear to be smart and

intuitive. Indexes play an even greater role in performance with this kind of

table than with a normal data structure.

Beware of multiple indexes on this kind of table. Look at the queries that run

against it and determine the minimum index configuration required. Updates

are usually more frequent against this kind of table, and therefore the index

values will need to be updated often.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch03/076-079.html (3 of 3) [1/27/2000 6:15:36 PM]

Microsoft SQL Server Black Book:Development Versus Production



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Setup Scripts**

A script is simply a file containing SQL commands that is executed against your SQL server to create, modify, or manipulate data or objects. Most large systems require you to run some sort of configuration scripts to prepare the system for operation. Very few good data-modeling tools are available that let you create a finished structure that takes into account revisions of objects and changes to permission structures. These types of issues are easily handled with scripts.

For example, scripts can be used to add the user-defined error messages for your application to the SQL server. Although these error messages can be configured by hand on a case-by-case basis, doing so lends itself to human error and inaccurately typed messages. Creating and executing a script file is much faster and allows you to configure two or more servers with exactly the same options. In fact, most data modeling tools allow you to create script files to run against your server for object creation.

When writing scripts, you should keep a few ground rules in mind:

**1.** A script is broken down into batches. These batches are designated by the word *GO*. **2.** If an error occurs in a batch, that batch will not execute. Other batches in the script will still run, however.

**3.** You cannot create and reference an object in the same batch. You also cannot drop and create an object with the same name in the same batch.

**4.** SET statements take effect at the end of a batch.

**5.** Be sure to use a lot of comments in your scripts.

Although I will not go into writing SQL statements in detail until Chapter 5, these tips should be committed to memory. The above list will help you troubleshoot performance problems when executing scripts.

**Sample Server Setup Script**

In Listing 3.1, I’ve provided a sample setup script for adding or modifying user-defined messages. Notice that I have commented extensively in this script, even though it will be run only once in a while and should not change often. This allows other programmers to read and understand quickly what I was attempting to do with the script.

The /\* and \*/ must be placed at the beginning and end of comments. Notice that the formatting shown here is for legibility only. Each of the EXEC lines is continued on the next line in this text (as indicated by indention) but should be on the same line in your script. See the sample scripts included on the CD-ROM for the actual

http://www.itknowledge.com/reference/standard/1576101495/ch03/079-082.html (1 of 3) [1/27/2000 6:15:37 PM]

Microsoft SQL Server Black Book:Development Versus Production

text file.

**Listing 3.1** Sample setup script.

/\*

------------------------------------------------------------------------- Name : Server Message Init

File Name : ServerMsgInit.sql

Dated : 08/17/97

------------------------------------------------------------------------- Description

------------------------------------------------------------------------- Create user-defined error messages to be called by SQL statements and stored procedures with the system. Drop any existing messages in range to meet my needs.

------------------------------------------------------------------------- \*/

/\* Must be called within the Master Database \*/

USE MASTER

/\* Add new messages to SQL server \*/

/\* The TRUE option enables NT event logging \*/

/\* The REPLACE option will overwrite existing message numbers \*/

/\* General SQL Error \*/

EXEC sp\_addmessage 60000, 10, "General SQL Error in <%s>. This will be logged in the Windows NT Eventlog.", us\_english, TRUE, REPLACE

/\* Business Rule Violation \*/

EXEC sp\_addmessage 60001, 10, "<%s> has violated business rule <%s>. This will be written to the Windows NT Eventlog, and E-Mail will be sent.", us\_english, TRUE, REPLACE

/\* Missing E-Mail Address \*/

EXEC sp\_addmessage 60002, 10, "There is no defined E-Mail recipient for Business Rule <%s>. This will be written to the Windows NT Eventlog.", us\_english, TRUE, REPLACE

/\* SQL Error In Procedure (Logged) \*/

EXEC sp\_addmessage 60003, 10, "SQL Error %s in function ['%s'].

Attempting to %s.", us\_english, TRUE, REPLACE

/\* SQL Error In Procedure (Non-Logged) \*/

EXEC sp\_addmessage 60004, 16, "SQL Error %s in function ['%s'].

Attempting to %s. This error can occur due to %s and will be ignored.", us\_english, FALSE, REPLACE

/\* Finished \*/

PRINT 'Finished.'

GO

Setup scripts can be executed from a graphical interface or from the DOS Query tool provided with SQL Server. To use this sample script, you open the text file with any of the available text-based query tools, then execute it. The following is the sample output from this script when executed:

Replacing message.

New message added.

Replacing message.

New message added.

http://www.itknowledge.com/reference/standard/1576101495/ch03/079-082.html (2 of 3) [1/27/2000 6:15:37 PM]

Microsoft SQL Server Black Book:Development Versus Production

Replacing message.

New message added.

Replacing message.

New message added.

Replacing message.

New message added.

Finished.

This sample is intended only to illustrate what a script looks like. I will explain in later chapters how to write scripts to perform different tasks on your server. Scripts are a very powerful tool that can be used to save a lot of time and ensure consistency among servers.

Another method of creating these scripts is to let SQL Server’s Enterprise Manager create them for you. Although Enterprise Manager cannot script user-defined error messages, it can generate scripts for just about anything else.

Previous Table of Contents Next

**Products | Contact Us | About Us | Privacy | Ad Info | Home**

Use of this site is subject to certain Terms & Conditions, Copyright © 1996-2000 EarthWeb Inc.

All rights reserved. Reproduction whole or in part in any form or medium without express written permission of EarthWeb is prohibited.

http://www.itknowledge.com/reference/standard/1576101495/ch03/079-082.html (3 of 3) [1/27/2000 6:15:37 PM]

Microsoft SQL Server Black Book:Development Versus Production



To access the contents, click the chapter and section titles. 

**Microsoft SQL Server Black Book**

*(Publisher: The Coriolis Group)*

Author(s): Patrick Dalton

ISBN: 1576101495

Publication Date: 07/01/97

Brief Full 

Advanced Search

Search Tips 

**Search this book:** 

Previous Table of Contents Next

**Scripting Objects**

You can use SQL-DMO (SQL Data Management Objects) and the SQL Enterprise Manager to generate scripts of objects that exist on your SQL server. You can select pre-existing objects in the Enterprise Manager and have the script created that would be necessary to drop and re-create the object on your server or any other SQL server. This can be very beneficial when you need to modify an existing object quickly.

Figure 3.2 shows the dialog box that appears when you select an object in the Server Manager window, then select the Object menu and the Generate SQL Script option. This is a very useful feature in a development environment. You have the ability to generate scripts for groups of objects, as well as for specific objects. Notice the Object Creation and Object Drop checkboxes. These options allow you to drop and re-create objects from the script or scripts you generate.



**Figure 3.2** The Generate SQL Scripts dialog box.

You might want to generate scripts to document the objects created by any third-party vendor’s software residing on your server. To do so, you could create a file with all the code essential to running other people’s software on your server. Remember that the Enterprise Manager will generate scripts that

http://www.itknowledge.com/reference/standard/1576101495/ch03/082-086.html (1 of 4) [1/27/2000 6:15:40 PM]